L'APRENENTATGE AUTOMÀTIC INCREMENTAL I LA SEVA APLICACIÓ AL PLN INTER-ACTIU

FRANCESC BENAVENT I PORTABELLA

TESI DOCTORAL UPF / 2013

DOCTORAT EN COMUNICACIÓ LINGÜÍSTICA I MEDIACIÓ MULTILINGÜE.

DIRECTOR DE LA TESI

DR. TONI BADIA I CARDÚS

DEPARTAMENT DE TRADUCCIÓ I CIÈNCIES DEL LLENGUATGE

upf. Universitat Pompeu Fabra Barcelona

L'APRENENTATGE AUTOMÀTIC INCREMENTAL I LA SEVA APLICACIÓ AL PLN INTER-ACTIU

FRANCESC BENAVENT I PORTABELLA

TESI DOCTORAL UPF / 2013

DOCTORAT EN COMUNICACIÓ LINGÜÍSTICA I MEDIACIÓ MULTILINGÜE.

DIRECTOR DE LA TESI

DR. TONI BADIA I CARDÚS

DEPARTAMENT DE TRADUCCIÓ I CIÈNCIES DEL LLENGUATGE

upf. Universitat Pompeu Fabra Barcelona

A l'Ester,
per voler compartir la meva il·lusió;
a la Júlia i el Guillem,
per deixar-me compartir els seus somnis.

AGRAÏMENTS

Aquesta tesi no hauria estat possible sense l'ajuda, la paciència i els ànims de moltes persones, per això vull agrair les aportacions de tots aquells que m'han ensenyat i enriquit al llarg d'aquests anys en el món acadèmic.

Vull donar les gràcies molt especialment al meu director de tesi: el Dr. Toni Badia i Cardús. No tinc cap dubte que sense ell aquesta tesi no existiria. Haig d'agrair-li que em proposés matricular-me en la Llicenciatura de Lingüística quan ens vam conèixer a un congrés de la SEPLN, que m'oferís entrar al GLiCom per col·laborar en un del seus projectes, que em suggerís continuar amb el *Màster de Ciència Cognitiva i Llenguatge* i, finalment, que m'animés a fer aquest Doctorat. Evidentment, també vull d'agrair-li tot el temps, l'ajuda i els valuosos consells que m'ha ofert durant la planificació, desenvolupament i redacció d'aquesta tesi.

També vull agrair el tracte rebut, i els valors i coneixements transmesos, a tots els professors que m'han format durant aquests apassionants anys: als professors de Lingüística de la Universitat Pompeu Fabra i als professors de Màster de la Universitat de Barcelona. Gràcies per ser, a més de bons professionals, persones tan properes.

No puc oblidar-me dels companys del GLiCom, i posteriorment grup de Veu i Llenguatge, que m'han acompanyat al llarg d'aquest viatge. A la Gemma Boleda per la cordialitat amb que em va rebre i introduir en el grup. A tots amb els que vaig compartir projectes: l'Stefan Bott, el Beto Boullosa, la Judith Domingo, el Juanma Garrido, el Bernat Grau, la Yesika Laplaza, el Carlos Rodríguez i l'Oriol Valentín, per ensenyar-me a treballar en equip. A la Maite Melero per la seva franquesa, i al Guillem Massó i la Teresa Suñol per la seva transparència. A la Roser Saurí per les seves assenyades opinions i excepcionals classes de ioga. I, especialment, a la Montse Marquina per aquells cafès de després de dinar acompanyats de llargues i valuoses converses. A tots ells, per compartir les seves recerques i ajudar-me a entendre i valorar aquest món.

En darrer lloc, al Martí Quixal vull agrair-li la seva amistat, la seva actitud davant la vida i la seva serenitat contagiosa, consells i ànims que m'han ajudat a relativitzar els problemes i a no prendre decisions precipitades. Gairebé tant important com això han estat l'esforç i l'ajuda que m'ha ofert en aquest tram final de redacció de la tesi, sense les seves correccions i revisions no sé si m'hauria atrevit a presentar-la.

Finalment, vull donar les gràcies a la meva familia: l'Ester, la Júlia i el Guillem. Gràcies per entendre que necessitava fer aquest camí per poder tancar una etapa, gràcies per la vostra paciència i recolzament, i gràcies per totes les hores que injustament us he robat.

Barcelona, 11 de setembre del 2013

RESUM

En aquest treball es proposa utilitzar tècniques d'Aprenentatge Automàtic Incremental, també conegut com Aprenentatge On-line, per resoldre tasques de Processament de Llenguatge Natural de manera més eficient. També s'estudia la viabilitat tècnica de la seva aplicació en el desenvolupament d'entorns Inter-Actius d'anotació lingüística.

El document està estructurat en tres parts: la justificació conceptual de la proposta, la viabilitat tècnica a partir de l'estat de la qüestió i les proves experimentals per obtenir dades quantitatives sobre l'eficiència assolida.

La primera part descriu la situació actual, basada en el paradigma d'aprenentatge *batch*, en qüestiona el consens existent i exposa les seves limitacions: econòmiques, tècniques i metodològiques. A continuació, presenta el paradigma incremental i planteja la manera en què una arquitectura Inter-Activa, basada en l'aprenentatge actiu i els algorismes incrementals, podria minimitzar el coll d'ampolla associat a l'anotació manual del corpus.

La segona part presenta l'estat de la qüestió de l'Aprenentatge Automàtic Incremental: els algorismes d'inducció de models, les arquitectures de combinació de classificadors i les tècniques auxiliars d'optimització i avaluació.

La tercera part del treball descriu la metodologia utilitzada en una sèrie de proves experimentals, amb quatre tasques de PLN, amb l'objectiu de quantificar la qualitat dels models induïts i l'eficiència dels entrenaments. Presenta els resultats de més d'un centenar d'experiments, analitza i justifica les corbes d'avaluació obtingudes i compara els entrenaments en termes de precisió i eficiència assolida.

Els resultats dels experiments validen la hipòtesi principal del treball, que defensa que mitjançant l'entrenament Inter-Actiu és possible obtenir models classificadors tant o més precisos que amb l'entrenament estàndard, però utilitzant tan sols una fracció del corpus existent; concretament, i segons les proves realitzades, requerint entre 5 i 100 vegades menys exemples.

Així mateix, també s'aprofundeix en l'anàlisi de les dades obtingudes durant els entrenaments basats en l'aprenentatge actiu, especialment en l'evolució dels graus de certesa de les seves classificacions i de la precisió d'aquestes estimacions. A partir d'aquestes dades es conclou que la selecció d'exemples basada en un llindar de certesa constant és massa sensible al valor triat, i es suggereix investigar algorismes d'entrenament actiu basats en llindars de certesa dinàmics.

RESUMEN

En este trabajo se propone utilizar técnicas de Aprendizaje Automático Incremental, también conocido como Aprendizaje On-Line, para resolver tareas de Procesamiento de Lenguaje Natural de manera más eficiente. También estudia la viabilidad técnica de su aplicación en el desarrollo de entornos Inter-Activos de anotación lingüística.

El documento está estructurado en tres partes: la justificación conceptual de la propuesta, la viabilidad técnica a partir del estado de la cuestión y las pruebas experimentales para obtener datos cuantitativos sobre la eficiencia conseguida.

La primera parte describe la situación actual, basada en el paradigma de aprendizaje *batch*, cuestiona el consenso existente y expone sus limitaciones: económicas, técnicas y metodológicas. A continuación, presenta el paradigma incremental y plantea la forma en que una arquitectura Inter-Activa, basada en el aprendizaje activo y los algoritmos incrementales, podría minimizar el cuello de botella asociado a la anotación manual de corpus.

La segunda parte presenta el estado de la cuestión del Aprendizaje Automático Incremental: los algoritmos de inducción de modelos, las arquitecturas de combinación de clasificadores y las técnicas auxiliares de optimización y evaluación.

La tercera parte del trabajo describe la metodología utilizada en una serie de pruebas experimentales, con cuatro tareas de PLN, con el objetivo de cuantificar la calidad de los modelos inducidos y la eficiencia de los entrenamientos. Presenta los resultados de más de un centenar de experimentos, analiza y justifica las curvas de evaluación obtenidas y compara los entrenamientos en términos de precisión y eficiencia alcanzada.

Los resultados validan la hipótesis principal del trabajo, que defiende que mediante el entrenamiento Inter-Activo es posible obtener modelos clasificadores tan o más precisos que con el entrenamiento estándar, pero utilizando únicamente una fracción del corpus existente; concretamente, y según las pruebas realizadas, requiriendo entre 5 y 100 veces menos ejemplos.

Así mismo, también profundiza en el análisis de los datos obtenidos durante los entrenamientos basados en el aprendizaje activo, especialmente en la evolución de los grados de certeza de sus clasificaciones y de la precisión de estas estimaciones. A partir de estos datos se concluye que la selección de ejemplos basada en un umbral de certeza es demasiado sensible al valor elegido, y se sugiere investigar algoritmos de entrenamiento basados en umbrales de certeza dinámicos.

ABSTRACT

In this work we propose the use of Incremental Machine Learning, also known as On-Line Learning, to solve Natural Language Processing tasks in a more efficient way. We also study the technical feasibility of its application to the development of inter-active environments of linguistic annotation.

The document is structured in three parts: the conceptual justification of the proposal, the technical feasibility by grounding it on state of the art techniques, and the experimental tests performed to obtain quantitative data about the efficiency achieved.

The first part describes the current trends in NLP based on the *batch* learning paradigm, it questions the existent consensus and it exposes its limitations: economical, technical and methodological. Right after, it presents the incremental paradigm and it outlines how an Inter-Active architecture, based on active learning and incremental algorithms, could minimize the bottleneck related to the manual corpus annotation.

The second part presents state of the art Incremental Machine Learning: the algorithms of model induction, the classifier combination architectures and the auxiliary techniques for optimization and evaluation.

The third part of this work describes the methodology used in a set of experimental tests, on four NLP tasks, with the goal of quantifying the quality of the induced models and the training efficiency. It presents the results of more than a hundred experiments, and it analyzes and justifies its evaluation curves and compares the different trainings on the achieved precision and efficiency.

The results of the experiments validate the main hypothesis of this work, which is that Inter-Active training makes it possible to obtain classifier models with as much or higher precision than with standard training, but using just a fraction of the existent corpus; in particular, and according to the results, reducing the number of training examples needed between 5 and 100 times.

Additionally, it also goes into detail in the analysis of the data obtained during the training based on active learning, especially on the evolution of the confidence levels of its classifications and the precision of these estimations. From this data we conclude that the example selection based on a constant confidence threshold is too sensitive to the given value, and we propose to research active training algorithms based on dynamic confidence thresholds.

PRESENTACIÓ

En la darrera dècada s'ha anat produint una convergència entre la Lingüística Teòrica de base empírica i la Lingüística Computacional basada en dades. Aquest apropament ha estat propiciat per l'objectiu compartit d'inferir models que expliquin el llenguatge i reforçat per la necessitat comuna de disposar de corpus de text anotats lingüísticament.

Malgrat els importants avanços aconseguits, facilitats en bona part per la utilització de tècniques tradicionals d'Aprenentatge Automàtic supervisat, encara hi ha camí per avançar. Lamentablement aquest paradigma presenta un important coll d'ampolla, el cost d'anotació dels corpus d'entrenament, que condiciona i limita tant la recerca bàsica com les aplicacions comercials.

Amb aquest treball es desitja explorar la utilització de les tècniques d'Aprenentatge Automàtic Incremental, mitjançant la seva integració en entorns eficients d'anotació assistida, per a superar o minimitzar algunes d'aquestes limitacions. I, indirectament, revaloritzar el paper dels algorismes incrementals d'Aprenentatge Automàtic en el camp del Processament del Llenguatge Natural. Després de l'interès despertat als anys 90, l'augment de potència i memòria dels ordinadors va fer pensar que ja no eren necessaris. Però, 20 anys després, la disponibilitat de quantitats massives de text sense anotar podria propiciar que aquesta tecnologia tornés a ser rellevant.

Les **hipòtesis** de partida en les que es basa aquest treball són:

- que el paradigma *batch* presenta algunes limitacions, la principal d'elles el *coll d'ampolla* que suposa l'anotació del corpus d'entrenament,
- que les eines d'anotació assistida poden mitigar aquest problema accelerant l'anotació manual i reduint el seu cost, i
- que la combinació de tècniques d'Aprenentatge Actiu amb algorismes d'Aprenentatge Automàtic incremental poden donar lloc a entorns d'anotació altament eficients.

I els o**bjectius** perseguits:

- presentar els avantatges de la utilització del paradigma incremental d'Aprenentatge Automàtic en el camp del Processament de Llenguatge Natural,
- verificar la viabilitat tecnològica de la seva aplicació en el desenvolupament d'entorns eficients d'anotació lingüística Inter-Activa[1], i

[1] L'adjectiu *Inter-Actiu* , definit amb detall al Capítol 5, és un terme creat *ad-hoc* per descriure els sistemes interactius basats en la combinació de tres tecnologies: l'aprenentatge actiu, el *bootstrapping* i l'aprenentatge incremental.

- confirmar aquesta eficiència, demostrant que aquestes tècniques permeten obtenir classificadors amb una precisió equivalent a les tècniques estàndard, però utilitzant una quantitat inferior d'exemples d'entrenament.

La **metodologia** utilitzada ha estat essencialment experimental, tot i que ha tingut tres fases clarament diferenciades:

i) la justificació conceptual de la proposta, basada en la recerca de les tècniques actualment majoritàries, de les seves limitacions i dels potencials beneficis de la seva superació,
ii) l'estudi de la seva viabilitat tècnica, a partir de la recerca sobre l'estat de la qüestió de la tecnologia incremental per garantir que no suposa una limitació,
iii) les proves experimentals per obtenir dades quantitatives que permetin comparar, a partir d'una selecció de tasques de PLN representatives, el nivell d'eficiència i precisió assolit per les diferents tècniques d'entrenament.

El treball, que reflecteix aquest procés, està estructurat seguint aquestes tres parts.

La **Part I** d'aquest treball descriu la situació actual basada en el paradigma *batch* i qüestiona el consens existent a l'hora de plantejar les tasques de PLN com un problema no-incremental i descriu les limitacions que se'n deriven. Tot seguit argumenta que la utilització d'algorismes incrementals d'Aprenentatge Automàtic podria resoldre parcialment aquestes limitacions i facilitar les tasques d'anotació mitjançant el desenvolupament d'eines Inter-Actives.

El **Capítol 1** ofereix el context històric que ha permès la convergència de diferents disciplines al voltant del processament automàtic dels corpus lingüístics. El **Capítol 2** descriu l'Aprenentatge Automàtic *batch*, el paradigma actual responsable dels importants avanços produïts en aquest camp, així com de les importants limitacions que presenta: econòmiques, tècniques i metodològiques. El **Capítol 3** fa un parèntesis per analitzar la importància dels corpus d'entrenament, i els beneficis i reptes que suposa la seva utilització a gran escala.

El **Capítol 4** presenta el concepte d'algorismes d'aprenentatge incremental i discuteix les seves característiques funcionals, ideals i formals; seguidament, presenta els avantatges que ofereixen en relació als algorismes *batch*. Finalment introdueix el tema de la seva dependència respecte de l'ordre dels exemples, que tradicionalment ha estat considerat com una limitació. El **Capítol 5** presenta l'anotació Inter-Activa, un model d'anotació assistida basat en la utilització interactiva de l'Aprenentatge Actiu, que pot beneficiar-se considerablement dels algorismes incrementals.

La **Part II** del treball, amb un enfocament més tècnic, presenta l'estat de l'art de l'Aprenentatge Automàtic Incremental i totes les tècniques associades. El **Capítol 6** repassa les diferents famílies d'algorismes d'Aprenentatge Automàtic i presenta les variants incrementals que existeixen en cada paradigma. El **Capítol 7** presenta les combinacions de

classificadors, una tècnica plenament compatible amb el model incremental i que pot millorar considerablement la precisió dels sistemes d'anotació. Finalment, el **Capítol 8** descriu, des del punt de vista incremental, algunes tècniques auxiliars necessàries en qualsevol tasca d'Aprenentatge Automàtic: el preprocessament dels exemples, la selecció de trets, l'optimització de paràmetres i l'avaluació dels sistemes.

La **Part III** descriu la metodologia i les proves experimentals realitzades per comprovar els beneficis de diferents tècniques d'entrenament incremental i validar la hipòtesi que els algorismes incrementals poden ajudar a obtenir models classificadors igual de precisos mitjançant entrenaments més eficients que requereixin l'anotació de només una fracció del corpus total.

El **Capítol 9** descriu el programari utilitzat i els corpus textuals a partir dels quals s'han preparat les tasques de referència. El **Capítol 10** presenta la definició de les tasques i les proves preliminars de selecció que han permès triar les quatre tasques de PLN representatives que han estat utilitzades en els diferents experiments. El **Capítol 11** descriu les corbes d'avaluació i presenta els resultats de l'entrenament incremental estàndard, que serà utilitzat com a referència en els capítols següents.

El **Capítol 12** presenta una tècnica d'entrenament incremental basada en el pre-entrenament del model amb un corpus auxiliar i analitza els resultats obtinguts amb la seva utilització. El **Capítol 13** descriu extensament els resultats de l'entrenament mitjançant aprenentatge actiu, tècnica que permet accelerar considerablement l'entrenament reduint el cost d'anotació, i n'analitza els resultats. El **Capítol 14** analitza la possibilitat de combinar simultàniament les dues tècniques anteriors, amb l'objectiu de sumar els beneficis individuals.

El **Capítol 15** analitza l'evolució de l'eficiència del classificador a partir de l'ASR (*Annotation Sampling Ratio*) dinàmic i n'extreu algunes conclusions. Els darrers dos **Capítols 16 i 17**, analitzen l'evolució dels nivells de certesa del classificador i l'evolució de la precisió d'aquesta estimació, informació que facilita la comprensió dels processos interns durant l'entrenament incremental.

Finalment, la **Part IV** presenta les **Conclusions** extretes al llarg d'aquesta recerca i apunta futures línies de recerca que permetrien aprofundir en aquest camp, i a la **Part V** es poden consultar els **Annexes** amb dades concretes sobre els experiments realitzats.

ÍNDEX DE CONTINGUTS

ÍNDEX DE FIGURES

ÍNDEX DE TAULES

PART I:

L'APRENENTATGE INCREMENTAL EN PLN: MOTIVACIONS I AVANTATGES

1 ANTECEDENTS

1.1 INTRODUCCIÓ

No hi ha cap dubte que el llenguatge és una de les característiques fonamentals que ens defineixen com a humans, tant pels efectes cognitius que té en el pensament dels individus com per la influència determinant que ha tingut en el desenvolupament de les societats i en la seva evolució cultural. Per això les tecnologies associades a la comunicació i a l'emmagatzemament de la informació han estat entre les més influents en la història de la humanitat, al mateix nivell que les tecnologies bèl·liques, energètiques i alimentàries.

Amb els primers ordinadors va néixer la possibilitat de tractar la informació de manera automàtica, i no va passar gaire temps fins que es va intentar apropar la informàtica cap al llenguatge natural, de manera que els ordinadors poguessin tractar la informació en el seu format habitual: el text. Però els anys han demostrat que no era una tasca senzilla, i que tot i que s'han anat assolint molts altres objectius de la Intel·ligència Artificial[1] (IA), els resultats de mig segle de recerca en el tractament del llenguatge no han estat els esperats.

Curiosament, la dificultat d'aquesta tasca ens ha ensenyat molt sobre la complexitat del propi llenguatge i els seus mecanismes. La necessitat de formalitzar intuïcions lingüístiques, la disponibilitat de corpus electrònics i la possibilitat de contrastar les teories amb dades objectives, ha permès el desenvolupament de la Lingüística Empírica[2] (LE), que amb l'ajut del Processament del Llenguatge Natural[3] (PLN) recolzat per tècniques d'Aprenentatge Automàtic[4] (AA), han donat lloc a una interessant àrea multidisciplinària.

En aquest capítol es descriu breument l'evolució de tres disciplines que, malgrat que inicialment van desenvolupar-se separadament, han acabat convergint en el que constitueix el paradigma actual del tractament automàtic del llenguatge.

1.2 LINGÜÍSTICA EMPÍRICA

La Lingüística és una disciplina relativament jove amb poc més d'un segle d'existència que té com a objecte d'estudi el llenguatge humà, i que té com a objectiu crear models i teories que expliquin el seu funcionament. Després d'una primera etapa descriptiva centrada en el lèxic i la morfologia va arribar, a mitjans del segle XX, a una segona etapa més teòrica centrada en l'estudi de les regles combinatòries que generen la riquesa expressiva del llenguatge.

---

[1] *Artificial Intelligence (AI)*
[2] *Empirical Linguistics (EL)*
[3] *Natural Language Processing (NLP)*
[4] *Machine Learning (ML)*

Lamentablement, més enllà de les gran aportacions fetes per la lingüística generativa desenvolupada per **[Chomsky, 1965]**, en aquests anys es va iniciar una lliscant tendència cap a un teoricisme que, sense adonar-se'n, va acabar substituint l'objecte d'estudi de la lingüística. Amb la justificació de voler descartar elements superflus i irrellevants, es va substituir l'estudi de la llengua real utilitzada pels parlants per l'estudi de les seves intuïcions sobre la llengua, un reflex idealitzat i simplificat més fàcil de tractar. La conseqüència d'aquest fet va ser que la Lingüística es va allunyar de l'enfocament científic **[Sampson, 2002]** i per tant va renunciar a obtenir dades de la realitat amb les quals poder contrastar les diferents teories:

> *"Language is people talking and writing. It is a concrete, tangible aspect of human behavior. //. Strange as it seems, in recent decades linguistics has not been an empirical science in practice. Linguists' 'grammars' have not been responsive to observations of concrete linguistic behavior." [Sampson 2002:1]*

Però a mesura que es van anar disposant de corpus suficients com per fer recerques *de camp* estadístiques, van començar a aparèixer les primeres discrepàncies entre les dades i algunes idees de gramaticalitat que fins al moment eren àmpliament acceptades **[Bybee, 2001]**. A partir d'aquest moment alguns lingüistes van defensar la necessitat de tornar a l'objecte original d'estudi, de centrar-se en les produccions reals fetes pels parlants i supeditar les teories a les dades observades, cosa que dona lloc a la Lingüística Empírica disposada a replantejar les bases de la Lingüística:

> *"[…] there is a very serious mismatch between the results of quantitative studies and grammatical accounts –both descriptive and normative- that rely exclusively on imaginary data. //. Hallan goes even further in suggesting that the availability of large corpora might call for a general reassessment of grammatical categories." [Bybee 2001:4-5]*

Així doncs, sembla que la Lingüística comença a assumir la necessitat de substituir la introspecció i les intuïcions lingüístiques per una metodologia més científica basada en els estudis quantitatius realitzats sobre corpus de produccions reals.

1.3 APRENENTATGE AUTOMÀTIC

L'Aprenentatge Automàtic (AA) és un subcamp nascut de la Intel·ligència Artificial (IA) que ha rebut aportacions de diferents disciplines, principalment de la ciència cognitiva, de la computació, del reconeixement de patrons i de l'estadística **[Aha, 1995]**. I el seu objectiu és desenvolupar algorismes que permetin als ordinadors aprendre tasques a partir d'exemples. Des dels orígens de la IA es va observar que moltes de les tasques proposades, que les persones podien realitzar amb facilitat, eren difícilment formalitzables en forma d'algorisme de manera que el pogués executar un ordinador. I per tant, es va intuir que potser seria més senzill recopilar conjunts d'exemples de realitzacions de les tasques i desenvolupar algorismes que induïssin les regles i els mètodes a partir d'aquestes dades.

Així doncs, la recerca teòrica en AA tracta principalment amb un tipus d'aprenentatge inductiu anomenat aprenentatge supervisat. En aquest tipus d'aprenentatge és presenten a l'algorisme un conjunt d'exemples etiquetats, i l'algorisme utilitza aquestes dades per induir un classificador. Un classificador és una funció que relaciona exemples amb determinades classes preestablertes, fins i tot, gràcies a la seva capacitat de generalització, a exemples que no ha vist anteriorment. L'objectiu de l'algorisme d'AA és crear un model, o ajustar els seus paràmetres, de manera que minimitzi el seu error de classificació davant d'exemples desconeguts **[Mitchell, 1997]**.

Tot i que existeixen molts algorismes d'AA basats en mètodes simbòlics, en els darrers temps l'enfocament estadístic ha anat convertint-se en dominant, fins al punt que s'ha començat a esvair la línia que separa l'AA de l'estadística. Potser, la principal diferència és que a l'estadística no s'acostumen a emprar representacions conceptuals de primer ordre, ni interessa la plausibilitat cognitiva dels processos d'aprenentatge. A més, no és estrany que els investigadors en AA tinguin influències, o s'autoimposin restriccions, motivades per aspectes cognitius, com la comprensibilitat dels models induïts, la tolerància al soroll en les dades o, com és el cas d'aquest treball, la incrementalitat de l'aprenentatge. És per això que l'AA cau clarament dins de l'àmbit de la IA i l'estadística no **[Aha, 1995]**.

Les tècniques d'AA són *disciplinàriament neutres* i, per tant, aplicables pràcticament a qualsevol camp en el qual es disposi d'exemples descrits estructuradament mitjançant informació qualitativa o quantitativa. Només cal tenir en compte que la naturalesa del problema pot afectar les característiques de les dades. Per exemple, tot i que l'existència de grans quantitats de dades en format electrònic fa que el tractament automàtic del llenguatge es pugui beneficiar d'aquestes tècniques, cal ser conscient que els problemes de PLN presenten alguns reptes **[Màrquez, 2001]** a l'hora d'aplicar l'AA a les seves dades:

- *Dimensionalitat*: representacions amb gran quantitat d'atributs
- *Dispersió*: baixa densitat d'exemples en l'espai d'atributs
- *Irrellevància*: gran quantitat d'atributs no significatius per a la classificació
- *Redundància*: elevats graus de correlació entre atributs dependents
- *Soroll*: presència significativa d'errors en les dades anotades

Aquestes característiques de les dades lingüístiques suposen una dificultat afegida a l'hora de processar eficientment els corpus disponibles, però existeixen tècniques per minimitzar-les i algorismes especialment capacitats per resoldre un o altre aspecte del problema.

1.4 PROCESSAMENT DE LLENGUATGE NATURAL

Històricament, el Processament del Llenguatge Natural (PLN) ha estat un camp situat a la intersecció entre la Informàtica i la Lingüística, i ha tingut com a objectiu desenvolupar eines i tècniques que permetin als ordinadors tractar automàticament el llenguatge humà amb més o menys profunditat. Al llarg del temps, la necessitat d'incorporar coneixement lingüístic en els processos d'anàlisi ha anat afavorint la convergència entre el PLN i la

Lingüística Empírica (LE), dues disciplines que s'han realimentat mútuament a l'hora de dissenyar tècniques i desenvolupar recursos.

Tot i que el PLN cobreix tant el tractament del llenguatge parlat com l'escrit, i tant l'anàlisi com la generació, en els darrers anys el nucli de la disciplina s'ha anat situant en l'anàlisi del llenguatge textual. Els motius són diversos, però en bona part pot explicar-se per la disponibilitat creixent de grans corpus textuals en format electrònic i la influència de les eines automàtiques basades en l'AA.

Actualment el PLN és una disciplina basada per un costat en els coneixements teòrics i formals de la LC i per l'altre en les tècniques i avenços en l'AA. És aquesta combinació la que ha permès desenvolupar eines capaces de realitzar automàticament diferents tasques lingüístiques: anàlisi morfològica, desambiguació morfosintàctica, detecció d'entitats amb nom, reconeixement de sintagmes i oracions, anàlisi sintàctica, resolució anafòrica, detecció de rols semàntics o anàlisi de discurs. Habitualment la qualitat dels resultats disminueix a l'augmentar el nivell d'abstracció de la tasca, però en totes elles s'estan aconseguint resultats prometedors.

En un primer moment, el PLN es va centrar en la creació manual de gramàtiques que s'aplicaven automàticament al text d'entrada, però des de finals del segle XX els mètodes empírics basats en la utilització d'algorismes d'AA que indueixen models predictius a partir de corpus de llenguatge natural han anat agafant força fins arribar a ser, a inicis del segle XXI, totalment dominants **[Daelemans, 2005]**.

Els principals motius que expliquen aquesta evolució són tant tecnològics com pràctics. En els darrers anys, l'increment de la potència computacional i de la capacitat d'emmagatzemament dels ordinadors típics ha arribat a nivells que permeten, amb molta facilitat, aplicar mètodes estadístics a les gran quantitats de text i veu que hi ha disponible en format electrònic.

A més, la recerca s'ha vist influenciada per la creixent demanda comercial d'aplicacions de PLN que fossin escalables i prou robustes com per tractar llenguatge real. I en aquesta línia, els èxits aconseguits en el camp del tractament de la parla i de la recuperació d'informació, tots dos gràcies a mètodes estadístics d'una relativa simplicitat, han sigut un referent a seguir per tota la disciplina **[Daelemans, 2005]**.

1.5 ANOTACIÓ LINGÜÍSTICA DE CORPUS

Tant la LE com el PLN, recolzats per l'AA, tenen el mateix objectiu, crear descripcions predictives del llenguatge observat, mitjançant regles o models lingüístics estadístics. La diferència és que en el primer cas la motivació és científica, establir les lleis i propietats que defineixen el llenguatge, i en el segon la motivació és pragmàtica, augmentar la qualitat dels sistemes de processament de llenguatge natural.

Per tant és lògic que aquestes tres disciplines comparteixin molts elements comuns i presentin necessitats i capacitats complementàries. La Lingüística Empírica necessita corpus anotats tant per poder validar les seves teories com per fer recerca bàsica que n'inspiri de noves. L'Aprenentatge Automàtic, aplicat al tractament del llenguatge, també necessita corpus anotats per poder induir models que permetin analitzar texts de manera automàtica. I el Processament de Llenguatge Natural, inspirat per la LE i basat en l'AA, té com a objectiu desenvolupar eines que processin els texts i n'extreguin informació. És per això que aquestes tres àrees formen un sistema realimentat al voltant de l'anotació de corpus:

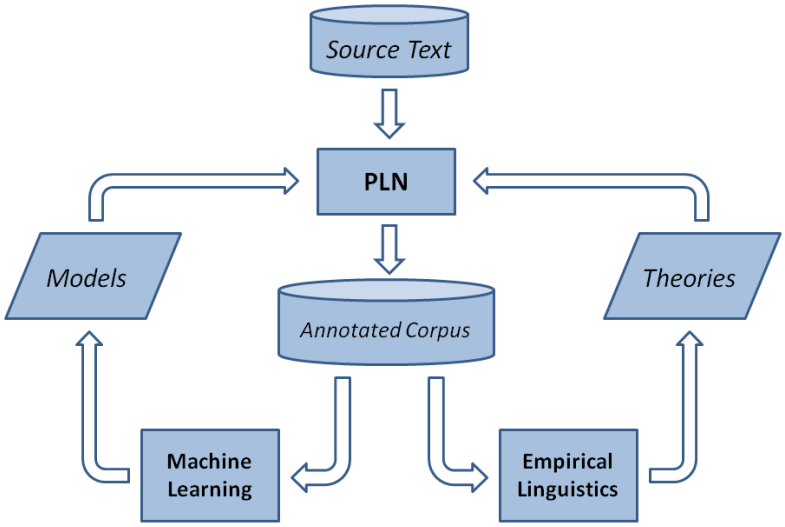

FIG. 1: Disciplines interrelacionades al voltant de l'Anotació de Corpus: LE + AA + PLN.

Si tenim en compte que el processament dels texts i l'extracció d'informació són aspectes parcials d'una anàlisi lingüística, i que l'anotació no és més que l'explicitació d'una anàlisi prèvia, és fàcil entendre perquè l'anotació lingüística de corpus se situa al centre del diagrama anterior. Ja sigui com a matèria primera de la qual extreure informació i validar models, o com a resultat dels processos d'anàlisi, l'anotació lingüística de corpus és un aspecte fonamental en el desenvolupament futur de tots tres camps.

Però per motius econòmics l'anotació lingüística de grans corpus només té sentit amb la utilització d'eines d'anotació automàtica. Afortunadament, tot i que els algorismes d'AA només poden induir models classificadors, qualsevol tasca lingüística pot plantejar-se amb més o menys èxit com una tasca de classificació. A més, encara que l'ambigüitat és una de les característiques fonamentals del llenguatge humà, i per tant, la classificació no-contextual acostuma a tenir problemes per identificar unívocament els elements, existeixen altres tècniques d'AA que ho poden resoldre.

La solució habitual consisteix a permetre que, en aquells casos ambigus, els classificadors assignin múltiples etiquetes i seguidament utilitzar un postprocés desambiguador. Aquesta desambiguació s'acostuma a basar en algorismes automàtics que indueixen models

probabilístics a partir de seqüències d'exemples etiquetats, models que poden estimar la versemblança d'una determinada etiqueta donat el seu context, i per tant permeten desambiguar aquests casos.

Així doncs, l'existència de tècniques que, a partir d'exemples, poden aprendre a resoldre els dos grans reptes transversals de l'anàlisi lingüística, la classificació i la desambiguació, apunta que aquesta associació de disciplines continuarà produint fruits en els propers anys. En la situació actual només cal anar millorant el rendiment de les eines i reduint els seus costos de desenvolupament. Precisament aquesta és una de les idees que hi ha darrera de la proposta d'aquest treball: la utilització d'algorismes incrementals a l'interior d'eines d'anotació inter-actives, amb l'objectiu d'accelerar el desenvolupament d'analitzadors lingüístics i reduir el cost d'anotació dels texts. .

2 APRENENTATGE AUTOMÀTIC

2.1 INTRODUCCIÓ

La motivació de l'Aprenentatge Automàtic supervisat és reemplaçar la programació explícita de les solucions a una tasca difícilment formalitzable per l'entrenament d'un sistema mitjançant la presentació d'exemples resolts. La seva essència és el disseny d'algorismes que implementin mecanismes d'inducció que aprenguin a generalitzar a partir d'exemples concrets.

Conceptualment un sistema d'aprenentatge està format per un *mòdul actuador*[1] que realitza una determinada tasca, normalment generar una sortida en funció d'una entrada, i d'un *mòdul d'aprenentatge*[2] que en funció de l'experiència modifica el mòdul actor perquè millori el seu comportament davant de les mateixes dades o similars **[Daelemans, 2005]**.

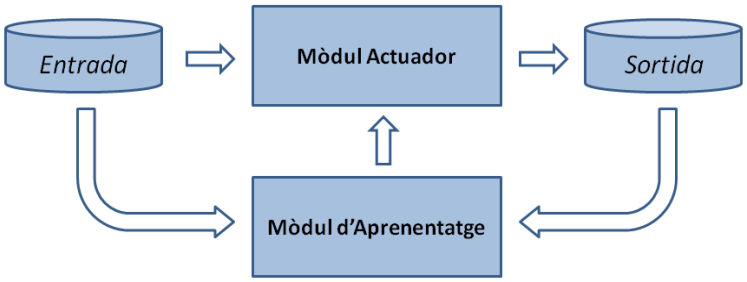

FIG. 2: Interacció entre el mòdul Actuador i el d'Aprenentatge

Aquesta arquitectura es basa en l'existència, a l'interior del *mòdul actuador*, d'un model representacional que determina o condiciona les seves sortides davant les diferents entrades. Sota aquest punt de vista, la tasca d'aprenentatge no es més que una cerca, dins l'espai de representacions, d'aquell model que optimitza el comportament del sistema. Tot i que en la majoria de casos, donada la mida d'aquest espai, la cerca exhaustiva de la representació òptima és computacionalment intractable. Per això és habitual incorporar heurístiques que redueixin l'espai de cerca, i per tant, introdueixen un primer biaix del qual cal ser conscients **[Mitchell, 1997]**.

En aquest capítol s'acota l'àmbit de l'AA per a centrar-se en l'àmbit més específic dels *classificadors supervisats*, seguidament es presenta i defineix el paradigma actual basat en l'aprenentatge en mode *batch*, i finalment es descriuen les principals limitacions d'aquest paradigma.

---

[1] *Performance Component*
[2] *Learning Component*

2.2 CLASSIFICADORS SUPERVISATS

L'AA és un camp que inclou diferents àrees especialitzades que tot i compartir la mateixa filosofia es mantenen relativament aïllades. Les següents seccions descriuen breument les seves diferències i acoten l'àmbit en el qual es mou aquest treball: els classificadors supervisats. Una vegada fet això es descriu l'aplicació d'aquests classificadors a l'automatització de tasques lingüístiques i s'introdueixen les dificultats específiques d'aquest tipus de tasques.

2.2.1 SUPERVISAT VS. NO-SUPERVISAT

El principal criteri que divideix el camp de l'AA és si l'aprenentatge es realitza en el que es denomina context *supervisat* o context *no-supervisat*.

Tal com s'ha explicat, el procés d'aprenentatge o entrenament, segons si es pren el punt de vista de l'algorisme o de l'expert, consisteix a extreure, a partir d'un conjunt d'exemples, generalitzacions que permetin realitzar una tasca de projecció entre les descripcions dels exemples d'entrada i les prediccions dels seus valors de sortida. Les descripcions dels exemples consisteixen en una sèrie de parells atribut-valor (numèrics, simbòlics o vectorials) que defineixen els seus trets formals i contextuals, juntament amb la predicció desitjada, que pot ser tant un valor numèric com un valor simbòlic –en aquest darrer cas s'acostuma a anomenar *etiqueta*. Per tant, la tasca de l'algorisme és predir un determinat valor de sortida a partir d'un patró d'entrada més o menys similar als que hagi vist durant l'entrenament, i per realitzar aquesta tasca necessita un conjunt d'exemples etiquetats amb els valors de sortida corresponents. Aquest model d'aprenentatge, on els exemples s'han etiquetat prèviament, és el que es coneix com a *supervisat*, i és el que s'utilitza habitualment en tasques d'anàlisi o reconeixement.

Però existeix un altre plantejament utilitzat en contexts *no-supervisats*, on els exemples contenen únicament la seva descripció, sense cap mena de valor de sortida associat. En aquests casos l'objectiu de l'algorisme no és aprendre a predir un determinat valor sinó estudiar els exemples per buscar similituds que permetin detectar patrons, regularitats o agrupaments naturals. Aquest model d'aprenentatge, conegut també com a *agrupament*[3] i molt relacionat amb la *mineria de dades*[4], s'utilitza en tasques exploratòries amb l'objectiu de descobrir categories o prototipus que facilitin l'anàlisi de les dades.

Finalment existeix un tercer plantejament anomenat aprenentatge *semi-supervisat*, que pretén utilitzar els avantatges de l'aprenentatge *no-supervisat* per agrupar conjunts d'exemples dels quals només una petita part estan etiquetats. Els algorismes *semi-supervisats* permeten reestimar iterativament els models sense necessitar d'anotar tot el corpus. L'objectiu és accelerar l'aprenentatge de tasques de predicció supervisades en contexts on, a més, es

[3] *Clustering*
[4] *Data Mining*

disposa d'una gran quantitat d'exemples sense valors de sortida definits, que indirectament poden aportar informació de la seva distribució a l'espai d'atributs.

2.2.2 CLASSIFICACIÓ VS. REGRESSIÓ

El següent criteri en el qual es poden agrupar les tècniques d'AA fa referència a la naturalesa del valor que s'ha de predir, segons sigui numèric o simbòlic parlarem de diferents tasques: *regressió*, en el primer cas, i *classificació*, en el segon.

En l'aprenentatge supervisat es parteix d'una col·lecció d'exemples d'entrenament que inclouen els valors de sortida de cada un, i a partir d'aquests exemples el sistema ha d'inferir un model que li permeti predir les sortides d'exemples que no es trobaven en el conjunt d'entrenament. Per tant l'algorisme ha d'obtenir una funció que generalitzi la projecció entre les descripcions dels exemples i el valor de sortida.

Si el valor de sortida d'aquesta funció és un valor numèric, la tasca s'anomena *regressió*, i en predir un valor real permet resoldre tasques quantitatives en les quals es vol estimar una magnitud. Per avaluar la qualitat del comportament d'un sistema d'aquest tipus s'acostuma a calcular l'error quadràtic mitjà (valor central i desviació estàndard) entre el valor predit i el valor real.

Si el valor de sortida és un valor simbòlic (o discret) la tasca s'anomena *classificació*, i en predir una etiqueta permet resoldre tasques qualitatives en les quals es vol determinar la categoria a la qual pertany. Per avaluar la qualitat d'un sistema classificador s'acostuma a calcular el tant per cent de classificacions correctes i incorrectes.

Alguns algorismes estan intrínsecament orientats a obtenir regressions, com les xarxes neuronals, i altres a realitzar classificacions, com els arbres de decisió, però tots ells presenten suficient flexibilitat com per utilitzar-se en les dues tasques. La transformació d'una regressió en una classificació binaria és especialment senzilla, només cal definir un llindar amb el qual comparar el valor numèric; si hi està per sobre pertany a una classe si hi està per sota pertany a una altra, això és exactament el que fan els classificadors lineals **[6.6 Classificadors Lineals]** i alguns sistemes connexionistes **[6.7 Sistemes Connexionistes]**.

2.2.3 INDUCCIÓ D'ANALITZADORS LINGÜÍSTICS

Els classificadors supervisats són l'eina fonamental a partir dels quals desenvolupar analitzadors lingüístics que puguin ser entrenats a partir de dades anotades. Tot i l'aparent distància conceptual entre classificar un objecte i analitzar les relacions i estructures d'una oració, les tasques no són gaire diferents.

El procediment habitual consisteix a entendre el text com un flux de dades, i per tant n'hi ha prou amb definir un context local, conegut com a *finestra*, que es desplaçarà al llarg de l'oració per cobrir tots els seus elements. Per cada una de les seves posicions pot definir-se

una descripció que codifiqui tant les característiques de l'element objectiu com les característiques dels elements i les relacions presents en el context. D'aquesta manera s'ha transformat un problema *dinàmic*, el flux textual, en un problema *estàtic* format per una sèrie d'elements independents contextualitzats.

The	white	[dog]	runs	on	…
the	white	[dog]	run	on	…
Det	Adj	[Noun]	Verb	Prep	…
sing	-	[sing]	sing	-	…
-	-	[-]	3rd	-	…
.	.	[.]	.	.	.
.	.	[.]	.	.	.
.	.	[.]	.	.	.

FIG. 3: Exemple de finestra de trets lingüístics

Arribats a aquest punt és senzill aplicar els classificadors supervisats sobre cada un d'aquests elements. Per exemple, si s'utilitzen classificadors binaris, que discriminen entre dues classes (A i no-A), s'obtenen detectors que poden identificar la presència de determinats objectes lingüístics; també es poden utilitzar classificadors estàndard per diferenciar les diferents sub-categories dels objectes detectats. La mateixa tècnica es pot utilitzar per detectar i classificar constituents i relacions en nivells lingüístics superiors, i unir els resultats parcials per obtenir l'anàlisi conjunta.

El que és important és que la majoria de tasques d'anàlisi lingüística poden descomposar-se com un conjunt de tasques de classificació i desambiguació, tasques parcials que poden resoldre's mitjançant classificadors supervisats induïts a partir d'exemples anotats, i per tant qualsevol tasca lingüística és *potencialment* susceptible de ser apresa a partir d'exemples per algorismes d'AA.

2.2.4 CARACTERÍSTIQUES DE LES TASQUES LINGÜÍSTIQUES

Els sistemes d'AA de base estadística, i per extensió la majoria d'algorismes d'AA, ofereixen molts avantatges en ser aplicats a tasques de PLN. Per un costat, presenten una gran cobertura i robustesa, cosa que els permet tractar texts reals, per un altre costat, són fàcilment reutilitzables a altres idiomes o dominis, i finalment, el seu temps de desenvolupament és més curt que en el cas dels sistemes basats en coneixement, on aquest s'explicita manualment.

Però la naturalesa del llenguatge i les característiques de les tasques lingüístiques presenten algunes peculiaritats que suposen una dificultat afegida respecte als problemes d'altres dominis. Alguns d'aquests reptes **[Daelemans, 2005]** són:

- El Problema de l'Escassetat de Dades[5]: sovint no es disposa de prou dades com per estimar raonablement les probabilitats dels esdeveniments poc freqüents.
- El Problema de la Rellevància[6]: és difícil determinar *a priori* la importància o rellevància d'un determinat atribut o esdeveniment a l'hora de resoldre una tasca.
- El Problema de la Interpretació[7]: la majoria de models estadístics no permeten obtenir explicacions o informació sobre els criteris utilitzats per la resolució de la tasca.

A més, la formalització de qualsevol problema lingüístic, la representació de les seves dades i, fins i tot, la tria i ajust de l'algorisme d'AA introdueixen biaixos que condicionen els resultats. Els biaixos són essencialment restriccions que limiten la cerca a determinades àrees de l'espai de solucions, però no sempre són intencionats, ja que aquests biaixos poden tenir diferents orígens, tant tècnics, com metodològics o lingüístics.

En alguns casos representen un avantatge ja que permeten reduir l'espai de cerca i, per tant, acceleren l'aprenentatge, descartant àrees on el nostre coneixement lingüístic ens permet saber que no contenen solucions correctes. Però habitualment els biaixos s'han de considerar una limitació, ja que en reduir l'espai de cerca també redueixen la probabilitat de trobar models òptims. Aquests biaixos poden venir tant per motius tècnics, la representació dels atributs que descriuen els exemples o les heurístiques d'un algorisme a l'hora d'augmentar el seu rendiment, com per motius més profunds, com les limitacions en l'expressivitat dels models o principis intrínsecs en el funcionament de l'algorisme.

En relació als biaixos introduïts pels algorismes d'AA és important saber que existeixen dues aproximacions a l'hora de definir els models **[Raykar, 2005; Raykar, 2007]**, la *paramètrica* i la *no-paramètrica*:

> **Aproximació *paramètrica*:** Aquesta aproximació assumeix que el model és representable mitjançant una funció paramètrica, una família de funcions, i considera que l'aprenentatge consisteix en estimar els paràmetres que defineixen el model òptim que minimitza l'error de classificació. En aquesta aproximació el coneixement és emmagatzemat en els paràmetres que defineixen el model i, per tant, una vegada obtinguts i l'essència de les dades capturada, els exemples poden ser oblidats. Aquesta aproximació és més eficient ja que els models induïts acostumen a requerir menys espai que les dades, però pot portar a inferències errònies en aquells casos on la forma del model no sigui coneguda *a priori*.
>
> **Aproximació *no-paramètrica*:** Aquesta aproximació no fa cap assumpció respecte la forma de la funció subjacent que caracteritza el model, deixant que les dades la reflecteixin implícitament. El model està constituït per les pròpies dades d'entrenament que permeten, per analogia, classificar nous casos. Aquesta

[5] *Sparse Data Problem*
[6] *Relevance Problem*
[7] *Interpretation Problem*

aproximació dóna millors resultats que els mètodes paramètrics, ja que pot modelar més fidelment qualsevol funció, però presenta el cost d'haver d'emmagatzemar les dades d'entrenament durant la classificació per poder fer la inferència, cosa que té efectes gens menyspreables en la quantitat de recursos necessaris, tant computacionals com de.

Cal aclarir que el fet que un algorisme pertanyi a l'aproximació *no-paramètrica* no significa que no tingui paràmetres, sinó que el model o funció subjacent del problema d'aprenentatge no pot codificar-se amb un nombre finit de paràmetres, ja que la quantitat de paràmetres necessaris augmenta amb la quantitat d'exemples utilitzats durant l'entrenament.

2.3 PARADIGMA *BATCH*

Fins ara s'han presentat els algorismes d'AA des del punt de vista abstracte, sense fer referència ni a l'arquitectura ni a la seqüència dels processos necessària per entrenar un classificador i aplicar-lo a noves dades. En aquesta secció es descriuen aquests aspectes que defineixen el paradigma dominant en l'AA: l'aprenentatge en mode *batch*.

2.3.1 GENERAL

El paradigma *batch* d'AA, també conegut com a *off-line*, assumeix que el desenvolupament de qualsevol classificador supervisat es realitza seguint tres fases necessàriament ordenades en el temps. En primer lloc es recopila una mostra representativa de les dades i s'anota amb les etiquetes que es vol que aprengui el classificador. En segon lloc es mostren al classificador els exemples etiquetats de manera que pugui analitzar-los globalment tantes vegades com sigui necessari. Finalment, una vegada induït el model, el classificador pot ser utilitzat per classificar nous exemples que no hagi vist anteriorment.

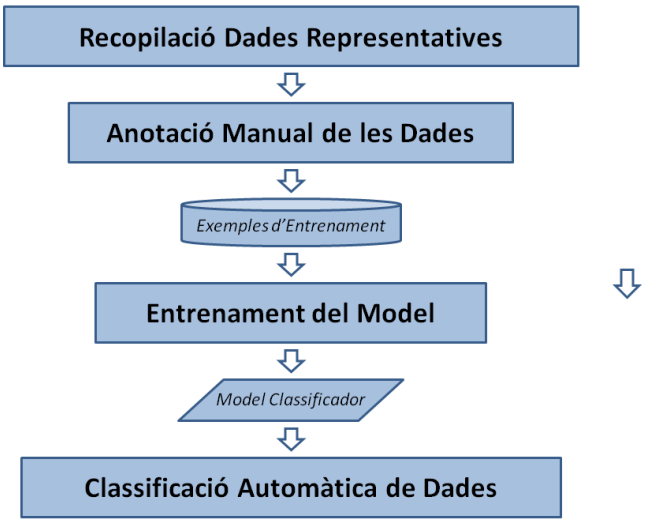

FIG. 4: Dades -> Entrenament -> Anotació

Històricament, aquest ha estat el paradigma dominant utilitzat per la comunitat de l'AA i que s'ha considerat com el procés estàndard per induir models predictors. Fins al punt que s'assumeixen i s'accepten amb més o menys inconsciència les importants restriccions que comporta.

En primer lloc assumeix que cal tenir totes les dades anotades abans de començar l'entrenament, és a dir, que el corpus s'ha de construir *a priori*. En segon lloc assumeix que l'entrenament és un procés acotat en el temps, que s'inicia i finalitza en moments determinats, i que durant aquest interval el model no és funcional. Finalment, assumeix que al finalitzar l'entrenament, s'ha obtingut un model òptim i estàtic que romandrà invariable durant la seva utilització futura.

Més enllà de les conseqüències pràctiques d'aquestes restriccions, que s'expliquen més endavant **[2.4 Limitacions *Batch*]**, el que és sorprenent és la relativa facilitat amb la qual s'ha assumit que aquestes restriccions eren insalvables o amb la qual directament han estat ignorades.

2.3.2 FORMALITZACIÓ

Abans de continuar és important oferir una descripció més formal del funcionament d'aquesta família d'algorismes. L'entrenament d'un algorisme d'AA supervisat, o la inducció d'un classificador, pot definir-se en els següents termes.

Partim d'un conjunt d'exemples d'aprenentatge que es descriuen mitjançant una sèrie de parells atribut-valor i una etiqueta de classe, és a dir, un conjunt d'exemples amb la forma $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ que relaciona els exemples $\boldsymbol{x}_i$ amb la variable y_i, seguint una funció desconeguda $y = f(\boldsymbol{x})$. Cada un dels exemples és un vector amb la forma $\langle x_{i,1}, x_{i,2}, \ldots, x_{i,m} \rangle$ que descriu les seves característiques mitjançant valors reals o discrets, i on les classes y pertanyen a un conjunt finit de classes o categories $\{1, \ldots, \mathrm{K}\}$.

Donat aquest conjunt de dades d'aprenentatge $T = \{ (x_i, y_i) \}_{i=1}^{n}$, l'objectiu de l'algorisme és induir una hipòtesi *h*, aproximació de la funció desconeguda *f*, que sigui coherent amb els exemples. Normalment la obtenció d'*h* dintre de l'espai de funcions possibles es realitza mitjançant tècniques de minimització d'alguna mesura de discrepància (error, distància, ...) promitjada per tots els elements del conjunt d'entrenament.

Una vegada finalitzat l'entrenament, és a dir, obtingut el model de la funció *h* que minimitza la discrepància respecte els exemples, es pot aplicar aquesta funció a exemples sense etiquetar amb l'objectiu de classificar-los amb més o menys encert segons l'èxit que hagi tingut el model a l'hora de generalitzar el coneixement implícit en les dades d'entrenament.

2.4 LIMITACIONS *BATCH*

Com s'ha avançat al punt anterior, l'aprenentatge en mode *batch* presenta importants limitacions que, com s'afirma a **[Màrquez, 2001]**, han arribat a qüestionar la seva aplicabilitat a tasques reals, entre elles:

- L'elevat cost d'adquisició dels corpus etiquetats per l'aprenentatge
- El salt computacional que suposa passar de treballar amb petits corpus a fer-ho en un domini obert
- L'alta dependència dels classificadors obtinguts respecte al domini d'aprenentatge

Als apartats següents es desenvolupen aquestes limitacions englobades en termes de limitacions *econòmiques*, limitacions *tècniques* i limitacions *metodològiques*.

2.4.1 ECONÒMIQUES: COST D'ANOTACIÓ

Actualment, és relativament fàcil aconseguir grans conjunts de dades que continguin milions d'exemples d'entrenament, tant de corpus editorials com de la pròpia web, però el problema real no és la obtenció de dades en brut, sinó la seva anotació. És conegut que un dels primers problemes que es plantegen a l'hora d'intentar resoldre una tasca de PLN mitjançant algorismes d'AA és la disponibilitat d'un corpus d'entrenament per a la tasca corresponent.

Donat que l'objectiu últim és desenvolupar models que realitzin *correctament* tasques de processament textual o d'anotació lingüística, i que *correctament* acostuma a voler dir el més semblant possible a com ho faria un expert humà, cal que els exemples d'entrenament siguin anotats per persones, a ser possible, expertes. Això suposa un cost econòmic important, ja que no només es tracta de personal qualificat sinó que el procés d'anotació acostuma a requerir molt de temps, ja que és una tasca laboriosa que no pot fer-se durant llargues sessions ja que el cansament multiplica la taxa d'errors. Aquest és el problema conegut com *el coll d'ampolla de l'adquisició de coneixement*[8] **[Gale et al., 1992]**.

El fet que el *corpus de referència*[9], constituït pels exemples d'avaluació, hagi d'estar etiquetat manualment per persones, implica que aconseguir un corpus amb desenes de milers o milions d'exemples té un cost molt considerable. A **[Ng 1997b]** s'estima que el cost d'etiquetació manual per un corpus d'aprenentatge per la tasca de Desambiguació Semàntica de Sentits[10] (DSS) és d'aproximadament 16 persones/any; evidentment altres tasques de nivells lingüístics superiors suposen costs encara més elevats.

[8] *Knowledge acquisition bottleneck*
[9] *Gold Standard*
[10] *Word Sense Desambiguation (WSD)*

2.4.2 TÈCNIQUES: MANCA D'ESCALABILITAT

Però encara que desaparegués el problema associat al cost d'anotació, i es pogués disposar de corpus d'entrenament il·limitats perfectament anotats, existeix una limitació tecnològica que dificultaria l'entrenament d'algorismes d'aprenentatge *batch* amb aquestes grans quantitats de dades: la seva manca d'escalabilitat tant en *espai* com en *temps*.

Recordem que en PLN és habitual tractar conjunts de dades massius[11] amb milions d'exemples d'entrenament, amb representacions de milers d'atributs i tasques de classificació amb centenars de classes, per tant els recursos necessaris per a l'emmagatzemament i processament d'aquests volums d'informació no són gens menyspreables.

Els problemes d'escalabilitat es poden dividir segons el domini on es produeixin, és a dir, *escalabilitat en l'espai*, si el problema és un creixement desproporcionat dels recursos de memòria necessaris, i *escalabilitat en el temps*, si el problema és un creixement desproporcionat en els recursos computacionals.

Molts dels algorismes d'AA utilitzats habitualment no són linealment escalables, els costos computacionals d'entrenament poden ser $O(N^2)$ o $O(N^3)$ i els seus requeriments de memòria també poden ser $O(N^2)$. Aquesta manca d'escalabilitat suposa una important limitació a l'hora d'utilitzar conjunts de dades massius i, per tant, la majoria d'implementacions actuals només poden tractar alguns milions d'exemples.

	Training (N examples)	Prediction (at N points)	Choosing parameters
Kernel regression	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Gaussian processes	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$
SVM	$\mathcal{O}(N_{sv}^3)$	$\mathcal{O}(N_{sv}N)$	$\mathcal{O}(N_{sv}^3)$
Ranking	$\mathcal{O}(N^2)$		
KDE		$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
Laplacian eigenmaps	$\mathcal{O}(N^3)$		
Kernel PCA	$\mathcal{O}(N^3)$		

TAULA 1: Comparativa de l'escalabilitat de diferents algorismes, per entrenament, classificació o ajust. Font [Raykar, 2005].

ESCALABILITAT EN L'ESPAI

Els problemes d'escalabilitat en l'espai poden produir-se en dos punts. En primer lloc, en la memòria necessària per emmagatzemar les dades d'entrenament, aquest problema pot ser significatiu en aquells algorismes que requereixin d'un accés intensiu i repetit a les dades d'entrenament. I, en segon lloc, en la memòria necessària per emmagatzemar el model final

[11] Els conjunts de dades massius inclouen el que a **[Raykar, 2005]** s'anomena '*tall data*', conjunts de dades amb milions d'exemples, '*fat data*' conjunts de dades amb gran nombre d'atributs, i '*tall fat data*', aquells conjunts de dades on el nombre d'exemples és similar a la quantitat de trets que els defineixen.

o els seus estadis entremitjos, aquest problema pren especial importància en alguns algorismes estadístics que requereixen estimar co-ocurrències.

Alguns autors **[Witten & Frank, 2005]** han destacat les dificultats associades a tenir corpus d'entrenament més grans que la memòria principal. Tot i que el problema és tècnicament soluble mitjançant indexacions i paginacions en memòria secundària, aquesta solució simplement intercanvia memòria per velocitat amb la qual cosa el problema d'escalabilitat es trasllada d'un domini a un altre. Però el veritable problema és degut, no a la mida de les dades d'entrenament, sinó a la mida dels models generats, que requereixen estar ubicats en la memòria principal, i que en alguns casos poden créixer linealment o, fins i tot, quadràticament amb el nombre d'exemples.

ESCALABILITAT EN EL TEMPS

Els problemes d'escalabilitat en el temps també poden produir-se en més d'un punt. En primer lloc, en el temps necessari per entrenar-se, és a dir, per induir el model a partir dels exemples. En segon lloc, en el temps necessari per classificar una instància, especialment en aquells casos on el model representacional és complex i no s'hi pot accedir de manera òptima. Finalment, en alguns casos la manca d'escalabilitat es produeix en seleccionar els paràmetres o atributs òptims.

Si el temps d'entrenament no creix, com a màxim, linealment amb el nombre d'exemples, s'arriba ràpidament a un límit pràctic en la mida màxima dels corpus. Si el temps de classificació augmenta amb el nombre d'exemples, també s'arriba a un límit pràctic en la complexitat del model induït, com el coll d'ampolla que presenten els algorismes no paramètrics a causa de la complexitat computacional de la classificació basada en memòria **[Raykar, 2005]**. Afortunadament, existeixen diferents algorismes que escalen linealment tant amb el nombre d'exemples com amb el nombre d'atributs.

2.4.3 METODOLÒGIQUES: APRENENTATGE SEGREGAT EN DUES FASES

Finalment, fins i tot si, a més de grans quantitats de dades anotades, tinguéssim algorismes perfectament escalables capaços de processar-les sense problemes, l'arquitectura *batch* continuaria presentant una limitació metodològica: la separació de la fase d'entrenament de la fase d'explotació. Aquest paradigma d'AA permet induir models que s'adapten a les dades d'exemples, però els classificadors resultants, una vegada entrenats, es tornen estàtics, el seu coneixement es congela i l'aprenentatge s'atura.

Informalment, es pot considerar que un sistema és intel·ligent quan és capaç d'aprendre a partir de les dades i per tant té la capacitat d'adaptar el seu comportament. Sota aquest punt de vista el paradigma *batch* ofereix sistemes que són intel·ligents durant la fase d'entrenament però que deixen de ser-ho durant la seva explotació. En altres paraules, una vegada entrenats, els sistemes es tornen fixos i actuen com qualsevol sistema basat en heurístiques. Per tant no poden actualitzar els seus models per adaptar-se a variacions en les

dades, ni aprofitar l'experiència que suposa accedir a una quantitat de dades molt més gran que les que van veure durant l'entrenament.

La principal conseqüència d'aquesta limitació és el que es coneix com el problema de *fora de domini*[12], consistent en una pèrdua del rendiment dels classificadors en ser aplicats a dades qualitativament diferents a les de l'entrenament. Aquest problema, àmpliament reconegut per la comunitat de l'AA, té el seu origen en la manca d'adaptabilitat dels classificadors *batch*, i en la bibliografia sorgeix repetidament la necessitat de la *transportabilitat* dels classificadors entrenats. A **[Màrquez, 2001]**, un estudi comparatiu d'algorismes aplicats a la tasca de Desambiguació Semàntica de Sentits, s'afirma que si no es minimitza el problema mitjançant algun procés d'adaptació, la manca de *transportabilitat* dels algorismes és un problema real que pot arribar a qüestionar l'aplicació d'aquestes tècniques.

[12] *Out of domain*

3 CORPUS D'ENTRENAMENT

3.1 INTRODUCCIÓ

Una de les constants en l'evolució de la tecnologia de la informació és el creixement exponencial dels recursos disponibles per unitat de cost. Això es reflecteix en el fet que des de fa més de 30 anys la potència computacional dels ordinadors i la capacitat d'emmagatzemament que ofereixen (primària i secundària) es dupliquen periòdicament.

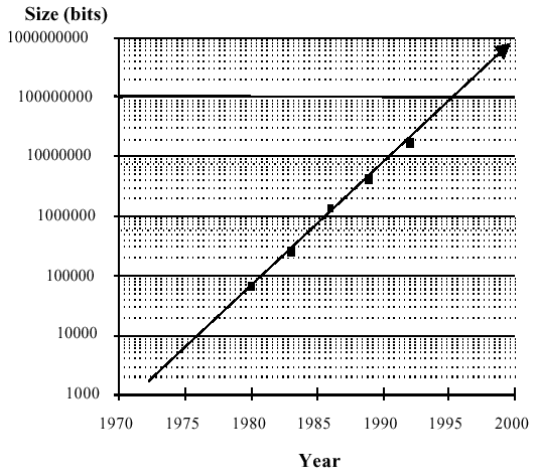

FIG. 5: Creixement de la memòria dels ordinadors per unitat de cost. Font [Mafla 2001]

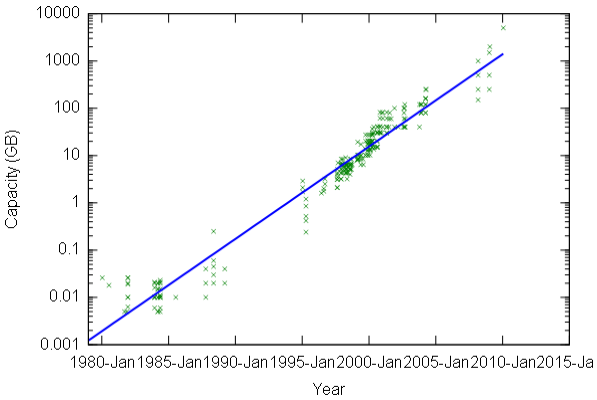

FIG. 6: Augment de la capacitat de disc dels ordinadors en termes absoluts

Per contra, tot i que la mida dels principals corpus anotats ha augmentat amb els anys, el seu creixement recent ha estat molt més reduït, per no dir estancat. A **[Banko & Brill 2001b]** es considera que durant els darrers anys la mida dels corpus utilitzats s'ha mantingut pràcticament constant. La **Taula 2** mostra com malgrat l'existència de corpus de centenars

de milions de paraules, els corpus anotats i revisats manualment es mantenen per sota del milió de paraules, tant en anotacions morfològiques com sintàctiques:

Tipus	Any	Corpus Text	Paraules
		SENSE ANOTAR	
Text	2005	Bank of English (CoBuild)	525 M
Text	2006	Oxford English Corpus	2000 M
		ANOTATS AUTOMÀTICAMENT	
PoS_A	1992	Wall Street Journal corpus (WSJC)	47 M
PoS_A	1994	British National Corpus (BNC)	100 M
PoS_A	2000	Corpus of Contemporary English (COCA)	385 M
PoS_A	2003	American National Corpus (ANC)	22 M
PoS_A	2004	Corpus Referencia Español Actual (CREA)	170 M
		ANOTATS/REVISATS MANUALMENT	
PoS_R	1967	Brown Corpus of Standard American English	1,0 M
PoS_R	1993	Susanne Corpus	0,1 M
$Tree_M$	1995	Penn Treebank	1,0 M
$Tree_M$	1998	International Corpus of English - British (ICE-GB)	1,0 M
$Tree_M$	2005	Bosque Treebank	0,3 M
$Tree_M$	2002	CAST3LB Spanish Treebank	0,1 M

TAULA 2: Mida, en milions de paraules, dels principals corpus segons la metodologia d'anotació: automàtica (A), revisada (R) o manual (M).

Per tant, si tenim en compte que la disponibilitat de textos electrònics és pràcticament il·limitada, el fet que les mides dels corpus anotats no hagin seguit el creixement exponencial del recursos informàtics és un clar indici de l'existència del coll d'ampolla en el procés d'anotació.

En aquest capítol es presenten algunes dades que suggereixen que el cost d'anotació dels corpus és més que un simple problema pràctic, ja que les seves conseqüències (la restricció a corpus petits, la reutilització forçada i la inviabilitat dels corpus *ad-hoc*) condicionen i limiten la recerca realitzada en el camp.

3.2 REUTILITZACIÓ DE CORPUS ESTÀNDARD

L'elevat cost que té l'anotació d'un corpus de mida mitjana, i fins i tot petita, fa que la majoria de departaments i grups de recerca ni es plantegin la possibilitat de desenvolupar un corpus *ad-hoc*, adaptat a les necessitats dels seus projectes i anotat amb criteris propis. I, per tant, amb els anys s'ha anat consolidant la tendència, molt raonable des del punt de vista econòmic, de reutilitzar corpus ja existents que han acabat convertint-se en estàndard de fet de la comunitat.

En primer lloc, aquest condicionant restringeix directament la llibertat d'explorar nous enfocaments i teories lingüístiques. Cal tenir en compte que els corpus estàndard no ho són tant per la seva qualitat consensuada com per la seva simple existència. Els criteris lingüístics utilitzats per anotar-los pertanyen a una disciplina que tot just comença a definir les seves bases, i en la qual no hi ha unanimitat ni en els etiquetaris més simples. Això dificulta que es puguin explorar, i descobrir, nous plantejaments lingüístics, i pressiona subtilment els investigadors per adaptar els seus treballs als punts de vista dels corpus disponibles.

A més, tot i que la utilització de corpus estàndard facilita la recerca més computacional, al permetre la comparació de diferents algorismes i tècniques amb les mateixes dades, també augmenta el risc que la sobrevaloració de les competicions[1] acabi produint algorismes optimitzats per a tasques artificials que no siguin transportables .

En segon lloc cal tenir en compte que alguns d'aquests corpus van ser desenvolupats fa anys i, sense entrar en la seva representativitat lingüística (diferències diacròniques a nivell lèxic), cal ser conscients que les seves mides van estar condicionades per la tecnologia present en el seu moment.

I malgrat que en molts casos pugui semblar que les mides actuals són més que suficients, és possible que es tracti d'una percepció errònia, ja que alguns experiments amb corpus massius han proporcionat resultats molt sorprenents.

3.3 EXPERIMENTS AMB CORPUS MASSIUS

El 2001 i 2002 es van publicar dos articles **[Banko & Brill 2001a; Van den Bosch & Buchoold, 2002]** que qüestionaven el conformisme existent respecte la utilització de corpus de mida estàndard a l'hora d'entrenar algorismes d'AA. Tot i ser plantejats sota dos punts de vista diferents, els resultats convergien en una mateixa conclusió: la utilització de corpus massius podria replantejar la direcció del PLN basat en l'AA.

3.3.1 INDEPENDÈNCIA DE L'ALGORISME

L'existència de tasques de PLN en les quals els exemples poden crear-se sintèticament a partir de text sense anotar permet obtenir corpus d'entrenament a un cost pràcticament nul. Això va permetre a **[Banko & Brill, 2001a]** realitzar diversos experiments per avaluar el comportament de diferents algorismes entrenats amb corpus fins a mil vegades més grans que els habituals.

La tasca de PLN utilitzada és la que es coneix com a *Desambiguació de Conjunts de Confusió*,[2] on l'objectiu és seleccionar l'ús correcte d'una paraula entre diferents alternatives

[1] Com la coneguda *Shared Task* de la CoNLL (*Conference on computational Natural Language Learning*)
[2] *Confusion Set Disambiguation (CSD)*

homòfones, conegudes com a *conjunt de confusió* [3], que acostumen a ser confoses pels parlants. Per generar un corpus d'entrenament per a aquesta tasca, n'hi ha prou amb substituir totes les ocurrències dels elements d'un d'aquests conjunts per un marcador (*****) de manera que, durant l'entrenament, l'algorisme no tingui accés a la paraula contextualitzada, però la conservem per avaluar la classificació posterior. Aquesta mena de problemes permeten obtenir grans quantitats de corpus anotat sense pràcticament cap cost, cosa que permet comprovar els límits dels algorismes.

En treballar amb corpus de mida estàndard és habitual obtenir resultats com els mostrats a la **Fig. 7**, que mostra l'evolució de l'exactitud d'un classificador segons la mida del corpus. Aparentment a partir d'un cert punt la utilització de més exemples d'entrenament no suposa una millora considerable en la seva precisió.

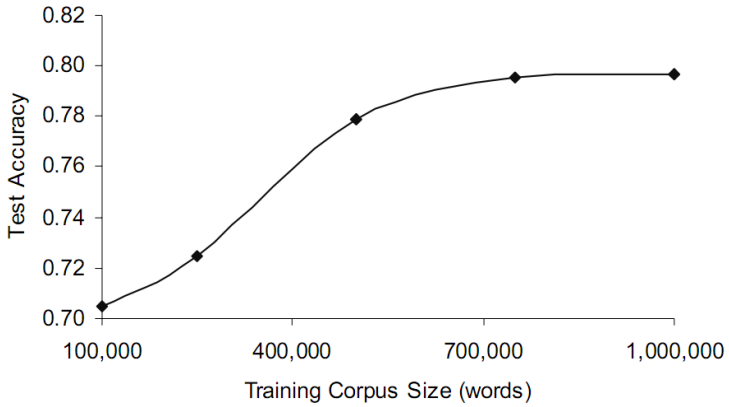

FIG. 7: Evolució típica de l'exactitud d'un classificador segons la mida del corpus d'entrenament. Font [Banko & Brill, 2001b].

La interpretació habitual és que l'algorisme ha arribat al seu màxim, sense tenir en compte que aquesta *saturació* pot estar causada per les restriccions imposades a la complexitat del model per alguns paràmetres del classificador. Per exemple, en els sistemes connexionistes, un increment en el nombre de capes i neurones, eliminaria la saturació i permetria aprofitar corpus més grans; en els arbres de decisió es podria reduir el nivell de poda, i en algorismes basats en memòria es podria augmentar el nombre d'instàncies veïnes.

Però si entrenem diferents algorismes [4] amb corpus de mides 10, 100 i 1000 vegades superior **[Fig. 8]** es comprova com les evolucions no són paral·leles, sinó que diferents classificadors aprenen a diferents ritmes i presenten precisions molt diferents segons la quantitat d'exemples utilitzats. Amb corpus petits, d'unes 200 mil paraules, el *Memory-Based* obté resultats molt superiors (≈83%) a tots els altres, en especial al *Winnow* (≈75%); amb un milió de paraules les diferències s'han reduït i tots es situen en una franja propera (≈83%-86%); amb 10 milions de paraules el *Memory-Based* es superat clarament per tots els altres; i

[3] Exemples d'aquests conjunts de confusió són els següents: {*principle, principal*}, {*then, than*}, {*to, two, too*} i {*weather, whether*}. Font **[Banko & Brill, 2001a]**.
[4] Els algorismes utilitzats són un Memory-Based, un Winnow, un Perceptró i un Naïve Bayes, tot i que a **[Banko & Brill, 2001b]** també es parla d'un arbre de decisió i d'un algorisme basat en transformacions.

amb 100 i 1000 milions de paraules tots ells continuen millorant fins augmentar 10 punts (≈93%-96%).

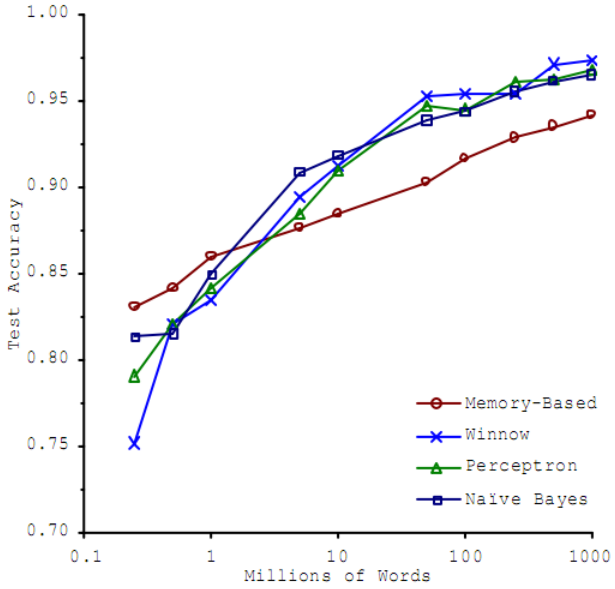

FIG. 8: Evolució de la precisió de diferents classificadors segons la mida del corpus d'entrenament. Font [Banko & Brill, 2001a].

Els resultats mostren que per aquesta tasca s'ha pogut construir un sistema a partir d'un algorisme molt bàsic, que supera els resultats dels millors classificadors actuals simplement entrenant-lo amb més dades. Per això **[Banko & Brill, 2001b]** es pregunta fins a quin punt són vàlides les conclusions de molts articles experimentals que demostren la superioritat d'un algorisme respecte d'altres basant-se en resultats *marginalment* superiors després de ser entrenats amb petits corpus d'un milió de paraules.

Les conclusions de Banko i Brill són clares: les mides dels corpus utilitzats habitualment estan limitant els resultats dels classificadors i falsejant les avaluacions comparatives dels algorismes. A més, donat que, segons l'avaluació realitzada amb un corpus d'entrenament d'un milió de paraules, el millor algorisme és superat clarament pel pitjor quan aquest és entrenat amb un corpus de 10 milions de paraules, conclou que potser val la pena invertir en l'anotació de grans corpus en comptes de continuar invertint en desenvolupar algorismes que aconsegueixen millores marginals.

3.3.2 INDEPENDÈNCIA DEL NIVELL D'ANOTACIÓ

Com s'ha comentat, una de les característiques de les dades lingüístiques utilitzades per entrenar els algorismes d'AA és la seva alta dimensionalitat, que comporta que determinats trets només apareguin en una fracció dels exemples tot i que quan apareixen poden ser determinants per classificar-los correctament **[Dredze et al., 2008]**. A més, aquesta alta dimensionalitat afavoreix l'absència de moltes combinacions de trets, i per tant dóna lloc a

una baixa densitat d'exemples, conegut com a *data sparseness*[5], que requereix grans corpus d'entrenament.

Per tant la utilització de corpus massius suposa un altre efecte beneficiós: la reducció dels efectes del *data sparseness*, cosa que permet reduir la granularitat dels trets i per tant obtenir models més fins i precisos. Seguint aquesta argumentació a **[Van den Bosch & Buchoold, 2002]** van portar a terme alguns experiments d'anàlisi superficial a partir de la sorprenent premissa que les etiquetes de categoria gramatical[6] no són més que categories intermèdies fruit de la necessitat històrica de reduir els efectes de la dispersió de dades, i que, a partir de corpus de certes mides, poden ser innecessàries.

Durant els diferents experiments van avaluar el funcionament d'un classificador basat en memòria entrenat amb corpus amb diferents nivells d'anotacions: un *gold standard*[7] amb la categoria gramatical original, un corpus amb només formes lèxiques, un altre de similar però agrupant les formes poc freqüents, i un darrer que combinava la forma lèxica i la categoria gramatical. En tots quatre casos es va incrementar progressivament la quantitat de dades d'entrenament amb l'objectiu d'obtenir les següents corbes:

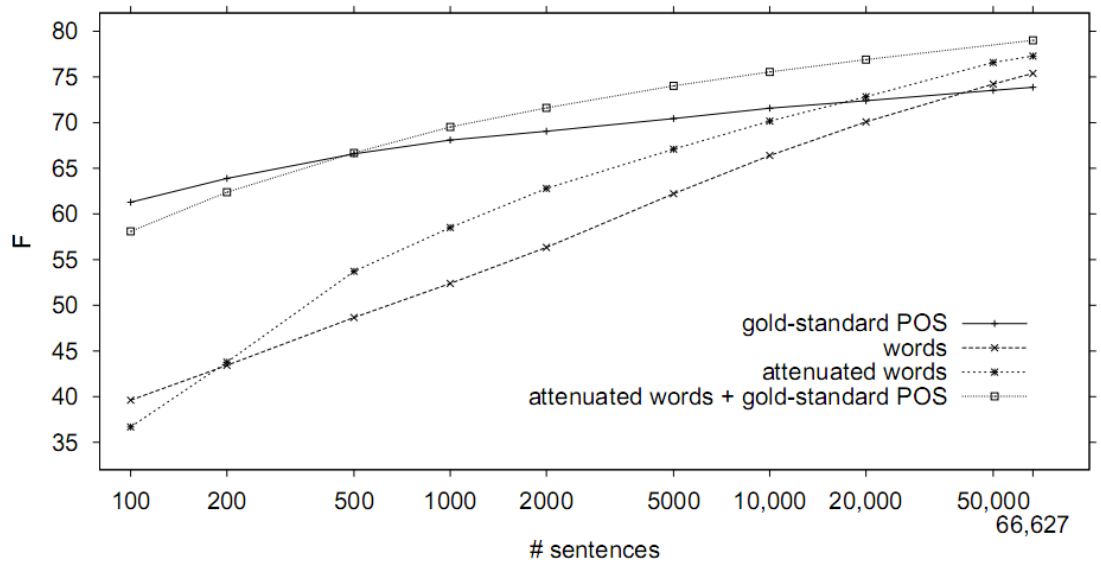

FIG. 9: Evolució del F-Score per corpus d'entrenament amb diferents nivells d'anotació. Font [Van den Bosch & Buchoold, 2002].

La **Fig. 9** mostra l'evolució del *F-Score*[8] **[8.5.1 Mètriques Estàndard]** per cada un dels corpus anotats, s'observa com fins a uns pocs milers d'oracions el *gold standard* que només inclou categories gramaticals obté els millors resultats, però a partir de les vint mil oracions tendeix a estabilitzar-se i és superat per altres anotacions més pobres a mesura que disminueixen els efectes de la dispersió de dades a nivell lèxic. De la mateixa manera que les

[5] Escassetat de dades
[6] *Part-of-Speech (PoS)*
[7] Corpus de referència
[8] Mesura-F

corbes de la **Fig. 8** mostren com la quantitat d'exemples d'entrenament poden suplir les diferències inicials entre algorismes d'aprenentatge, les corbes de la **Fig. 9** mostren com també poden suplir diferències en la riquesa o granularitat de les anotacions del corpus d'entrenament.

3.4 REPTES DELS CORPUS MASSIUS

El fet que els algorismes classificadors continuïn millorant la seva precisió al ser entrenats amb corpus 100 o 1000 vegades més grans, i que la quantitat de les dades pugui suplir la qualitat de l'anotació, és una dada engrescadora, que semblaria la solució a moltes tasques de PLN. Lamentablement la utilització de corpus massius d'entrenament comporta dificultats, tant de desenvolupament com de processament, que abans caldrà resoldre si es vol seguir aquesta direcció. Una vegada resolts, com s'apunta a **[Banko & Brill, 2001a]**, s'hauria de plantejar la necessitat de reequilibrar les inversions en recerca dedicades a nous algorismes i les dedicades al desenvolupament de grans corpus.

A continuació es presenten algunes de les possibilitats existents a l'hora de reduir aquestes dificultats, tant a l'hora de reduir els costos de desenvolupament de corpus anotats, com a l'hora de buscar dreceres que permetin el processament i aprenentatge a partir d'aquestes dades.

3.4.1 DESENVOLUPAMENT DE CORPUS MASSIUS

Lamentablement, les tasques de PLN que permeten obtenir anotacions de franc són poques, la immensa majoria requereix corpus d'entrenament que han de ser anotats manualment i que, per tant, representen costos importants; si parlem de corpus massius les inversions necessàries són simplement implantejables.

A **[Banko & Brill, 2001b]** es realitza una estimació del cost d'un hipotètic corpus massiu de 1.000 milions de paraules etiquetat únicament amb etiquetes gramaticals. Suposant que l'anotació consistís en la revisió d'una etiquetació automàtica amb una precisió típica del 95%, la tasca requeriria 1,5 milions d'hores o el que és el mateix, unes 750 persones/any. Fins i tot suposant un cost per hora corresponent al de mà d'obra no qualificada ($10/h) el cost total superaria els 15 milions de dòlars.

Sembla doncs que la viabilitat del paradigma de l'AA supervisat depèn en gran part de la solució del problema d'adquisició de coneixement, o en terminologia de PLN, del problema del cost d'anotació. Afortunadament, existeixen tècniques que poden minimitzar aquest cost, gràcies a la reducció dels dos aspectes que afecten directament al temps necessari per anotar un corpus.

SIMPLIFICAR LA TASCA D'ANOTACIÓ

Històricament, l'anotació de corpus ha estat un procés manual que ha consumit una gran quantitat de temps. Per això, era previsible que evolucionés en la línia de simplificar al

màxim la intervenció de l'expert, permetent que processos automàtics assumissin tanta feina com fos raonable. Des d'aquest punt de vista, la tasca d'anotació pot classificar-se, seguint l'evolució que ha experimentat amb els anys, segons el seu grau d'intervenció humana:

I. **Anotació:** L'anotació directa suposa introduir manualment les etiquetes de tots els exemples del text. Inicialment aquest procés es feia editant directament el text i posteriorment mitjançant entorns gràfics que reduïen la utilització del teclat.

II. **Desambiguació:** Si s'introdueixen processos automàtics que marquen els exemples amb un subconjunt, més o menys reduït, de possibles etiquetes, la feina de l'expert consisteix a utilitzar el seu coneixement per desambiguar entre les propostes fetes pel sistema automàtic.

III. **Revisió:** El pas següent és incorporar un sistema automàtic de desambiguació, però com la seva precisió no és absoluta, cal que l'expert revisi les etiquetes assignades i corregeixi les que consideri errònies.

IV. **Supervisió:** Arribats a aquest punt, per reduir encara més la feina de l'expert, és necessari introduir un sistema automàtic de detecció d'errors. O com a mínim algun sistema de priorització que permeti a l'expert centrar-se en la revisió dels casos més dubtosos. Aquest plantejament permetria no haver de revisar la totalitat de les etiquetes assignades com en el nivell anterior.

Per tant, el que genèricament es considera "anotació" pot referir-se a tasques lleugerament diferents: *anotació*, en sentit estricte, *desambiguació*, *revisió* i *supervisió*. En l'actualitat, la majoria d'eines d'anotació són simples interfícies gràfiques que permeten l'anotació manual de petits corpus, nivell I. En projectes de més abast **[Martí et al., 2008]**, es combinen sistemes d'anotació automàtica, en mode *batch*, amb interfícies gràfiques que faciliten la revisió de l'anotació, nivell III.

Però fins i tot aquests sistemes més avançats continuen necessitant que l'expert revisi la totalitat de les etiquetes, cosa que suposa una feina monòtona que predisposa a l'aparició d'errors i una feina ineficient que desaprofita el temps de l'expert fent-li revisar casos trivials on difícilment hi ha errors. Aquest raonament obre la porta a la segona línia d'atac a l'hora de reduir el cost d'anotació: només fer les anotacions que siguin necessàries.

MINIMITZAR LA QUANTITAT D'ANOTACIONS

Si el coneixement de l'expert és un recurs valuós, cal utilitzar-lo eficientment en aquells casos on el benefici obtingut sigui més alt. Per tant, a l'hora d'aconseguir un corpus d'entrenament anotat manualment amb la menor taxa d'error possible, sembla lògic centrar-se en aquells casos on més probablement l'anotació automàtica sigui errònia.

A **[Banko & Brill, 2001b]** s'estima que una tecnologia que permetés revisar únicament els casos dubtosos reduiria el cost de creació del corpus de 1.000 milions de paraules **[3.4.1**

Desenvolupament de Corpus Massius] dels 15 milions de dòlars al voltant de 200 mil dòlars, és a dir, en un factor de més de 50. Així doncs, pot ser interessant aprofitar el coneixement intern del propi classificador automàtic per orientar l'expert a l'hora de decidir quines etiquetes requereixen una revisió més profunda, quines en tenen prou amb una revisió superficial i quines altres poden simplement acceptar-se sense revisar.

Més enllà d'aquestes estratègies aplicables a l'actual paradigma d'anotació, pot ser interessant qüestionar una assumpció relacionada amb la manera com actualment s'anoten els corpus, per si existís alguna manera més eficients d'explotar-los. Actualment els corpus són anotats de manera *homogènia*, és a dir, la totalitat del corpus és anotada amb la mateixa densitat d'informació: els mateixos nivells lingüístics i la mateixa granularitat en els trets corresponents. Si es té en compte que el cost d'una anotació és proporcional tant a l'extensió del corpus com a la profunditat de l'anotació, sorgeix el dubte de si l'anotació homogènia pot estar malbaratant recursos.

En certa manera, el paradigma *batch* sembla "suggerir" una anotació homogènia de corpus. Potser els algorismes incrementals **[4.5.2 Entrenament *Pedagògic*]** permetrien obrir la porta a la utilització d'anotacions heterogènies. Una part del corpus podria estar anotada superficialment amb una baixa granularitat, altres parts podrien incloure anotacions més riques que permetessin resoldre els casos ambigus amb l'anotació superficial, i altres parts podrien estar formades per texts especialitzats que permetessin adaptar els models a diferents dominis. Si això fos possible també ho seria optimitzar la utilització de recursos i estalviar-se determinats nivells d'anotació en diferents seccions del corpus.

3.4.2 PROCESSAMENT DE CORPUS MASSIUS

Com ja s'ha dit anteriorment, l'entrenament de sistemes d'AA amb corpus massius presenta, a més, algunes dificultats tècniques. Per això s'han proposat diferents alternatives per poder aprofitar el coneixement present en aquests corpus, segons **[Rykar, 2007]** diversos enfocaments possibles: mostrejar el corpus, optimitzar l'algorisme *batch*, combinar diversos classificadors o utilitzar algorismes incrementals.

MOSTREIG DEL CORPUS

La manera més senzilla d'utilitzar un corpus massiu per entrenar un algorisme que no té prou recursos per processar-lo íntegrament, és seleccionar una mostra representativa del corpus original. Per determinades tasques pot ser suficient entrenar l'algorisme amb un subconjunt dels exemples disponibles, i experimentalment determinar si la utilització de més dades permet o no millorar el seu rendiment.

Aquest mostreig[9] pot fer-se aleatòriament o mitjançant estratègies més sofisticades que intenten obtenir mostres distribucionalment equilibrades o detectar aquells exemples que són més representatius. Com s'explica a **[Rykar, 2007]**, aquests enfocament permeten

[9] *Subsampling*

inferir una funció exacte de classificació a partir d'un model de dades aproximat. L'inconvenient d'aquesta estratègia és que no és clar que s'estigui aprofitant tot el coneixement del corpus massiu, ja que no hi ha garanties de que el mostreig no hagi deixat fora exemples valuosos però poc freqüents.

OPTIMITZACIÓ D'ALGORISMES BATCH

Un altre enfocament consisteix a buscar estratègies que permetin reduir la càrrega computacional de l'algorisme a l'hora de processar la totalitat del corpus. Aquestes estratègies poden consistir en una optimització directa de l'algorisme (suposant que la seva implementació fos sub-optima), en la utilització d'estructures de memòria més eficients, o en el reaprofitament de càlculs previs mitjançant *caches*.

Una altre estratègia per reduir cost computacional és el desenvolupament d'aproximacions numèriques que siguin quasi-equivalents. A **[Rykar, 2007]** es presenta un model aproximat del *Matrix Vector Product* (*MPV*), una operació utilitzada en el nucli de molts algorismes i que contribueix a la seva complexitat quadràtica $O(N^2)$. Com diu l'autor, això permet inferir funcions aproximades a dades exactes.

Finalment, a l'hora de reduir la càrrega computacional d'un algorisme pot distribuir-se entre diferents processadors, és el que es coneix com a paral·lelització. Aquesta estratègia consisteix a dividir el problema en tasques no dependents que puguin ser realitzades simultàniament per diferents processadors. Per aconseguir-ho cal que l'algorisme sigui paral·lelitzable però, tot i que alguns ho són (*K-NN*, *Decision Trees*, ...), la majoria no permeten millores importants.

L'inconvenient d'aquestes estratègies és tant que no sempre són factibles com que, encara que ho siguin, només permet reduir els recursos de forma lineal.

COMBINACIÓ DE CLASSIFICADORS

Un altre enfocament interessant és la utilització de combinacions de classificadors. Aquests sistemes utilitzen diferent tècniques de votació **[7.3 Sistemes de Votació]** per combinar en un únic resultat les prediccions fetes per diversos classificadors. Això permet dividir el corpus en fragments i utilitzar cada un d'ells per entrenar un classificador diferent.

D'aquesta manera es pot aprofitar tota la informació disponible al corpus, cada classificador extreu una part del coneixement, i aquest és recombinat en un sistema que es comporta com un únic classificador però amb un comportament millor que qualsevol dels classificadors individuals.

L'avantatge d'aquest sistema és que és aplicable a qualsevol tipus de classificador, tot i que els recursos necessaris, en temps i espai, són proporcionals al nombre de classificadors combinats.

ALGORISMES INCREMENTALS

Finalment, existeix un darrer enfocament basat en la utilització d'algorismes incrementals. Aquests algorismes construeixen els seus models a partir d'una sèrie d'actualitzacions progressives a mida que van processant les dades, per tant només necessiten accedir a un exemple cada vegada. A més, molts dels algorismes incrementals utilitzen models que són clarament escalables, uns creixen linealment amb el nombre d'exemples i altres utilitzen models de mida constant.

Els algorismes incrementals poden resoldre eficientment el processament de corpus massius, però a més ofereixen altres avantatges, com la seva velocitat que els permet ser integrats en entorns interactius, que s'explicaran més endavant.

4 APRENENTATGE INCREMENTAL

4.1 INTRODUCCIÓ

Com s'explica a **[Schlimmer & Fisher, 1986]**, la majoria de sistemes d'inducció són no-incrementals, és a dir, necessiten que tots els exemples sobre els quals han d'aprendre estiguin presents des del primer moment. Per contra els sistemes incrementals reben els exemples al llarg del temps, induint els models de manera progressiva a mesura que analitzen els exemples d'un en un.

L'Aprenentatge Automàtic Incremental[1] (AAI) és un paradigma contraposat a l'AA en mode *batch* que ofereix diversos avantatges especialment interessants en el desenvolupament d'aplicacions de PLN. La motivació per a aquests sistemes neix de la percepció que si els sistemes d'aprenentatge han de tractar amb una gran quantitat d'observacions, els sistemes no-incrementals no són computacionalment viables. En altres paraules, el principal benefici de la inducció incremental és que la base de coneixement pot actualitzar-se ràpidament davant de cada nou exemple, cosa que permet l'adaptació i millora continuada dels sistemes.

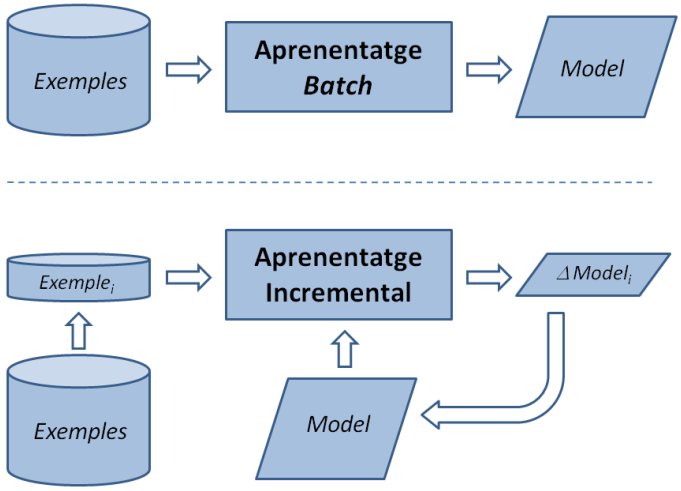

FIG. 10: Arquitectures *Batch* vs. Incremental

A més dels avantatges, els sistemes incrementals presenten alguns inconvenients fruit de les restriccions imposades per aconseguir una ràpida actualització. Els sistemes *batch*, no-incrementals, solen basar-se en cerques intensives en l'espai de solucions que, tot i ser

[1] *Incremental Machine Learning (IML)*

computacionalment costoses, garanteixen la trobada d'hipòtesis òptimes. Per contra els sistemes incrementals minimitzen el cost d'actualització, i per això no acostumen a guardar els exemples ni cap informació equivalent. Per aquest motiu els sistemes incrementals generalment necessiten analitzar més exemples per convergir en una hipòtesi estable, i solen sacrificar la certesa d'obtenir una hipòtesi òptima.

En aquest capítol es presenta i s'acota el significat d'incrementalitat, i es defineixen formalment les condicions, tan funcionals com d'escalabilitat, que ha de complir un algorisme d'aprenentatge per poder-se considerar incremental. També es descriuen els principals avantatges dels algorismes d'AAI, i es reflexiona sobre un dels punts febles que se'ls atribueixen: la dependència de l'ordre dels exemples d'aprenentatge.

4.2 CONCEPTES PREVIS

Per diferents motius, no existeix consens en la terminologia utilitzada per referir-se als algorismes d'aprenentatge incremental, i tampoc en les característiques que ha de tenir un algorisme incremental per poder-s'hi referir com a tal. Per aquest motiu en aquesta secció es presenten una sèrie de conceptes previs que volen ajudar a clarificar les idees.

La paraula *incremental* fa referència a "allò que succeeix en increments especialment petits", però en un algorisme d'AA hi ha diferents processos susceptibles de realitzar-se incrementalment. Recordem que els algorismes dels que parlem *analitzen* els exemples per extreure coneixement amb el que *construir* un model que s'utilitzarà per *classificar* altres elements **[Fig. 11]**.

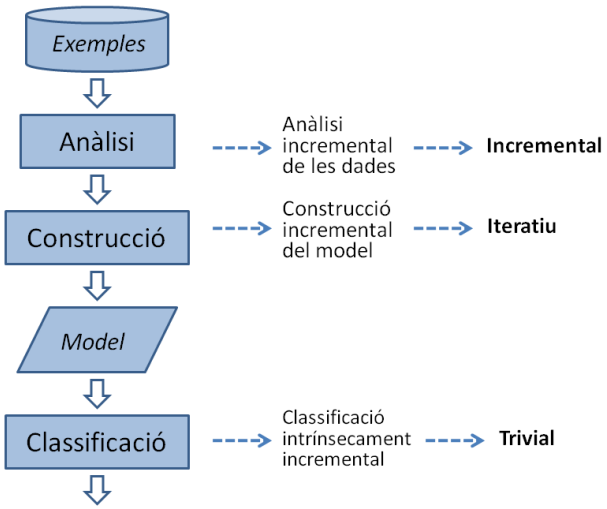

FIG. 11: Fases Incrementalitzables

Per tant, en principi, el terme "incremental" podria aplicar-se a qualsevol algorisme que realitzés algun dels tres processos de manera incremental: l'anàlisi dels exemples, la construcció del model o la pròpia classificació. És aquesta ambigüitat produïda per

l'adjectiu "incremental" la que probablement ha dificultat la utilització consistent d'una terminologia.

En els següents punts s'aprofundeix en els significats d'aquests tres nivells d'incrementalitat, començant pel cas més senzill (la classificació) i acabant amb el més valuós (l'anàlisi).

4.2.1 CLASSIFICACIÓ INCREMENTAL: TASCA TRIVIAL

Com explica **[Giraud-Carrier 2000]**, el terme *incremental* s'ha aplicat indistintament tant als algorismes d'aprenentatge com a les tasques a aprendre, el que ha provocat algunes confusions. Per això, ofereix algunes definicions formals que intenten aportar una mica de llum:

> ***Definition 1****: A learning task is incremental if the training examples used to solve it become available over time, usually one at a time.*

Aquesta primera definició subratlla la diferència entre un algorisme incremental i una tasca incremental. És a dir, són tasques incrementals aquelles en les quals, per la seva naturalesa, no es té accés simultani al conjunt de casos a resoldre o en les que el temps necessari per recopilar-los excediria els requeriments del problema[2]. És important entendre que la incrementalitat d'una tasca és un tret intrínsec del problema, no de l'algorisme utilitzat per resoldre'l, i està determinada pel ritme en el qual apareixen els exemples que s'han de processar.

Des d'aquest punt de vista, és clar que la classificació és una tasca intrínsecament incremental, classificar un exemple no necessita la presència de la resta d'elements. Per contra, l'ordenació és una tasca essencialment no-incremental, perquè no és possible ordenar un element sense la presència de la resta. Una tasca és incremental si es pot resoldre processant individualment els elements, i una tasca és no-incremental si suposa el processament de tots els elements com a conjunt. Per tant, si qualsevol classificador realitza la seva feina de manera incremental, independentment de que hagi estat induït incrementalment o en mode *batch*, la classificació incremental és un requeriment trivial.

Una altra característica que converteix una tasca en incremental és que el context de la tasca, i fins i tot la tasca mateixa, variïn al llarg del temps. En problemes situats en entorns canviants, amb tasques que evolucionen al llarg del temps, és imprescindible resoldre els problemes i aprendre dels resultats a mesura que van apareixent, ja que la simple recopilació de casos proporcionaria un conjunt d'exemples massa heterogeni com per poder beneficiar-se del processament conjunt.

---

[2] Aquesta matisació és important, ja que conceptualment qualsevol tasca incremental podria transformar-se en no-incremental si estiguéssim disposats a esperar-nos el temps suficient per recopilar tots els casos a processar.

Així doncs, les principals característiques que fan que una tasca sigui incremental són a) que no es disposi *a priori* de tots els exemples a processar, i b) que l'aprenentatge i adaptació hagi de realitzar-se indefinidament. Es pot veure que aquestes dues característiques poden aplicar-se perfectament a la majoria de tasques reals de PLN. L'existència de fenòmens pocs freqüents és una característica del llenguatge que impedeix l'existència de corpus *complets*; i al mateix temps justifica el desenvolupament d'aplicacions *vives* que funcionin amb processos de millora continuada. És per això que les tasques de PLN poden considerar-se tasques incrementals, que poden ser resoltes més eficientment per algorismes incrementals.

4.2.2 CONSTRUCCIÓ INCREMENTAL: MODEL ITERATIU

El següent nivell d'incrementalitat que pot presentar un algorisme d'AA és en la construcció del model. En aquests casos la inducció del model es realitza incrementalment mitjançant algorismes iteratius.

Un algorisme iteratiu és un algorisme que realitza una tasca, per exemple calcular un valor o obtenir un model, mitjançant una sèrie d'aproximacions successives a partir d'una primera aproximació heurística. És a dir, per acomplir la seva tasca realitza una sèrie d'iteracions que acaben convergint, més o menys ràpidament, en la solució del problema. Si parlem d'AA iteratius, parlem d'algorismes que processen els exemples mitjançant múltiples passades, de manera que en cada iteració s'extrau més coneixement i es millora el model.

Un exemple clàssic d'aquest aprenentatge iteratiu és l'etiquetador proposat per **[Brill 1992]** que obté un conjunt de regles de correcció, mitjançant un procés iteratiu que, en cada passada, analitza el corpus d'entrenament i afegeix al conjunt una regla addicional.

El que és important tenir en compte és que aquesta mena d'algorismes, malgrat ser iteratius, continuen sent algorismes *batch*; ja que durant cada una de les iteracions necessiten tenir accés a la totalitat dels exemples, i les actualitzacions que enriqueixen el seu model es fan a partir d'estadístiques globals del conjunt d'entrenament.

Per això, és preferible referir-se a aquests algorismes com *iteratius*, i reservar l'adjectiu *incremental* pels algorismes del següent nivell.

4.2.3 EXTRACCIÓ INCREMENTAL: ANÀLISI ON-LINE

Finalment arribem a aquells algorismes que no només construeixen el model de manera incremental, sinó que ho fan processant incrementalment les dades d'entrenament. A diferència dels anteriors, la actualització incremental del model es pot realitzar després d'analitzar un exemple individual, la qual cosa permet analitzar incrementalment el conjunt d'entrenament. Aquesta característica és la que permet a l'AAI processar grans corpus sense patir limitacions causades per problemes d'escalabilitat.

Aquests algorismes també són coneguts com algorismes d'aprenentatge *on-line*; però degut a els diferents significats[3] que es poden associar a aquesta expressió, aquest terme pot donar lloc a confusió. Essencialment, un algorisme *on-line* és un algorisme que és utilitzat en temps real sobre un determinat flux de dades. Per tant, el que el fa *on-line* és la forma com s'aplica, no l'algorisme en sí; sota aquest punt de vista el terme *on-line* defineix més un tipus de tasca que un tipus d'algorisme.

Per tant, al parlar d'aprenentatge *on-line* ens referim a aprenentatge incremental, un sistema d'inducció de coneixement que aprèn a partir d'un exemple cada vegada. El motiu és donat que l'aprenentatge *on-line* ha de respondre en temps real, és habitual que utilitzin algorismes incrementals, ja que tenen fases d'aprenentatge molt eficients. Així doncs, tot i que molts classificadors *on-line* utilitzen algorismes d'inducció incremental, no són característiques equivalents: ni els algorismes incrementals han d'utilitzar-se forçosament en tasques *on-line*, ni els algorismes *on-line* han de basar-se necessàriament en algorismes incrementals.

Finalment, comentar que, com tot aprenentatge supervisat, només poden aprendre a partir d'exemples anotats, amb el cost corresponent que suposa. Tot i que existeix un tipus de tasques *on-line* on el *feedback* és gratuït: la predicció d'esdeveniments futurs. En aquest tipus de taques l'algorisme simplement ha d'esperar al següent element per obtenir l'anotació de l'exemple anterior, el que permet estalviar l'anotació dels exemples.

4.3 INCREMENTALITAT

L'aprenentatge incremental es contraposa a l'aprenentatge *batch* en la manera com processa els exemples a l'hora de construir el seu model. En el *batch*, l'aprenentatge es realitza després d'analitzar tots els exemples, entesos com un conjunt no ordenat, mentre que en el l'aprenentatge incremental es duu a terme progressivament amb cada exemple presentat, entenent els exemples con una seqüència d'elements individuals.

Segons **[Giraud-Carrier, 2000]**, els principals avantatges d'un algorisme d'aprenentatge incremental són a) que no necessiten re-processar exemples anteriors, cada exemple és analitzat una única vegada, i b) com que cada model és, fins al moment, la millor aproximació possible de la tasca, el model predictiu pot utilitzar-se en qualsevol moment i la seva precisió anirà augmentant en el temps. Com hem vist en la secció anterior, aquestes característiques responen a la doble incrementalitat del seus processos: en l'anàlisi de les dades i en la construcció del model.

En aquesta secció es descriu amb més detall les característiques que fan que un algorisme d'aprenentatge sigui incremental, i es presenten definicions més formals sobre el funcionament i l'escalabilitat que idealment hauria de presentar.

---

[3] *On-line*: a) que funciona sota el control de l'ordinador principal; b) que funciona connectat a una xarxa informàtica; c) que ofereix serveis mitjançant Internet; d) que es troba actiu i preparat per funcionar; e) que es realitza mentre el sistema està operatiu; [Font *Random House Dictionary*, 2009].

4.3.1 INCREMENTALITAT FUNCIONAL

Des del punt de vista estrictament funcional, model *black-box*[4], és incremental qualsevol algorisme d'aprenentatge que pugui processar els exemples individualment i que, en qualsevol moment, puguem demanar-li el model classificador induït a partir dels exemples anteriors. Però en aquest cas, com s'apunta a **[Giraud-Carrier, 2000]**, qualsevol algorisme d'aprenentatge *batch* podria utilitzar-se de manera incremental, només caldria que anés emmagatzemant tots els exemples rebuts i re-entrenés el model des de zero per cada nou exemple.

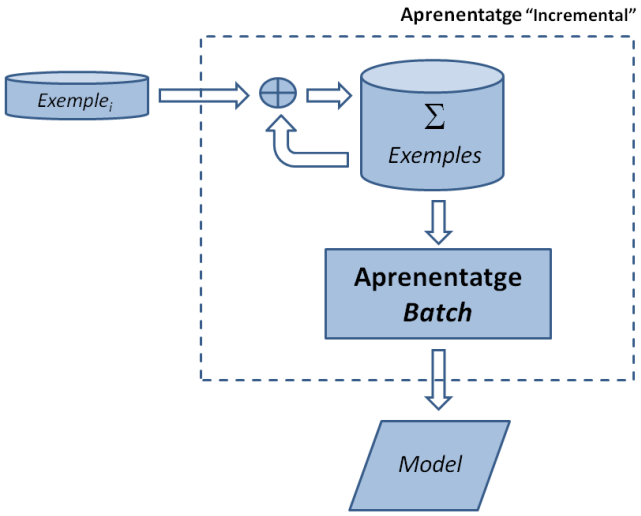

FIG. 12: Incrementalització funcional d'un algorisme *Batch*

Intuïtivament, es veu que aquest sistema traeix l'esperit de la incrementalitat tot i processar els exemples individualment, el seu model creix linealment amb els exemples analitzats, i el que és pitjor, el temps computacional necessari per *actualitzar* el model augmenta amb cada exemple. A la **Fig. 13** es mostra l'augment lineal del temps de computació d'un algorisme realment incremental comparat amb l'augment exponencial d'un algorisme *batch* utilitzat incrementalment **[Aseltine, 1999]**.

[4] En la teoria de sistemes es denomina *black box* (caixa negra) a aquells elements que s'analitzen únicament en termes de les seves entrades i sortides, és a dir, dels que es descriu la seva interacció amb altres mòduls però sense tenir en compte el seu funcionament intern.

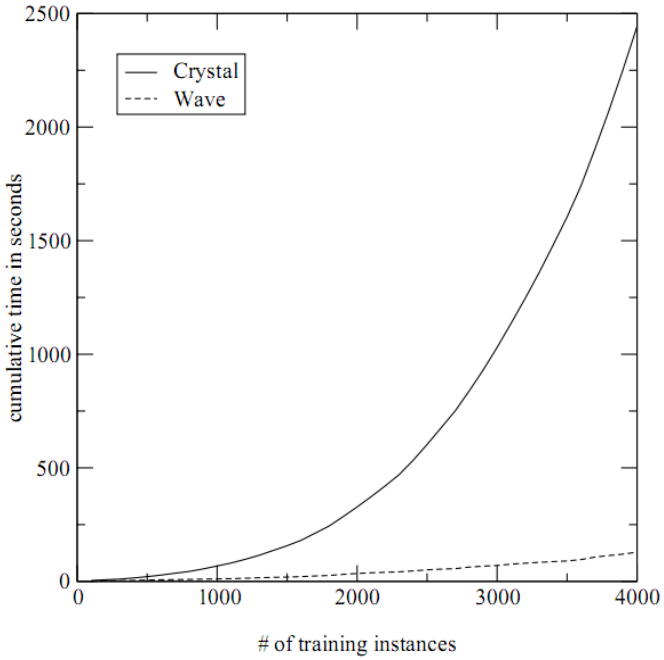

FIG. 13: Comparativa entre el temps de computació d'un algorisme incremental (Wave) i d'un *batch* utilitzat incrementalment (Crystal). Font [Aseltine, 1999].

4.3.2 INCREMENTALITAT IDEAL

Per tant, sembla clar que la *incrementalitat funcional* no és un criteri suficient per determinar aquells algorismes que poden oferir-nos els avantatges potencials que es deriven de la incrementalitat. És el que a **[Schlimmer & Fisher, 1986]** anomenen "*more than incremental behavior*":

> *"An important point though, is that any non-incremental algorithm can be made to behave in an incremental fashion. In general, however, incremental behavior does not insure the computational efficacy of such an algorithm." [Schlimmer & Fisher, 1986:500]*

On, a més, subratllen la importància d'explicitar les propietats computacionals de les tècniques d'inducció incremental, i no limitar-se a la caracterització del seu comportament incremental. Ja que és possible adaptar qualsevol cerca exhaustiva perquè accepti els exemples incrementalment, tot i que aquesta solució no seria de cap utilitat en un entorn que requerís aprenentatge incremental.

Algunes de les característiques que podrien ajudar a descriure la qualitat d'un algorisme incremental ideal serien: a) el nombre d'exemples necessaris perquè el sistema aprengui un conjunt estable de conceptes, b) el cost computacional d'actualitzar el model per incloure el nou exemple observat, i c) la qualitat de les descripcions dels conceptes induïts pel sistema.

A l'hora de determinar el grau d'idealització que presenta un algorisme incremental, pot ser més important centrar-se en el grau d'escalabilitat del sistema. Un algorisme d'aprenentatge *ideal* hauria de presentar:

1) **Escalabilitat del temps d'*Entrenament*:** els recursos computacionals necessaris a l'hora d'actualitzar el model haurien de ser independents del nombre d'exemples observats, a ser possible constant.
2) **Escalabilitat en el temps de *Classificació*:** els recursos computacionals necessaris a l'hora de classificar un nou exemple haurien de ser independents del nombre d'exemples observats, a ser possible constant.
3) **Escalabilitat del *Model* de classificació:** els recursos de memòria necessaris per emmagatzemar el model induït haurien de ser independents del nombre d'exemples observats, a ser possible constant.

4.3.3 FORMALITZACIÓ

Si a mesura que s'adquireix més coneixement en forma d'exemples, només cal fer petites modificacions al model actual, i no cal re-processar tots els exemples vistos fins aquell moment, els sistemes incrementals poden suposar una disminució important de la complexitat algorítmica final **[Giraud-Carrier & Martinez, 1994a]**.

En l'aprenentatge incremental, la màxima eficiència computacional i de memòria, s'aconsegueix mitjançant l'actualització iterativa del model a partir de la informació present a cada nou exemple. Suposem que tenim un conjunt de dades D format per N exemples, a més d'un nou exemple individual d_i. En aquest cas, un algorisme incremental hauria de poder obtenir un model actualitzat que inclogués el nou exemple, sense accedir a les dades anteriors D, únicament a partir del model anterior i el nou exemple:

$$Model\ (D + d_i) = \mathcal{F}\ (\ Model(D),\ d_i\ )$$

Aquesta noció essencial d'un algorisme d'aprenentatge incremental també es clarifica a **[Giraud-Carrier, 2000]**:

> ***Definition 2$_a$:*** *A learning algorithm is incremental if, for any given training sample e_1, ..., e_n, it produces a sequence of hypotheses h_0, h_1, ..., h_n, such that h_{i+1} depends only on h_i and the current example e_i.*

En alguns casos pot ser convenient relaxar lleugerament aquesta definició per permetre que la hipòtesi h_{i+1} depengui, no només d'h_i, sinó també d'un petit subconjunt d'exemples [e_{i-k} , ..., e_{i-1} , e_i]:

> ***Definition 2$_b$:*** *A learning algorithm is incremental if, for any given training sample e_1, ..., e_n, it produces a sequence of hypotheses h_0, h_1, ..., h_n, such that h_{i+1} depends only on h_i and a small subset of the last training example [e_{i-k} , ..., e_{i-1} , e_i].*

Aquesta modificació ens permet considerar incrementals alguns algorismes que creïn models predictius de seqüències, per exemple d'n-grames, a partir de recursos de memòria mínims.

Però a més, com s'ha explicat a **[4.3.2 Incrementalitat Ideal]** l'algorisme incremental ideal hauria de tenir una propietat addicional: la incrementalitat dels recursos utilitzats. De manera que un increment de dades suposés un increment del model obtingut en un increment de temps:

$$\Delta\text{Dades} \Rightarrow \Delta\text{Model}_{\,\Delta\text{Temps}}$$

O dit d'una altra manera, si volem descartar els perills de la incrementalitat estrictament funcional, mitjançant l'aplicació d'un algorisme *batch* a un registre de tots els exemples observats, és necessari afegir dues restriccions addicionals: la incrementalitat temporal i la incrementalitat espacial.

TEMPS: COST COMPUTACIONAL

Una primera restricció que hauria de tenir un algorisme d'AAI és que el cost computacional associat a l'actualització del model *M*, per a cada nou exemple d_i, hauria de ser molt inferior al cost necessari per induir un model des de zero a partir del tots els exemples $D+d_i$, en cas contrari la incrementalitat no tindria sentit:

$$Cost_{Update}\ (M, d_i) \ll Cost_{Train}(D + d_i)$$

En segon lloc és raonable demanar que el temps necessari per induir un model a partir d'un conjunt d'exemples D, sigui proporcional al nombre d'exemples N:

$$Cost_{Train}\ (D) = \sum_{i=1}^{N} Cost_{Train}(d_i) \approx N \cdot Cost_{Train}(d_i)$$

Per tant, el cost computacional necessari per entrenar un determinat conjunt d'exemples seria equivalent a entrenar el mateix sistema en mode *batch*, però en el mode incremental el cost total es distribueix entre tots els exemples.

ESPAI: MEMÒRIA NECESSÀRIA

Per tant, de manera anàloga es pot aplicar la mateixa restricció als recursos de memòria, de manera que la mida del model generat sigui, com a màxim, proporcional al número d'exemples processat. :

$$Memoria_{Model}\ (D) < \sum_{i=1}^{N} Memoria_{Model}(d_i) \approx N \cdot Memoria_{Model}(d_i)$$

En el pitjor cas, això suposaria que l'espai necessari per emmagatzemar el model tindria un creixement del tipus O(N), tot i que alguns algorismes presenten unes necessitats O(log(N)) i, fins i tot, alguns models són de mida constant O(k).

És important tenir en compte la mida de les representacions del model induït, ja que és habitual que augmentin segons la quantitat d'exemples **[Banko & Brill, 2001a]**. A la **Fig.**

14 es mostra el creixement de la memòria ocupada per dos models, un induït mitjançant un classificador Winnow i un altre per un algorisme basat en memòria.

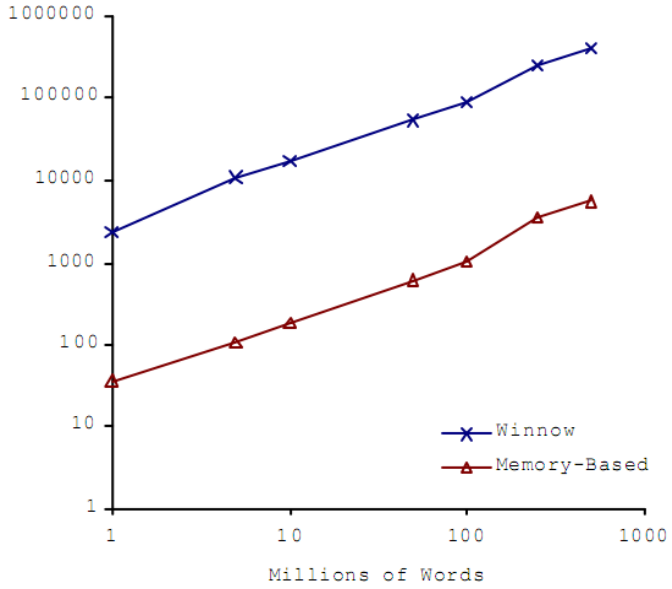

FIG. 14: Evolució de l'espai ocupat pel model segons la mida del corpus d'entrenament. Font [Banko & Brill, 2001b].

4.3.4 DECREMENTALITAT

No es pot tancar aquesta secció sobre els algorismes incrementals sense esmentar una altra característica que presenten alguns algorismes d'AAI: la *decrementalitat*. Informalment, la *decrementalitat* és la capacitat de *desaprendre* el coneixement obtingut a partir d'un exemple i restaurar l'estat del model previ a la presentació d'aquell exemple.

Els algorismes d'aprenentatge *decremental* poden obtenir el model predictiu induït d'un subconjunt de les dades, a partir del model complet i les dades complementàries:

$$Model\ (D - d_i) = \mathcal{F}\ (\ Model(D),\ d_i\ )$$

El fet de poder desaprendre exemples individuals, permet utilitzar determinades tècniques d'avaluació que eliminen redundàncies computacionals i milloren significativament la seva eficiència **[8.5.2 Validació Creuada]**.

4.4 AVANTATGES

A l'hora de resoldre tasques no-incrementals, la utilització d'algorismes incrementals pot ser ineficient ja que la informació a la que tenen accés és limitada, el model actual i el darrer exemple, sense poder accedir a la totalitat dels exemples. Per tant, en no tenir accés per avançat a tota la informació, és possible que l'algorisme prengui decisions locals, a curt termini, que acabin donant lloc a solucions sub-òptimes.

Aleshores, quin motiu hi ha per utilitzar algorismes incrementals?

Evidentment, l'existència de tasques incrementals. En aquests casos, l'elevat cost dels recursos dels algorismes *batch*, tant en temps com en memòria, fa ineficient i, fins i tot, impracticable, la creació des de zero d'un nou model cada vegada que l'algorisme observa un nou exemple.

Però existeixen altres motius que fan que els algorismes incrementals siguin atractius fins i tot en tasques aparentment no-incrementals, beneficis derivats de replantejar-les com a tasques incrementals. Per exemple, és possible que part del problema existent sigui considerar que les tasques de PLN són tasques no-incrementals. Si tenim en compte que les dades lingüístiques es distribueixen segons la llei de Zipf i, per tant, no és possible obtenir una mostra finita que sigui plenament representativa, el paradigma *batch* no sembla ser el model d'aprenentatge més raonable.

Per això, un nou punt de vista que consideri les tasques de PLN com a incrementals permet aprofitar els avantatges associats a l'AAI: 1) disposar de classificadors intel·ligents que contínuament milloren els seus resultats, 2) disposar d'algorismes d'entrenament ràpids que poden utilitzar-se en entorns interactius, i 3) disposar d'algorismes eficients que superen les restriccions a l'hora d' utilitzar grans corpus.

4.4.1 CLASSIFICADORS *VIUS*

A diferencia dels *batch*, els AAI generen classificadors *vius* que aprenen durant tota la seva vida, que enriqueixen els seus models i milloren la seva precisió de manera continuada.

Amb l'aprenentatge incremental desapareix la divisió entre període d'entrenament i període d'explotació. Per tant són classificadors utilitzables en qualsevol moment, cosa que suposa un avantatge econòmic en no haver-ne d'ajornar la utilització fins a l'anotació completa del corpus d'entrenament.

A més, el seu aprenentatge continuat els permet adaptar el seu model a noves dades que puguin aparèixer, cosa que indirectament resol el problema de *fora de domini*: en no existir un corpus d'entrenament diferenciat del corpus d'explotació, no existeix la necessitat de la *transportabilitat*.

4.4.2 UTILITZACIÓ INTERACTIVA

Una de les característiques dels algorismes incrementals és la seva eficiència computacional, en haver de processar un únic exemple cada vegada poden actualitzar el seu model en molt poc temps, de l'ordre de centèsimes o mil·lèsimes de segons en un ordinador típic.

Això permet utilitzar-los en entorns interactius, tant d'anotació com d'entrenament, en els quals després d'una primera classificació automàtica l'usuari corregeix els errors i l'algorisme utilitza aquesta informació per millorar el seu model classificador. La repetició d'aquesta seqüència al llarg de les dades permet anar augmentant la precisió de les classificacions i, al mateix temps, anar reduint la feina associada a la seva revisió.

Aquesta utilització dels AAI en entorns interactius permetria avançar en el desenvolupament d'eines d'anotació més eficients, donant lloc a eines d'*Anotació de Corpus Assistida per Ordinador (ACAO)*[5] que permetin reduir el coll d'ampolla de l'anotació.

4.4.3 CORPUS MASSIUS

Finalment, la utilització de sistemes incrementals permet enfrontar-se als corpus massius, eliminant les limitacions tècniques i metodològiques, i reduint significativament les econòmiques.

Les limitacions tècniques desapareixen, ja que els algorismes són escalables, amb temps d'entrenament lineal i recursos de memòria sub-lineals, i per tant poden processar corpus de qualsevol mida. Les limitacions metodològiques desapareixen, en no diferenciar entre entrenament i explotació no cal dedicar un temps previ (típicament mesos o anys) a l'anotació del corpus, i el sistema és utilitzable (amb diferents nivells de qualitat) des del primer moment. I les limitacions econòmiques es redueixen, la possibilitat d'anotar exemples interactivament redueix el cost de l'anotació, però a més, aquest cost menor es distribueix al llarg de tota la seva "vida", cosa que facilita la posada en marxa de projectes d'anotació a mitjà i llarg termini.

4.5 DEPENDÈNCIA DE L'ORDRE DELS EXEMPLES

Un dels aspectes més delicats de l'aprenentatge incremental és la seva dependència de l'ordre en el qual es mostren els exemples d'entrenament. Com s'explica a **[Giraud-Carrier, 2000]**, la *cronologia* és un aspecte inherent a la incrementalitat, és per això que la majoria dels algorismes incrementals generen models diferents segons l'ordre de la *seqüència d'entrenament*, un terme que en els algorismes incrementals hauria de substituir al de *conjunt d'entrenament*.

Però tot i que la cronologia pot utilitzar-se com a *biaix* positiu **[4.5.2 Entrenament *Pedagògic*]** això pressuposa que l'ordre dels exemples conté algun significat implícit, cosa que no sempre és certa. Per a la majoria de casos, en els quals l'ordre dels exemples pot considerar-se aleatori, el fet que un mateix algorisme pugui retornar models amb una important variació de qualitat predictiva pot considerar-se un inconvenient. Per això s'han proposat diferents tècniques per reduir la dependència de l'ordre dels sistemes incrementals **[Cornuéjols, 1993; MacGregor, 1988]**, tot i que encara no està clar si aquesta independència és possible.

Però més enllà d'això, la utilització de corpus massius i la redundància assumida en tasques incrementals, hauria de minimitzar qualsevol efecte provocat per patrons locals, interpretables en termes de soroll, i, per tant, els patrons estadístics globals haurien d'acabar determinant el model generat **[Giraud-Carrier, 2000]**.

[5] *Computer Aided Text Annotation (CATA)*

4.5.1 COMPLEXITAT INCREMENTAL

El 1993 es va publicar un article **[Elman, 1993]** en el qual es descriu una sèrie d'experiments amb xarxes neuronals artificials amb uns resultats que aporten una nova perspectiva en relació al *problema* de la dependència de l'ordre en els algorismes d'AAI.

L'objectiu dels experiments era entrenar xarxes neuronals perquè aprenguessin a reconèixer relacions de dependència en oracions subordinades. El corpus d'entrenament va ser generat mitjançant una gramàtica que permetia subordinacions recursives i garantia el compliment de determinades restriccions com la concordança i diferents règims verbals:

```
boys who chase dogs see girls
girl who boys who feed cats walk
cats chase dogs
mary feeds john
dogs see boys who cats who mary feeds chase
```

És important destacar que la representació interna de les paraules no incloïa cap mena d'informació gramatical. La tasca final de la xarxa consistia a predir la següent paraula, una tasca que forçava a la xarxa a construir una representació interna que codifiqués la informació gramatical.

En un primer experiment es va entrenar la xarxa amb una part del corpus generat i avaluat amb l'altre part. Els resultats van ser decebedors, la xarxa no va poder aprendre ni els exemples d'entrenament, i molt menys va poder generalitzar prou com per analitzar noves oracions. En un segon experiment es van dividir les oracions en simples i complexes, i es van preparar diferents corpus que contenien una proporció creixent d'oracions complexes[6]. Després d'entrenar la xarxa neuronal progressivament amb aquests corpus els resultats van ser molt diferents, obtenint un encert estimat del 85%.

És a dir, quan la xarxa va ser entrenada amb la totalitat de les dades, el model construït no va ser capaç de tractar ni tant sols les oracions simples, però si les mateixes dades eren mostrades en ordre de complexitat creixent, no només modelava les simples sinó que també podia arribar a tractar les complexes.

Un tercer experiment va donar uns resultats encara més sorprenents. Es va utilitzar el corpus complet, tant amb oracions simples com complexes, per entrenar la xarxa mitjançant cinc iteracions amb les mateixes dades. En aquesta ocasió el que es va modificar va ser la capacitat representacional de la xarxa, de manera que la mida efectiva de la seva finestra va augmentar-se progressivament des de tres fins a set paraules. Tot i que en les primeres passades la xarxa, amb una representació més limitada de les dades, va trigar més a convergir, els resultats van ser equivalents als del segon experiment. I finalitzades les cinc

[6] El primer conjunt estava format per un 100% d'oracions simples i 0% de complexes, el segon conjunt per un 75% /25%, el tercer 50%/50%, el quart 25%/75% i l'últim 0%/100%, és a dir, només per oracions complexes.

iteracions la xarxa havia après totes les oracions de l'entrenament i predeia correctament moltes de les oracions d'avaluació.

En altres paraules, tot i que les dades presentades tenien la mateixa complexitat durant les cinc iteracions, el fet que la capacitat representacional del sistema d'aprenentatge anés augmentat, va tenir un efecte equivalent a haver ordenat les dades de menys a més complexes.

Aquest experiment suggereix que els sistemes d'aprenentatge incremental dependents de l'ordre, entrenats amb exemples de complexitat creixent, poden resoldre tasques que els sistemes *batch* no són capaços de resoldre. Encara més, suggereix que la complexitat objectiva de les dades no és el factor determinant, sinó la complexitat representacional interna del sistema. En certa manera, limitar artificialment la capacitat representacional del sistema suposa simplificar les dades i projectar el problema a un espai de solucions més petit, i per tant, obre la porta a desenvolupar sistemes d'aprenentatge que construeixin models de complexitat progressiva.

S'ha de tenir en compte que en darrer terme l'aprenentatge humà és incremental i dependent de l'ordre, i habitualment aquest fet no es considera una limitació, sinó un avantatge pedagògic.

4.5.2 ENTRENAMENT *PEDAGÒGIC*

Un dels problemes d'aprenentatge més estudiat en psicologia és l'adquisició del llenguatge per part dels infants, i probablement l'argument més polèmic d'aquest debat, és la suposada pobresa de l'estímul (tant quantitativa com qualitativa) que faria impossible la inducció de la gramàtica a partir únicament d'exemples positius **[Chomsky, 1957]**. Com que és evident que l'adquisició del llenguatge és un fet, s'han proposat diferents explicacions per aquesta aparent paradoxa, des d'afirmar que l'estímul sí inclou exemples negatius (més o menys implícits), fins a considerar que existeix una gramàtica universal innata que restringeix l'espai de cerca.

On sí hi ha consens és en reconèixer que els infants no disposen d'un mecanisme d'aprenentatge universal capaç d'induir models sense restriccions i, per tant, capaç d'aprendre qualsevol llenguatge formal de complexitat superior al natural. És a dir, les restriccions s'assumeixen però la discussió apareix a l'hora de determinar quins tipus de restriccions existeixin i quin és l'origen d'aquests *biaixos*. De les diferents explicacions proposades n'hi ha dues que estan relacionades amb la dependència de l'ordre en els algorismes incrementals.

A **[Gold, 1967]** s'esmenta la possibilitat que l'ordenació intencionada dels estímuls, de més simples a més complexos, sigui suficient per permetre aprendre fins i tot els llenguatges més complexos, únicament amb exemples positius. Per un altre costat, a **[Elman, 1993]** es suggereix que les restriccions de l'espai de cerca de les gramàtiques són dinàmiques i estan correlacionades per a la pròpia maduració del sistema cognitiu, és a dir, per a l'evolució de

la seva capacitat representacional i de processament. En altres paraules, que tot i que l'estímul real rebut pel nen no tingui complexitat creixent, sí que la té l'estímul percebut, ja que a mesura que el sistema cognitiu madura augmenta la quantitat i complexitat de la seva representació.

El que aquestes dues explicacions suggereixen és que, en un mecanisme d'aprenentatge, la dependència de l'ordre pot ser un efecte desitjable, ja que permet la construcció progressiva d'un model que no seria assolible analitzant les dades en un procés *batch*. En certa manera la dependència de l'ordre permet introduir un *biaix* en el model adquirit mitjançant la seqüència d'entrenament, i obre la porta a un nou camp dins l'AA que estudiï la millor manera d'ordenar els exemples per portar a terme *entrenaments pedagògics*. És a dir, aquells on es tria l'ordre dels exemples de manera que faciliti l'entrenament i, donats uns recursos limitats, permeti obtenir millors models.

Com s'explica a **[Giraud-Carrier & Martinez, 1995]**, l'experiència d'un sistema d'aprenentatge incremental podria controlar-se presentant els exemples en un ordre determinat (per exemple, els casos regulars o les regles generals abans que les excepcions) amb l'objectiu d'accelerar l'aprenentatge (facilitar la convergència del model) o de millorar-ne l'eficiència (reduint les reconstruccions dels models). Una perspectiva molt intuïtiva des del punt de vista psicològic: un petit conjunt de regles simples pot resoldre una gran quantitat de casos, i només quan apareixen casos no coberts per aquestes regles és necessari enriquir el model incorporant sub-regles o excepcions.

Aquestes propietats de l'AAI, permetria realitzar entrenaments més flexibles, utilitzant aquest avantatge per entrenar models amb anotacions heterogènies. Una possibilitat seria entrenar inicialment eines de PLN amb grans corpus anotats superficialment amb poca profunditat, i posteriorment continuar l'entrenament amb corpus anotats amb més detall i més nivells lingüístics. Una altra possibilitat seria reutilitzar grans corpus generalistes per portar a terme un primer entrenament que generés un primer model aproximat, i utilitzar corpus més petits i especialitzats per fer l'ajustament fi del model. Totes dues estratègies permetrien racionalitzar els recursos dedicats a l'anotació de dades d'entrenament, fent servir corpus rics i especialitzats, només quan fossin realment útils.

5 CLASSIFICACIÓ INTER-ACTIVA

5.1 INTRODUCCIÓ

L'elevat cost que suposa l'anotació de corpus és la principal limitació que frena el desenvolupament d'aquests recursos, tant si es tracta de grans corpus generals com de petits corpus *ad-hoc*. I, per tant, la principal limitació que restringeix potencials avenços de la Lingüística Empírica (LE).

Per això, com s'ha explicat anteriorment **[3.4.1 Desenvolupament de Corpus Massius]**, en els darrers anys el procés d'anotació s'ha anat optimitzant mitjançant la introducció de diferents nivells d'automatització: anotació directa manual, anotació mitjançant Interfícies Gràfiques d'Usuaris[1], desambiguació d'anotacions automàtiques o simple revisió d'aquestes anotacions. Arribats a aquest punt la tasca de l'expert consisteix únicament a revisar el resultat de l'etiquetació i desambiguació automàtica i, per tant, des del punt de vista qualitatiu és difícil simplificar encara més aquesta tasca. A l'hora de continuar reduint el cost d'anotació, proporcional al temps invertit per l'expert, el següent pas ha de ser reduir quantitativament la seva feina, és a dir, minimitzar el nombre d'anotacions que calgui revisar.

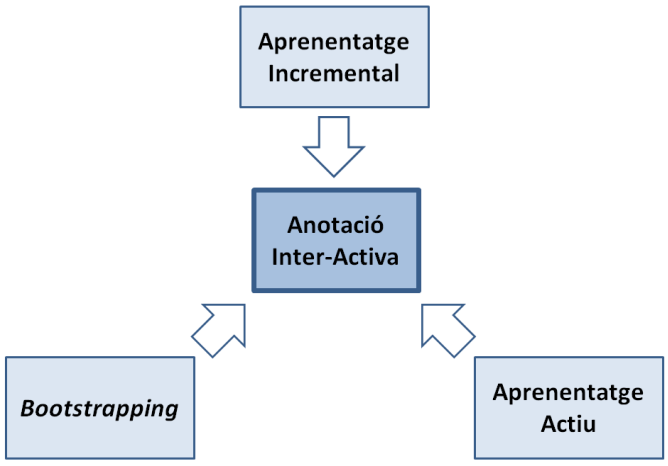

FIG. 15: Anotació Inter-Activa = A. Incremental + *Bootstrapping* + A. Actiu

En aquest capítol es presenta la classificació (o anotació) Inter-Activa, una proposta que es recolza en la combinació d'interfícies gràfiques, en el *bootstrapping* portat a la seva màxima expressió i en l'Aprenentatge Actiu, per obtenir eines interactives d'anotació assistida que ajudin a reduir el coll d'ampolla de l'anotació lingüística.

[1] *Graphic User Interface (GUI)*

5.2 BOOTSTRAPPING

S'anomena *bootstrapping* a un procediment utilitzat en Aprenentatge Automàtic (AA) per millorar els resultats d'un classificador mitjançant l'aplicació iterativa de fases d'entrenament i avaluació. El *bootstrapping* va ser presentat a **[Sung & Poggio, 1998]**, en aquest treball un classificador és entrenat inicialment amb un subconjunt de les dades, el classificador resultant és utilitzat per anotar la resta de les dades i els exemples classificats erròniament s'incorporen al subconjunt inicial d'entrenament; es continua repetint el procés fins assolir la qualitat desitjada.

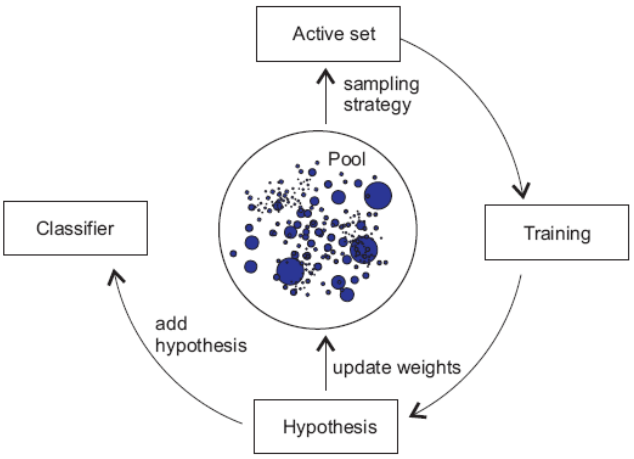

FIG. 16: Diagrama del procés de *bootstrapping* en l'anotació d'exemples d'un classificador. Font [Kalal 2008]

La idea que hi ha darrera d'aquesta tècnica és esbiaixar el mostreig de les dades cap als casos classificats erròniament, els més propers al llindar de decisió, de manera que el classificador rebi més exemples situats en les àrees dubtoses de l'espai de dades. A la **Fig. 16** es mostra l'esquema d'aquest procediment en què s'itera al llarg del cicle de mostreig, entrenament i esbiaixement **[Kalal, 2008]**.

Aquesta tècnica és utilitzada habitualment per entrenar eficientment sistemes automàtics d'anotació que s'utilitzaran en projectes d'anotació manual de grans corpus. En aquests casos es parteix d'un petit corpus anotat manualment que és utilitzat per induir un primer anotador mitjançant AA. Aquest anotador automàtic s'aplica a un fragment més gran del corpus, i el resultat es revisa manualment per garantir-ne la qualitat. Aquest corpus anotat i revisat, més gran que l'inicial, és utilitzat per entrenar un nou anotador, que previsiblement obtindrà millor qualitat. El procés es va repetint, anotant cada vegada fragments més grans amb anotadors que produeixen una menor taxa d'error i que, per tant, requereixen menys correccions **[Martí et al., 2008]**.

Aquesta tècnica accelera l'entrenament del classificador i l'anotació del corpus mitjançant la repetició del cicle: revisió manual, entrenament i anotació automàtica. Lamentablement la seva implementació habitual no és tan eficient com podria ser, ja que el procés no

acostuma a estar integrat en una única eina i els classificadors utilitzats solen basar-se en algorismes *batch*. Aquests dos condicionants fan que sigui habitual que cada iteració necessiti dies o setmanes per concloure's, endarrerint innecessàriament l'actualització de l'anotador automàtic i l'aprofitament del nou coneixement.

La utilització d'algorismes d'AAI dins d'eines interactives acceleraria el procés considerablement, reduint al mínim el temps necessari per a cada iteració. Cosa que permetria aplicar el *bootstrapping* en la seva màxima expressió, realitzant un cicle de *bootstrapping* per a cada document anotat, per a cada paràgraf anotat o fins i tot per a cada anotació individual, cosa que suposaria una reducció del nombre total de revisions o anotacions manuals.

5.3 APRENENTATGE ACTIU

L'Aprenentatge Actiu **[Thompson et al., 1999]** és una tècnica d'entrenament d'algorismes d'AA basada en la selecció d'exemples **[Cohn et al., 1994]**. Neix de la premissa que no tots els exemples són equivalents, sinó que existeixen alguns més interessants que altres a l'hora d'entrenar un classificador. L'Aprenentatge Actiu assumeix que no tots els exemples són igual d'informatius: alguns són redundants i no enriqueixen el model predictiu, per contra altres són molt prototípics, cosa que facilita la inducció del model, o són molt ambigus, cosa que permet afinar els llindars de classificació.

Les dues formes com habitualment s'aplica l'Aprenentatge Actiu són la selecció automàtica d'exemples i la reducció del conjunt de dades inicial. La selecció automàtica d'exemples analitza un corpus d'entrenament amb l'objectiu de seleccionar aquells exemples que, una vegada anotats per l'expert, maximitzin el coneixement extret per l'algorisme. La reducció del conjunt de dades s'utilitza per reduir un conjunt de dades que és massa gran per ser processat pels algorismes *batch* corresponents. La primera aproximació intenta atacar el problema del cost econòmic i, per tant, és rellevant tant amb algorismes *batch* com amb algorismes incrementals. La segona aplicació intenta resoldre el problema tècnic del processament de grans corpus, un problema inexistent si es treballa amb algorismes incrementals.

Així doncs, la selecció intel·ligent dels exemples d'entrenament permetria concentrar els esforços d'anotació en aquella fracció d'exemples més pedagògics i assolir precisions equivalents a l'anotació completa, però reduint els costos d'anotació de manera important.

L'Aprenentatge Actiu es defineix per oposició a l'aprenentatge passiu **[Finn & Kushmerick, 2003; Scheffer et al., 2002]** en el qual s'anoten tots els exemples del corpus d'entrenament de manera independent a l'algorisme. Com en aquest cas és l'algorisme qui selecciona quines dades cal anotar, és fonamental definir un criteri de selecció d'exemples. En el següent punt es presenten diferents criteris que poden fer-se servir a l'hora de portar a terme aquesta selecció dels exemples més informatius.

5.3.1 MESURES DE SELECCIÓ

En la majoria de casos l'Aprenentatge Actiu utilitza una forma de *bootstrapping* a partir d'un petit corpus germinal que s'utilitza per entrenar un primer classificador. La idea és aplicar aquest classificador sobre el corpus complet i, aleshores, permetre que sigui l'algorisme qui determini quins exemples cal anotar. L'assumpció habitual és que la significança d'aquests exemples és proporcional al grau d'incertesa de l'algorisme a l'hora d'etiquetar-los, de manera que siguin els casos més dubtosos els que l'expert ha de confirmar o corregir mitjançant la seva revisió.

Aquest sistema requereix que els classificadors utilitzats en un sistema d'Aprenentatge Actiu presentin una característica important: que la classificació realitzada inclogui una mesura quantitativa sobre el grau de certesa de l'etiqueta assignada. No és necessari que sigui estrictament una probabilitat d'error o una mesura de confiança precisa, simplement n'hi ha prou amb qualsevol mesura que es correlacioni amb l'error associat a cada classificació.

Existeixen diferents mesures que poden utilitzar-se per filtrar els exemples més informatius, però habitualment la mesura utilitzada ve determinada per l'algorisme d'aprenentatge del classificador. Per exemple, la mesura més directa és simplement l'*nivell de confiança* que generen alguns classificadors, corresponent a la probabilitat que l'anotació sigui correcta, però no tots els algorismes ofereixen aquesta possibilitat.

En aquests casos, si el classificador proporciona un valor quantitatiu del grau de pertinença a cada classe, tot i que no estigui normalitzat, pot aproximar-se el grau de certesa mitjançant el nivell relatiu de la sortida més alta en relació a la suma total de les sortides. Aquesta solució és la que s'ha aplicat tradicionalment en molts casos: a les sortides dels sistemes connexionistes, a la similitud en sistemes basats en memòria, al grau d'homogeneïtat associat a un element terminal d'un arbre de decisió, o a la precisió o suport d'una regla induïda, o a la distància de separació d'un classificador lineal. La **Fig. 17** il·lustra aquest darrer cas, mostra la distribució de probabilitat de selecció d'un exemple segons la seva distància a l'hiperplà separador; els exemples més propers tenen més probabilitat de ser seleccionats**[Sculley, 2007]**.

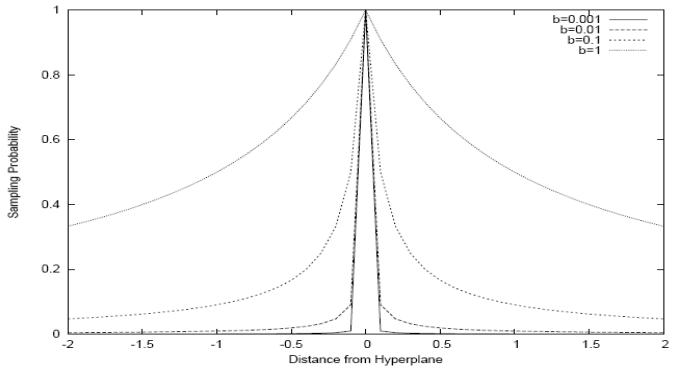

FIG. 17: Probabilitat de ser seleccionat segons la distància al hiperplà separador. Font [Sculley, 2007]

En altres algorismes, en els quals el resultat de la classificació no sigui *quantitativa* i només generi una etiqueta simbòlica, és possible obtenir un *índex de confiança* a partir del *grau d'acord* existent entre diferents classificadors combinats **[5.3.2 Selecció per Grau d'Acord]**. Finalment, existeix un darrer criteri, utilitzat únicament com a *línia basal*[2] a l'hora d'avaluar resultats, consistent en aplicar una selecció aleatòria. Això permet obtenir un mostreig neutre del corpus i comparar els resultats d'aquest entrenament *passiu* amb els de diferents variants de selecció intel·ligent de l'Aprenentatge Actiu.

5.3.2 SELECCIÓ PER GRAU D'ACORD

Com s'ha dit, l'Aprenentatge Actiu pot aplicar-se fins i tot amb classificadors que no ofereixin resultats numèrics que reflecteixin indirectament el grau de confiança de la classificació. La solució consisteix a agrupar un conjunt d'aquests classificadors **[7.3 Sistemes de Votació]** i utilitzar el *grau d'acord* entre els classificadors individuals com una mesura del grau de certesa.

A **[Banko & Brill, 2001a]** es va aplicar aquesta tècnica i es van obtenir els resultats mostrats a la **Fig. 18**; aquestes dades il·lustren la correlació entre el grau d'acord dels experts, en aquest cas format per un comitè de 10 classificadors, i la precisió real de la classificació consensuada:

Classifiers In Agreement	Test Accuracy
10	0.8734
9	0.6892
8	0.6286
7	0.6027
6	0.5497
5	0.5000

FIG. 18: Precisió de la classificació segons el grau d'acord en un comitè de 10 *experts*. Font [Banko & Brill, 2001a]

La utilització d'una combinació de classificadors amb l'objectiu d'obtenir un *índex de confiança* de la classificació és una tècnica que s'ha utilitzat en diferents treballs: a **[Breiman, 1996]**, a **[Dagan & Engelson, 1995]** per l'anotació de categoria gramatical, i a **[Banko & Brill, 2001a]** per la desambiguació de *conjunts de confusió*.

5.3.3 MIDA DEL REPOSITORI DE SELECCIÓ

A l'hora d'utilitzar l'Aprenentatge Actiu en una eina d'anotació l'objectiu és reduir el nombre de revisions necessàries, seleccionant els casos més dubtosos i donant com a bons la resta; per tant, l'objectiu final és anotar la totalitat del corpus amb el mínim esforç. Però existeix un altre escenari on l'eina d'anotació és utilitzada únicament com a eina

[2] *Baseline*

d'entrenament, en aquest cas l'objectiu final no és obtenir un corpus anotat sinó induir un model classificador que obtingui la major precisió possible donada una determinada assignació de recursos per a l'anotació.

En aquest cas és possible utilitzar una estratègia encara més eficient, la utilització d'un corpus massiu com a repositori[3] del qual seleccionar els millors exemples per entrenar el classificador. En aquest cas, l'algorisme explora el corpus i selecciona aquells exemples que considera més interessants per ser anotats i utilitzats per entrenar el classificador. Per això, tot i no estar anotades, l'existència d'una gran quantitat de dades entre les quals poder triar suposa un avantatge que permet obtenir models més precisos tot i ser entrenats amb la mateixa quantitat d'exemples anotats.

A l'experiment de **[Banko & Brill, 2001a]** s'avalua el comportament de tres combinacions de classificadors (formats per 10 classificadors *Naïve Bayes*) que són entrenats mitjançant Aprenentatge Actiu. En els tres casos els classificadors van ser entrenats amb 1 milió d'exemples seleccionats activament d'un conjunt més gran, la diferència es trobava en la mida del repositori del qual se seleccionaven els exemples[4]. La meitat dels exemples es va seleccionar aleatòriament, per mantenir un cert grau de representativitat del corpus, i l'altre meitat es va seleccionar segons el nivell de desacord que provocaven en el comitè de classificadors. Per ser utilitzat com a referència també es va entrenar tradicionalment un classificador sobre la totalitat dels 100 milions d'exemples disponibles.

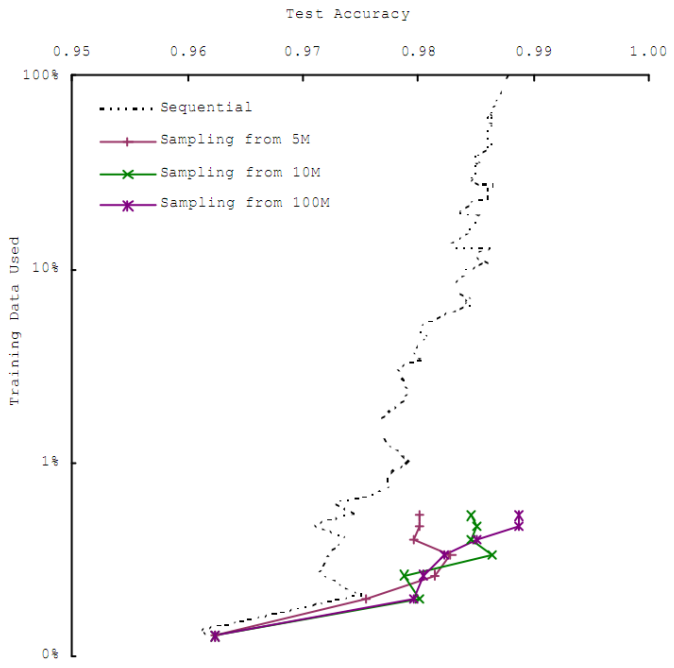

FIG. 19: Evolució de la precisió segons la quantitat d'exemples per diferents mides del repositori. Font [Banko & Brill, 2001a].

[3] *Training pool*
[4] Pels tres entrenaments les mides dels repositoris eren de 5, 10 i 100 milions d'exemples respectivament.

A la **Fig. 19** es mostra l'evolució de la precisió segons la mida del repositori; s'observa que, comparats amb l'entrenament tradicional, tots tres classificadors presenten una important acceleració en l'entrenament. Els resultats obtinguts després d'entrenar els classificadors amb 1 milió de paraules mostren que les precisions aconseguides mitjançant la *selecció activa* són més elevades (98%-99%) que les aconseguides mitjançant selecció aleatòria (97%).

A més, els resultats dels tres classificadors actius mostren diferències significatives, millors resultats com més gran era al conjunt de dades d'on triaven els exemples. En el millor dels tres casos, on es van seleccionar els exemples d'un repositori de 100 milions de candidats, el classificador obté una precisió equivalent a l'entrenament tradicional, però necessitant menys d'1% del total del corpus. Si aquests resultats fossin extrapolables a altres tasques de PLN suposaria una importantíssima reducció de costos en l'anotació de corpus.

5.4 ANOTACIÓ INTER-ACTIVA

Finalment ja tenim tots els elements necessaris per descriure l'Anotació Inter-Activa: una arquitectura d'anotació assistida que combina el *bootstrapping*, que permet incrementar progressivament la precisió de l'anotació automàtica, amb l'Aprenentatge Actiu, que permet minimitzar el número d'anotacions que cal revisar manualment.

Per un costat, recordem que el *bootstrapping* és una tècnica iterativa que utilitza una sèrie creixent de fragments del corpus per entrenar una seqüència d'anotadors automàtics progressivament més fiables. Però la utilització d'algorismes d'AA *batch* introdueix una certa ineficiència en la reutilització del coneixement adquirit **[Ciravegna et al., 2002a]**, ja que el fet que la iteració es realitzi després de l'anotació d'un grup de documents, impedeix que el coneixement extret en els primers documents es pugui aplicar en la resta.

Idealment, per aprofitar al màxim el coneixement present als exemples previs i reduir el màxim el nombre d'anotacions necessàries, el cicle s'hauria d'escurçar tant com fos possible, fins i tot repetint-lo per a cada exemple individual. Evidentment el cost computacional necessari per re-entrenar un classificador *batch* després de cada anotació fa que sigui inviable. Però no seria així si només calgués actualitzar un classificador incremental, un procés molt més ràpid; és per això que els algorismes d'AAI permeten portar al límit la idea del *bootstrapping*.

Per l'altre costat, en l'Aprenentatge Automàtic estàndard el sistema analitza el corpus amb l'objectiu de seleccionar activament els exemples més valuosos per, posteriorment, entrenar el classificador *batch*. És per això que la tasca és plantejada com una mena de preprocessament del corpus, conegut com a *mostreig selectiu* o *mostreig intel·ligent* **[Fujii et al., 1998]**, amb l'objectiu d'extreure els exemples més representatius del corpus.

Però la iterativitat intrínseca del procés obre la porta a la intervenció humana en finalitzar cada un dels cicles d'aprenentatge i, per tant, fa possible l'aplicació interactiva de l'Aprenentatge Actiu. En aquest cas l'expert pot anotar seqüencialment una sèrie de texts,

de manera que aquestes anotacions siguin incorporades immediatament en el model d'aprenentatge, cosa que permet al sistema utilitzar aquest nou coneixement en les classificacions immediatament següents. D'aquesta manera, excepte en els primers cicles, el sistema és capaç d'ajudar l'expert proposant-li anotacions que aquest només haurà de validar o corregir. A mesura que el procés es repeteix i l'anotació automàtica millora, l'eina d'anotació evoluciona i millora al llarg de la seva vida: des de l'anotació manual de tots el exemples, a la simple supervisió i correcció d'una petita part de les anotacions fetes pel sistema.

Com s'explica a **[Busser et al., 2005]**, tot i que l'Aprenentatge Actiu és una tècnica aplicable a qualsevol algorisme d'AA, val la pena tenir en compte algunes consideracions pràctiques a l'hora de triar l'algorisme, especialment en aplicacions interactives. En aquests casos calen algorismes que no requereixin temps d'entrenament desproporcionadament llargs, i per això són preferibles algorismes entrenables incrementalment.

Si es tenen en compte els dos criteris que permeten aconseguir un *bootstrapping* eficient i una interactivitat fluida, els algorismes d'aprenentatge incremental són els algorismes idonis per utilitzar en una eina d'Anotació Inter-Activa. Tot i que generalment acostumen a necessitar més exemples per aconseguir el mateix nivell de precisió, la **[Fig. 20]** mostra la seva comparació per una mateixa quantitat d'exemples, la combinació d'aquestes tècniques permet que la feina necessària per revisar les anotacions sigui menor.

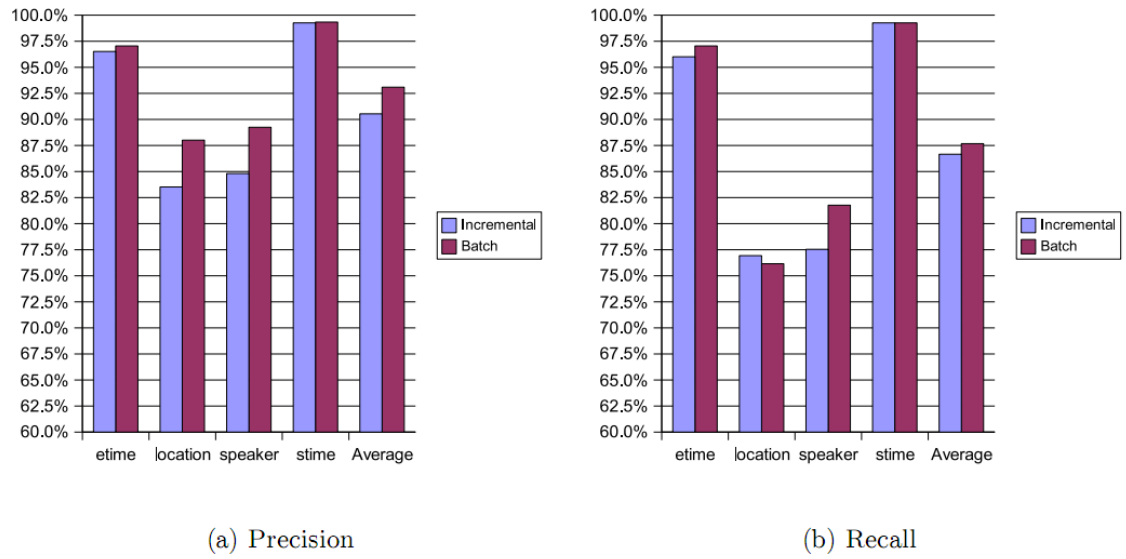

FIG. 20: Comparació dels resultats d'un sistema d'EI amb entrenament incremental i *batch*. Font [Siefkes, 2008].

El fet que la qualitat del classificador vagi augmentant amb el temps, significa que també augmenta el grau de certesa de les seves anotacions, i per tant va disminuint la quantitat d'anotacions que cal revisar. A la **Fig. 21** es mostra l'evolució de la proporció d'exemples que requereixen verificació per part de l'expert en un classificador Inter-Actiu de correu brossa **[Sculley, 2007]**, la proporció inicial d'un 10% arriba a reduir-se a menys de l'1% dels exemples.

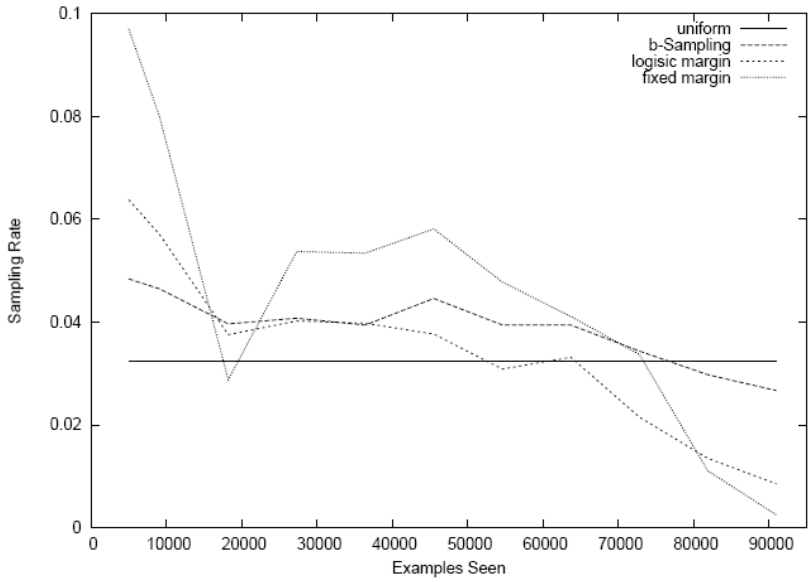

FIG. 21: Evolució de la proporció d'exemples que requereixen revisió manual, per 4 mesures de selecció, en un classificador Inter-Actiu. Font [Sculley, 2007]

Finalment, cal tenir en compte que moltes de les aplicacions industrials de PLN s'insereixen en processos continus on l'objectiu és processar un flux permanent de documents, com correus electrònics o noticies de diaris, i també en aquests casos la solució que millor s'adapta és l'aprenentatge incremental.

5.5 ARQUITECTURA INTER-ACTIVA

Per tant, el següent pas en l'evolució dels entorns gràfics d'assistència a l'anotació és la incorporació de sistemes que no només aprenguin a partir de les anotacions realitzades, sinó que a més permetin filtrar o assenyalar aquelles anotacions automàtiques que tinguin més necessitat de ser revisades per l'expert. Això permetria focalitzar la seva atenció en aquells casos dubtosos, amb una major probabilitat de ser erronis, i treure el màxim rendiment del seu coneixement a l'hora d'entrenar el sistema d'anotació automàtica.

Una característica interessant d'aquests sistemes Inter-Actius[5] és que la seva utilització produeix tres *productes* diferenciats **[Fig. 22]**: un model classificador, un corpus anotat i un corpus *gold standard* . El model classificador induït és un producte per si mateix que pot utilitzar-se en qualsevol moment de manera automàtica i no-interactiva. El corpus anotat és el resultat de l'anotació automàtica i de la revisió selectiva, amb una baixa taxa d'error suficient per a la majoria d'aplicacions. I el corpus *gold standard* està constituït pel subconjunt d'exemples revisats manualment, que tot i estar esbiaixat, ofereix la taxa d'error

[5] En aquest treball s'utilitza el terme *Inter-Actiu* per referir-se a aquells sistemes interactius, tant d'anotació com d'entrenament, que combinen interfícies gràfiques, amb tècniques d'Aprenentatge Actiu i algorismes d'Aprenentatge Automàtic Incremental.

més baixa possible en una anotació manual. A més, tots tres productes són vius: tot i millorar i créixer contínuament, poden ser utilitzats en qualsevol moment.

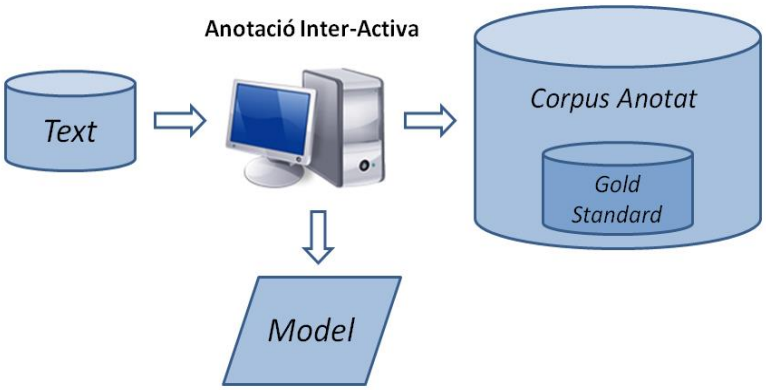

FIG. 22: Els tres productes generats per un sistema Inter-Actiu: un model, un corpus anotat i un *gold standard*.

Una altra característica important, des del punt de vista de la seva aplicació *industrial*, és la flexibilitat de l'arquitectura que aquests sistemes Inter-Actius ofereixen. La configuració més senzilla només necessitaria una sistema local per realitzar l'anotació/entrenament d'una determinada tasca de PLN **[Fig. 23]**.

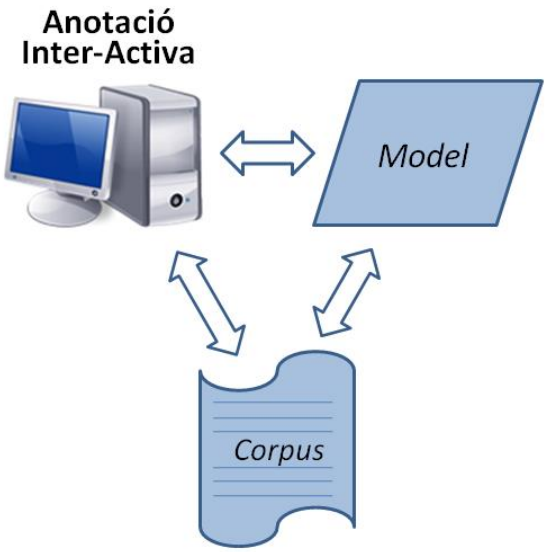

FIG. 23: Arquitectura mínima: 1 entrenador i 1 model.

A causa de l'eficiència de l'entrenament incremental, el coll d'ampolla d'aquest sistema interactiu es trobaria en el costat humà. El temps necessari perquè l'expert seleccioni o corregeixi els exemples, faria que el classificador estigués infrautilitzat, ja que la major part del temps estaria aturat esperant la correcció. Per això seria possible utilitzar una arquitectura tipus *web-service* on l'anotador automàtic, al mateix temps que és entrenat i corregit, pogués realitzar anotacions automàtiques a texts sol·licitats per ordinadors remots **[Fig. 24]**.

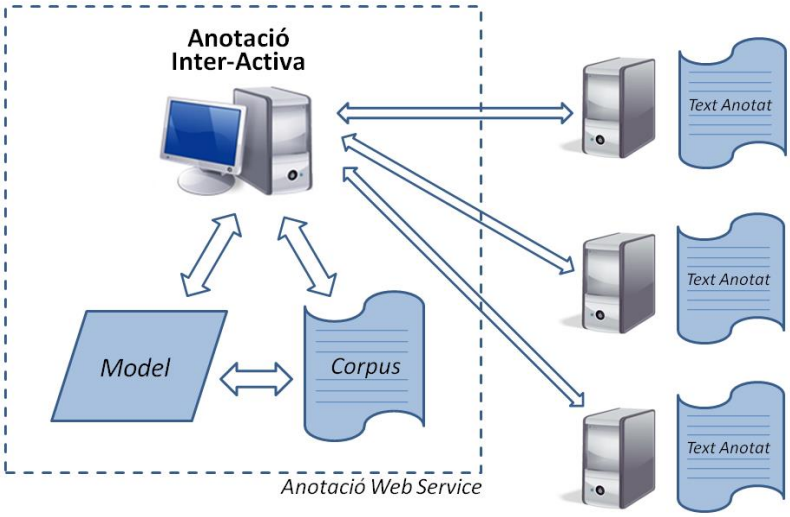

FIG. 24: Arquitectura *web-service*: 1 entrenador, 1 model i N clients.

Una altra possibilitat interessant seria la utilització d'una arquitectura d'entrenament client-servidor. La incrementalitat de l'algorisme d'aprenentatge implica un *no-state process*, de manera que els exemples d'entrenament són entitats aïllades que inclouen tot el context necessari. Això permetria l'entrenament col·lectiu d'un únic classificador per part de múltiples experts revisant texts diferents de manera simultània. Amb aquesta configuració la possibilitat d'una eina col·lectiva d'anotació interactiva obriria la porta a una veritable explotació industrial de les tecnologies lingüístiques **[Fig. 25]**.

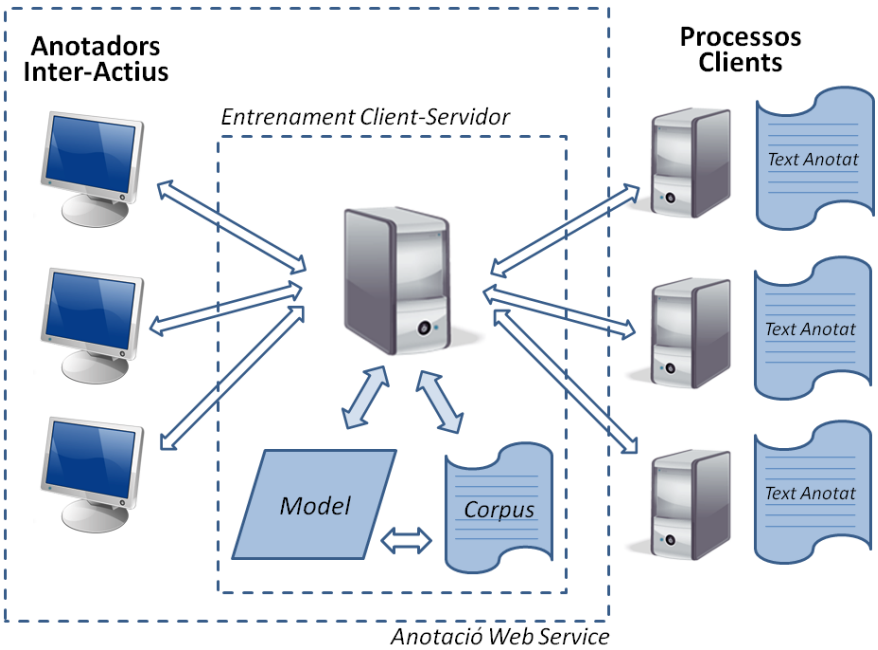

FIG. 25: Arquitectura client-servidor: N entrenadors, 1 model i N usuaris.

Finalment, en aquelles aplicacions on no n'hi hagués prou amb una anotació automàtica d'alta qualitat, l'arquitectura podria incloure un canal de realimentació, de manera que aquells exemples que l'anotació automàtica no pogués resoldre fossin reenviats a l'equip

d'experts per a ser anotats manualment de manera diferida **[Fig. 26]**; la qual cosa permetria reforçar l'aprenentatge amb els casos més difícils.

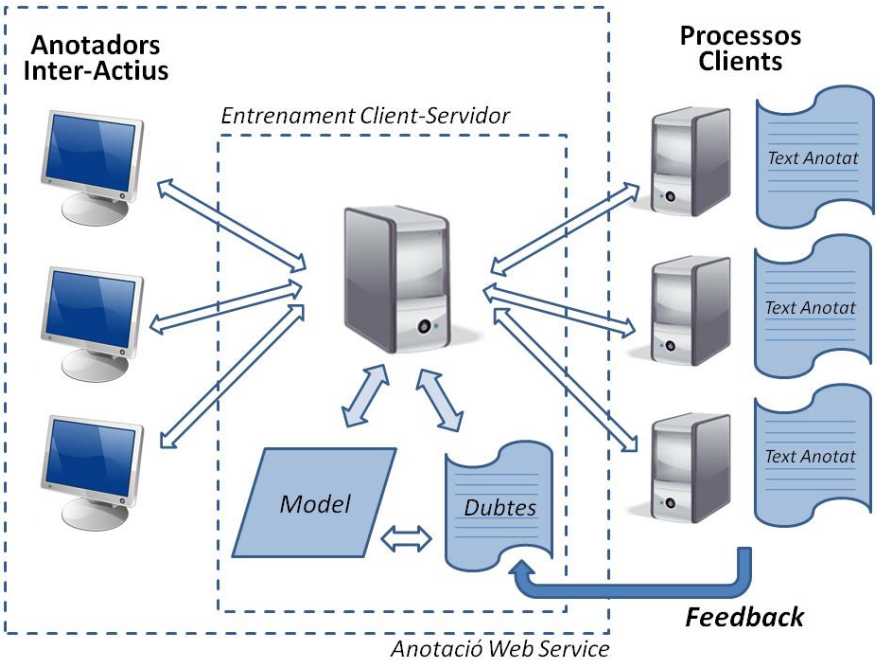

FIG. 26: Arquitectura amb *feedback*: N entrenadors, 1 model, N usuaris, i realimentació de dubtes.

5.6 EXEMPLES D'APLICACIÓ

L'anotació mitjançant Aprenentatge Actiu no és una tècnica massa estesa, i en el casos on s'aplica acostuma a tractar-se de tasques d'anotació en els nivells lingüístics més avançats, com l'anotació de noms d'entitat **[Shen et al., 2004]**, l'anotació de rols semàntics **[Busser et al., 2005]** o de resolució anafòrica **[Gasperin, 2009]**.

Si l'objectiu és trobar exemples que aprofitin aquesta tècnica en entorns gràfics interactius, la diversitat encara és menor i la tasca dominant és l'Extracció d'Informació[6] (EI), una de les àrees capdavanteres del PLN en l'aplicació de tècniques d'Aprenentatge Automàtic.

A **[Siefkes, 2005]** es descriu una aplicació, que acompleix tots els requeriments d'una eina d'anotació Inter-Activa, per realitzar EI a partir d'una variant del classificador Winnow **[6.6.1 Winnow]**. Amb aquest sistema l'usuari va anotant seqüencialment una sèrie de documents, de manera que les anotacions corresponents són incorporades immediatament en el model d'extracció. Això permet al sistema facilitar la feina a l'anotador proposant noves extraccions del proper document a processar. Una vegada el document ha estat corregit és utilitzat per actualitzar el model d'extracció abans de processar el següent document. D'aquesta manera, les propostes d'extracció d'informació continuen millorant

[6] *Information Extraction (IE)*

durant la utilització del sistema. A la **Fig. 27** es mostra l'evolució de la qualitat d'aquest sistema a mesura que augmenta al nombre de documents processats.

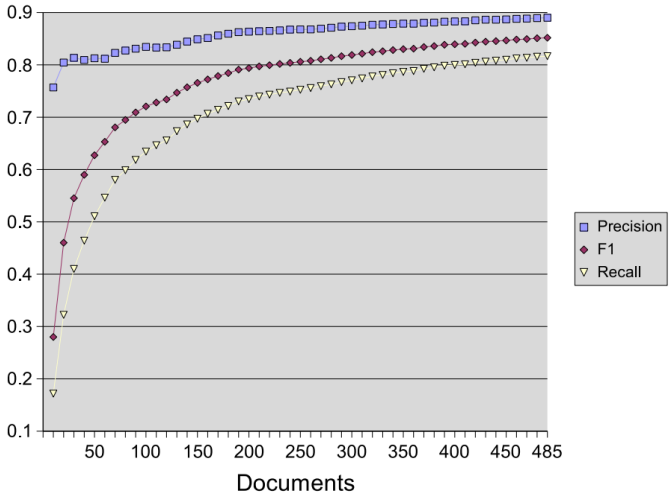

FIG. 27: Evolució de la precisió, cobertura i mesura F1 al llarg de l'entrenament interactiu. Font [Siefkes, 2005].

Una altre sistema interessant que utilitza anotació Inter-Activa és l'anomenat *Amilcare* **[Ciravegna et al., 2002a; 2002b; 2002c]**, una eina que permet l'anotació activa de documents per tasques d'EI en entorns de Gestió del Coneixement[7] (GC). Aquesta eina permet la definició prèvia d'una ontologia d'entitats i d'un corpus per ser processat, i ofereix un entorn gràfic que facilita la tasca d'anotació **[Fig. 28]**.

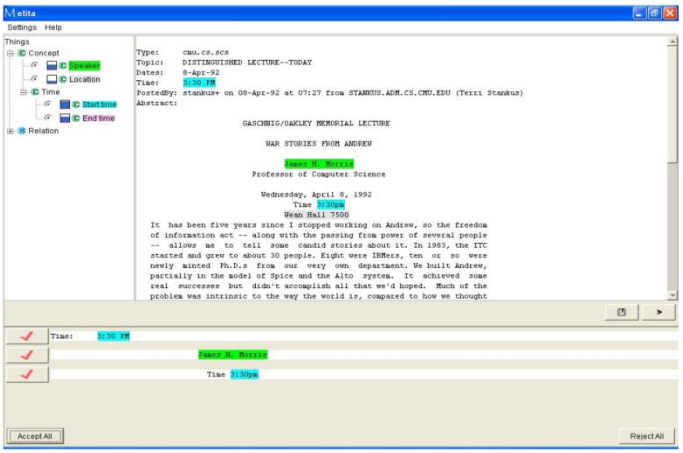

FIG. 28: Interfície gràfica d'Amilcare. Font [Ciravegna et al., 2002a]

L'eina, basada en un algorisme *batch* d'inducció de regles anomenat (LP^2) **[Ciravegna, 2001]**, permet a l'expert utilitzar inicialment l'aplicació com una simple eina d'anotació. Tot i que en un segon pla, l'algorisme va analitzant les anotacions per induir regles d'extracció equivalents. Així que les primeres regles comencen a assolir graus de confiança suficients, el sistema proposa en forma de pre-anotacions candidats d'extracció que l'expert pot validar o

[7] *Knowledge Management (KM)*

corregir. Les anotacions finals del document són enviades a l'algorisme d'aprenentatge perquè les utilitzi com a exemples amb els quals continuar el seu entrenament. A la **Fig. 29** es pot veure l'evolució de la precisió, cobertura i F1 per algunes de les informacions extretes pel sistema; el ritme d'aprenentatge és molt variable segons el tipus d'entitat que es tracti i la quantitat d'exemples necessaris per poder generalitzar l'estructura.

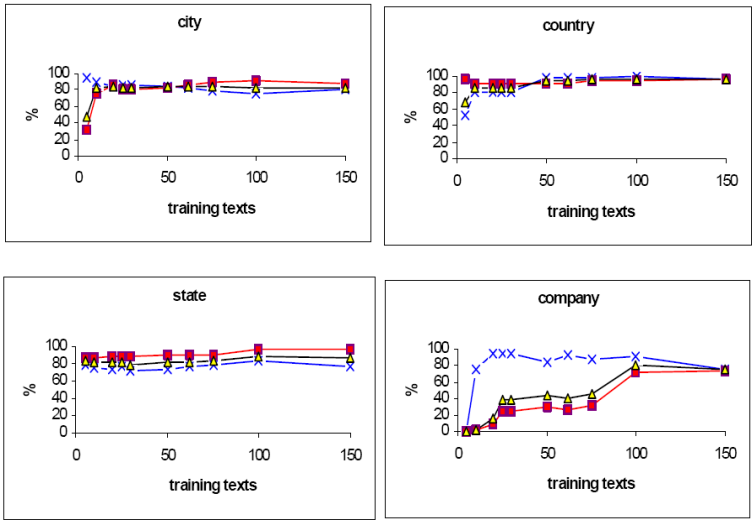

FIG. 29: Evolució de la precisió (x), cobertura (□) i F1 (Δ), per diferents entitats extretes per l'Amilcare. Font [Ciravegna et al., 2002b]

L'*Amilcare* és un prototipus que ha donat molt bons resultats; a més, la seva modularitat i capacitat d'adaptació ha permès integrar el seu motor en altres eines similars. Tot i això, el fet d'estar basat en un algorisme *batch* provoca que els recursos computacionals necessaris siguin considerables. De fet, com els mateixos autors expliquen, la necessitat que la interfície gràfica respongui de forma àgil ha forçat que l'algorisme d'aprenentatge s'hagi d'executar com un procés independent en segon pla.

Finalment, a **[Culotta et al., 2006]**, es presenta una eina similar que permet l'entrenament interactiu del que els autors anomenen *tasques de classificació estructurada*[8], que exemplifiquen amb una tasca típica d'EI. El motor d'inducció està basat en una variant del model estocàstic conegut com a Camp Aleatori Condicional[9] (CAC), especialment adaptat a la classificació de seqüències, que permet incorporar les correccions de l'usuari per re-anotar el text eficientment. A la **Fig. 30** es mostra la interfície gràfica corresponent utilitzada per a l'extracció d'informació de contactes.

[8] *Structured Classification Tasks (SCT)*
[9] *Conditional Random Fields (CRF)*

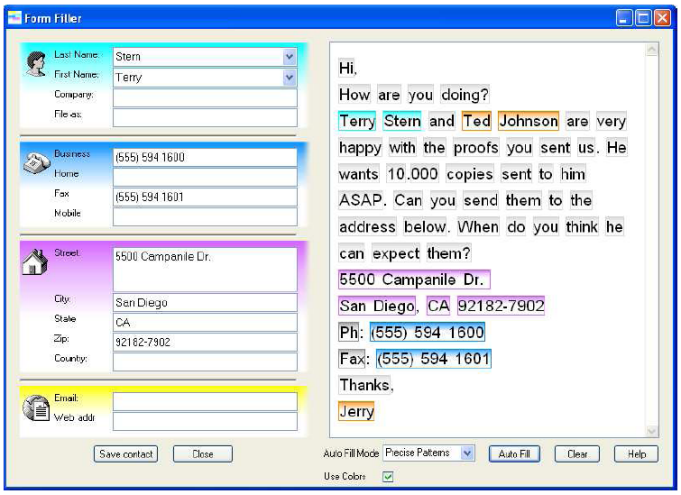

FIG. 30: Interfície gràfica del sistema inter-actiu d'extracció d'informació. Font [Culotta et al., 2006]

En aquesta eina s'ha cuidat especialment l'eficiència i usabilitat de la interfície, centrant els esforços en reduir al màxim la tasca de l'expert. Per això, una de les aportacions que presenten és el concepte de *propagació de la correcció*, conseqüència del model estocàstic que utilitzen, molt sensible a les relacions a distància entre entitats. Quan l'expert corregeix alguna de les anotacions el sistema re-avalua la resta d'anotacions, cosa que sovint suposa la correcció automàtica d'altres anotacions errònies, de manera que una única acció del revisor pot corregir múltiples anotacions a l'hora.

Una altra aportació del treball és el conjunt d'experiments que realitzen per comparar els efectes en la velocitat d'aprenentatge de diferents estratègies d'Aprenentatge Automàtic, amb l'objectiu de determinar quin sistema de selecció permet minimitzar el nombre d'anotacions. A la **Fig. 31** es mostra l'evolució del classificador per tres estratègies de mostreig: les anotacions amb menor grau de confiança, les anotacions amb menor grau d'acord en una consulta per comitè (QBC) i la selecció aleatòria d'anotacions. La simplicitat de la selecció per grau de confiança i la superioritat dels seus resultats fan que els autors la considerin l'opció més recomanable.

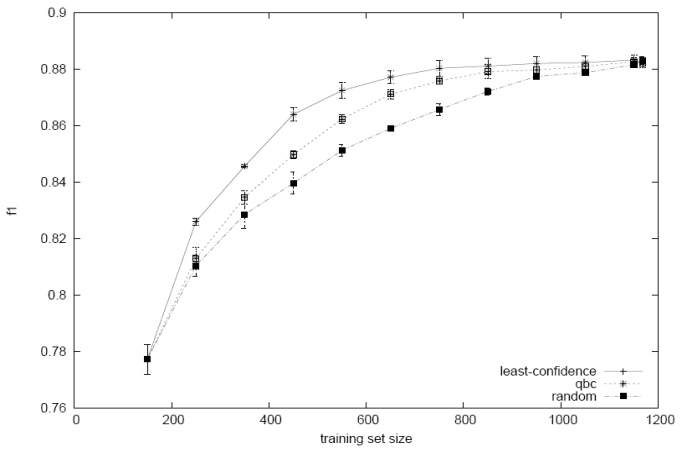

FIG. 31: Evolució del classificador (F1) segons el nombre d'exemples per tres sistemes de selecció. Font [Culotta et al., 2006]

En totes aquestes eines els autors descriuen importants reduccions de la feina necessària per revisar les anotacions automàtiques i, per tant, un augment considerable de la velocitat d'anotació dels documents. Per això no hi ha cap dubte que l'anotació Inter-Activa suposa un estalvi de recursos en aplicar-se a tasques d'EI. Queda pendent comprovar si aquests beneficis són aplicables a altres tasques de PLN on l'anotació del corpus ha de ser més exhaustiva i incloure més informació lingüística.

PART II:

L'APRENENTATGE INCREMENTAL EN PLN: ESTAT DE LA QÜESTIÓ

6 ALGORISMES INCREMENTALS

6.1 INTRODUCCIÓ

Existeixen diferents paradigmes d'Aprenentatge Automàtic (AA) i cada un d'ells ofereix una varietat d'algorismes per induir classificadors. La majoria d'aquests algorismes, al menys en la seva versió original, treballen a partir de la informació estadística derivada del conjunt de les dades d'entrenament, és a dir segueixen el paradigma *batch*. Afortunadament d'alguns d'ells s'han proposats versions que, directament o amb petites modificacions, poden utilitzar-se incrementalment. La finalitat d'aquest capítol es repassar les principals famílies d'algorismes d'AA amb l'objectiu de presentar les diferents versions incrementals que hi ha disponibles.

Històricament les tècniques d'AA s'han classificat en dos grans famílies segons si els models codificaven coneixement *simbòlic* o coneixement *subsimbòlic*. Es considera coneixement *simbòlic* aquell que es representa explícitament, tradicionalment en forma d'arbres o de regles, i coneixement *subsimbòlic* aquell que es representa holísticament en models globals que no permeten una interpretació directa, l'exemple clàssic són els pesos de les connexions d'una xarxa neuronal. Però aquesta divisió a anat perdent sentit a mesura que es desenvolupaven models híbrids basats en regles probabilístiques o en combinacions d'arbres de decisió.

Una altre divisió més recent classifica els algorismes segons si es basen en *tècniques estadístiques* o en *raonament inductiu*. Però, com explica **[Roth & Zelenko, 1998]**, tots els algorismes d'aprenentatge són essencialment estadístics, ja que intenten fer generalització inductiva a partir dels exemples observats, i després utilitzar-la per fer inferència respecte les noves observacions **[Màrquez, 2001]**.

Però com acostuma a passar aquest tipus de divisions són més o menys arbitràries i s'entrecreuen, ja que estan basades en criteris de difícil aplicació. Per això en aquest capítol s'han agrupat els algorismes segons els paradigmes generals als que pertanyen, i aquests s'han ordenat **[Daelemans, 2005]** segons la naturalesa del model induït; començant pels més concrets i propers a les dades, i acabant amb els més difosos i allunyat de les dades.

En les diferents seccions s'introdueix una explicació general dels principis en què es basa el paradigma i seguidament es presenten els algorismes o variants capaços d'induir incrementalment models classificadors a partir d'una seqüència d'exemples.

6.2 APRENENTATGE BASAT EN MEMÒRIA

L'aprenentatge basat en memòria **[Aha et al., 1991]** és el paradigma que utilitza un model més proper a les dades, és conegut per diferents variants[1] del mateix nom i és l'exemple prototípic del que s'anomenen algorismes *lazy*.

Els algorismes *lazy* van néixer inspirats pels primers treballs sobre reconeixement de patrons i pels models psicològics basats en prototipus **[Aha, 1995]**, i prenen el nom del principi en el que es basen: reduir al màxim el processament dels exemples durant l'entrenament i posposar tota la feina que sigui possible a la fase de classificació.

Això ho aconsegueixen mitjançant un entrenament molt simple consistent en un emmagatzemament passiu dels exemples, i una classificació basada en la cerca d'exemples similars. El funcionament general és molt simple, l'entrenament consisteix en memoritzar tots els exemples etiquetats, sense cap mena de modificació en la seva representació. A l'hora de classificar un nou exemple, es recuperen de la memòria els exemples més similars i es determina la seva classe segons la classe majoritària en aquest grup. El nucli de l'algorisme es troba en la definició d'una funció de distància que equilibri les influències dels diferents atributs i que normalitzi les distàncies.

En la majoria d'algorismes d'aprenentatge l'abstracció, o generalització, es realitza tan bon punt es rep l'exemple d'entrenament, extraient la informació associada i oblidant immediatament l'exemple individual. Però en els algorismes *lazy*, com els basats en memòria, la generalització es realitza durant l'etapa de classificació. En certa manera els algorismes *lazy*, posposen la generalització al moment de la classificació. Tot i que la distribució de probabilitats per cada classe al llarg de l'espai d'atributs és una informació que es troba implícita, des del primer moment, en la combinació del conjunt d'exemples memoritzats i la mètrica de similitud utilitzada. En certa manera el càlcul d'aquesta explicitació seria tan costosa (requeriria calcular les distàncies de tots els exemples emmagatzemats per cada un dels punts de l'espai) que és més eficient calcular-la únicament pels punts on es troben els exemples que cal classificar.

Una altre característica que fa diferent a aquest paradigma és que pertany al que es coneix com a *models no-paramètrics* **[2.2.4 Característiques de les Tasques Lingüístiques]**. Els models *no-paramètrics* no fan cap assumpció sobre la forma de l'espai de solucions i per tant els seus models no són parametritzables mitjançant un conjunt finit de paràmetres. Aquesta absència d'assumpcions suposa també una manca de biaixos, el que els hi permet modelar més fidelment qualsevol funció de classificació; ja que en certa manera la funció de classificació està reflectida implícitament a les pròpies dades.

[1] Aprenentatge també conegut com: basat en memòria, basat en casos, basat en exemples, basat en instàncies, basat en similaritat, basat en distància, ponderat localment, raonament basat en casos, ... **[Aha, 1995; Daelemans, 2005]**

Els punts forts d'aquest paradigma són el cost computacional utilitzat durant l'entrenament, pràcticament inexistent, i especialment la seva idoneïtat per resoldre problemes de PLN. Per contra, tot i la seva incrementalitat intrínseca, els recursos d'emmagatzemament creixen linealment amb el número d'exemples, com també ho fa el cost computacional necessari per realitzar la classificació. Ja que en les variants més simple, és necessari calcular la distància entre l'instancia desconeguda i cada un dels exemples memoritzats; un cost gens menyspreables si es vol aplicar a corpus massius o entorns interactius.

IDONEÏTAT EN PROBLEMES DE PLN

Alguns autors consideren que aquest paradigma és ideal per tasques de PLN ja que al no simplificar les dades, ni eliminar informació, no prescindeix dels exemples corresponents a les excepcions. A **[Daelmans et al., 1999]** s'argumenta a favor de la idoneïtat d'aquest paradigma a l'hora de tractar problemes de PLN, ja que al no pressuposar cap funció concreta i al emmagatzemar tant els casos generals com les excepcions en qualsevol nivell lingüístic, pot modelar millor les irregularitats pròpies del llenguatge.

Altres motivacions lingüístiques que justifiquen la seva utilització són l'absència d'una separació nítida entre els casos regulars i els irregulars del llenguatge natural, la simplicitat del raonament per analogia respecte l'obtenció de regles, i l'adaptabilitat d'aquest model comparada amb la rigidesa dels basats en regles **[Màrquez, 2001]**. Per això, aquest paradigma ha estat aplicat en múltiples ocasions en diferents tasques de PLN: anàlisi fonològic, anàlisi morfològic, etiquetat morfosintàctic, desambiguació de dependències de sintagmes preposicionals[2] i anàlisi sintàctic superficial[3].

Finalment, un avantatge especialment interessant és la possibilitat d'utilitzar el principi de similitud com a mètode per suavitzar les estimacions d'esdeveniments poc freqüents. A causa de la natura i evolució del llenguatge natural, qualsevol nivell lingüístic pot caracteritzar-se mitjançant uns quants models regulars i moltes sub-regularitats i excepcions. Aquesta *topologia* dificulta la feina als algorismes *eager*, contraposats als *lazy*, que durant el procés de generalització no poden diferenciar el que són excepcions i el que és simplement soroll. Aquest avantatge de l'aprenentatge basat en memòria és encara més marcat en l'aprenentatge incremental.

6.2.1 TAULES DE CONSULTA

El model més simple possible basat en memòria són les *taules de consulta*[4], en aquest models no es produeix cap mena de generalització, ni implícita no explícita, simplement es tracta de memorització.

---

[2] *PP-Attachment disambiguation*
[3] *Chunking*
[4] *Look-up tables*

Durant l'*entrenament* l'algorisme va emmagatzemant en una taula tots els exemples, incloent la seva classe. Durant la classificació busca en la taula el mateix exemple i si el troba retorna la classe corresponent, sinó el troba aplica alguna heurística per omissió, per exemple retornar la classe més freqüent. Es pot considerar que es tracta d'un aprenentatge basat en memòria en el qual la mesura de distància és un valor binari: coincidència exacte o no-coincidència.

Malgrat la seva simplicitat, en algunes tasques de PLN **[Daelemans, 2005]** pot donar resultats sorprenentment alts. Els seus resultats són inversament proporcionals a la mida de l'espai d'atributs que codifica als exemples, ja que és especialment sensible al problema de *data sparseness*. En la majoria dels casos acostuma a obtenir una gran precisió a costa d'una molt baixa cobertura provocada per la seva nul·la capacitat de generalització.

6.2.2 *K-NEAREST NEIGHBOUR*

El K-NN (*K-Nearest Neighbour*[5]) és l'algorisme clàssic de la família dels basats en memòria. El seu naixement es situa en els orígens del reconeixement de patrons **[Fix & Hodges, 1951]**, tot i que no va ser utilitzat com a classificador fins uns anys després **[Cover & Hart, 1967]**, però la seva maduresa com a algorisme d'AA arriba amb el treball de **[Aha et al., 1991]**.

L'entrenament consisteix simplement en l'emmagatzemament dels exemples en una memòria interna. A l'hora de classificar una nova instància, es recuperen de la memòria els *K* exemples més similars i es selecciona la classe majoritària present en els exemples recuperats. Cal recordar que, en aquest paradigma, la semblança es inversa a la distància i, per tant, els exemples més semblants són els mes propers.

Així doncs, per classificar l'exemple $\boldsymbol{x}=\langle x_1, ..., x_m\rangle$ cal comparar-lo amb cada un dels exemples emmagatzemats $\boldsymbol{y}=\langle y_1, ..., y_m\rangle$ i calcular la distància que els separa. La funció utilitzada per calcular aquesta distància és un dels paràmetres, l'altre és el valor de *K*, que defineixen el comportament de l'algorisme. Per atributs numèrics reals la distància és simplement la distància euclidiana, ponderant els atributs amb els pesos w_i:

$$\Delta(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m}(w_i\,\delta(x_i,y_i))^2} = \sqrt{\sum_{i=1}^{m}(w_i(x_i-y_i))^2}$$

Per atributs simbòlics, s'acostuma a utilitzar la distància de Hamming, també coneguda com de solapament, i consisteix en una suma ponderada de les distàncies individuals dels atributs:

[5] *K-veins-més-propers*

$$\Delta(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{m} w_i \, \delta(x_i, y_i),$$

on w_i és el pes associat a l'atribut *i*; en la versió més senzilla tots els pesos valen 1, i $\delta(x_i, y_i)$ és la distància individual, definida pels atributs simbòlics com:

$$\delta(x_i, y_i) = \begin{cases} 0, & x_i = y_i \\ 1, & x_i \neq y_i \end{cases}$$

Una vegada definides les mètriques a utilitzar només cal aplicar-les a tots els exemples memoritzats respecte l'instancia que es vol classificar. S'obtenen els *k* exemples més propers i se li assigna la classe majoritària ponderada, on cada exemple vota amb un pes proporcional a la seva similitud. Habitualment es prenen valors senars per *k* amb l'objectiu d'evitar empats, com 1, 3, o 5. Encara que en la versió *batch,* el valor òptim per aquest paràmetre ha de d'estimar-se *a priori*, la versió incremental permet actualitzar-lo dinàmicament **[8.4 Optimització de Paràmetres]**.

6.2.3 IB-1, IB-K, K-STAR

En l'aplicació d'aquest paradigma al PLN destaca la recerca de Daelemans, i el seu equip en Antwerp i Tilburg, pel desenvolupament del TiMBL (*Tilburg Memory-based Learning Environment*). Per resoldre els problemes d'espai a l'hora de tractar problemes amb gran quantitat d'exemples, han desenvolupat una estructura de dades anomenada *IGTree* **[Daelemans et al., 1997]** que permet comprimir i indexar els exemples sense pèrdua d'informació i amb una important reducció de l'espai necessari.

A més, s'han desenvolupat altres variants prometedores basades en la ponderació dels exemples i en la reducció de la memòria d'emmagatzemament mitjançant la selecció d'exemples. El punt de partida és una implementació directe del K-NN coneguda com a IB1. Aquest implementació utilitza una funció de distància típica, normalitza els trets numèrics per que tinguin una mateixa escala i, per tant, el mateix pes, i assumeix que els valors no-definits suposen una distància màxima respecte l'entitat a classificar.

El grup de Daelemans ha aplicat aquesta variant IB-1 a un gran numero de tasques de PLN **[Daelenas et al., 1999]**. I altres investigadors l'han provat en sistemes de conversió grafema-fonema **[Lehnert, 1987; Wijters, 1991]**, desambiguació semàntic de les paraules **[Ng & Lee, 1996; Fujii et al., 1998]** o anàlisis sintàctic depenent del context **[Simmons & Yu, 1992]**.

A partir d'aquesta versió bàsica introdueixen diferents modificacions per millorar els seus resultats **[Aha et al., 1991]**. La primera variant, IB2, presenta uns menors requeriments d'espai per emmagatzemar els exemples, això ho aconsegueix restringint l'emmagatzemament només a aquelles instàncies que no ha pogut classificar correctament. La segona variant, IB3, intenta millorar la seva tolerància al soroll. Periòdicament revisa la memòria d'exemples i esborra aquells exemples que són redundats o que han sigut mal

predictors. Això permet reduir l'espai necessari per emmagatzemar el model, però mantenir aquells exemples que històricament han demostrat ser informativament útils. Finalment a **[Aha, 1992]** descriu altres variants que permeten tractar atributs irrellevants, IB4, i atributs nous, IB5.

Uns anys després, **[Cleary & Trigg, 1995]** proposa una nova variant, la K-Star, que utilitza una distància basada en l'entropia com a alternativa a la distància euclidiana tradicional. L'apropament utilitzat per calcular la distància entre dos instancies està motivat per la teoria de la informació. La intuïció que hi ha darrera considera que la distancia entre dos instancies pot definir-se com la complexitat necessària per transformar una en l'altre. Seguint aquest punt de vista pot utilitzar-se el que es coneix com a distància Kolmogorov, definida com la longitud de la cadena d'edició més curta que uneix les dues instàncies. Però el resultat és una mesura de distància que és molt sensible als petits canvis en l'espai de trets i que no permet resoldre correctament l'*smoothness problem*. Per això els autors proposen una nova mesura de distància coneguda com a *K-Star* que millora els resultats i permet utilitzar-se satisfactòriament en un algorisme de *K-NN*.

6.3 INDUCCIÓ D'ARBRES DE DECISIÓ

La inducció d'arbres de decisió representa la família més coneguda dins de l'IA clàssica a l'hora d'induir classificadors **[Quinlan, 1993; Cohen, 1995]**. Els arbres de decisió són models que representen regles mitjançant estructures jeràrquiques seqüencials. Aquestes regles divideixen recursivament l'espai d'atributs amb l'objectiu d'obtenir la classe majoritària en una regió determinada. En un arbre de decisió cada camí des de l'arrel fins als elements terminals representa una regla conjuntiva de classificació. En aquest camí, cada node conté una condició sobre un atribut i una branca per cada un dels valors possibles. Els elements terminals de les branques, coneguts com *fulles*, determinen la classe a la que pertanyen les instàncies que compleixen les condicions precedents.

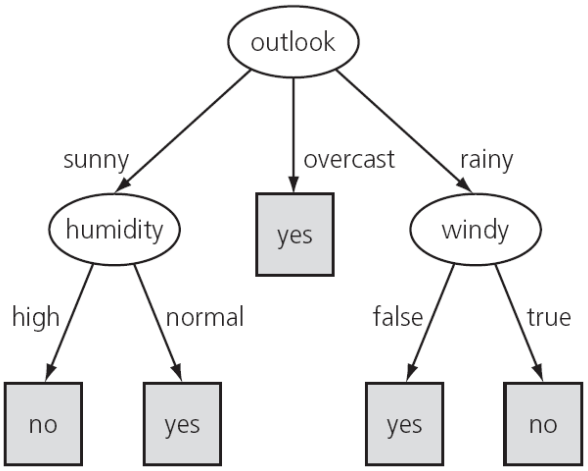

FIG. 32: Fragment típic d'arbre de decisió. Font [Witten & Frank, 2005].

El paradigma d'aprenentatge d'arbres de decisió pertany als coneguts com algorismes *eager*, contraposats als *lazy*, en els quals l'abstracció es realitza durant l'entrenament. La inducció es realitza mitjançant la divisió recursiva del conjunt d'exemples, el criteri de fraccionament es basa en agrupar exemples que comparteixen un mateix parell atribut-valor. El procés recursiu finalitza quan el subgrup corresponent és homogeni, és a dir, quan tots els seus exemples pertanyen a una mateixa classe. El punt crític es troba en el criteri utilitzat per seleccionar l'atribut més discriminador en cada node, és a dir, aquell que divideix millor els exemples restants. A la **Fig. 33** es mostra un algorisme genèric d'inducció d'arbres mitjançant la tècnica de divisió recursiva. Com la cerca exhaustiva de l'arbre òptim és un problema computacionalment intractable, s'han d'utilitzar heurístiques de selecció, normalment heurístiques estadístiques o basades en la teoria de la informació.

Given a set of examples T

- If T contains one or more cases all belonging to the same class C_j, then the decision tree for T is a leaf with category C_j.
- If T contains different classes then
  - Choose a feature, and partition T into subsets that have the same value for the feature chosen. The decision tree consists of a node containing the feature name, and a branch for each value leading to a subset of T
  - Apply the procedure recursively to subsets created this way.

FIG. 33: Algorisme genèric d'inducció d'arbres. Font [Daelemans, 2005].

Els arbres de decisió s'han utilitzat àmpliament en tasques de PLN: l'etiquetat morfosintàctic, la desambiguació semàntica de paraules, l'anàlisi sintàctic, la classificació de documents, el resum automàtic, la resolució de coreferència i, fins i tot, la traducció automàtica **[Màrquez, 2001]**.

Segons **[Magerman, 1995]**, l'expressivitat dels arbres de decisió és equivalent a la dels models d'*n-grames* interpolats. Però si en aquests els paràmetres que cal estimar creix exponencialment amb *n*, la mida del context, la mida dels arbres només depèn de la quantitat d'exemples. A més, com els atributs no informatius són descartats automàticament, els arbres de decisió poden tractar contexts més amplis, cosa que els hi permet resoldre millor les dependències a llarga distància **[Daelemans, 2005]**.

Finalment, una de les característiques més destacables dels arbres de decisió és la llegibilitat dels seus models. La interpretació d'un arbre de decisió és relativament trivial, ja que cada una les seves branques pot transformar-se directament en una regla mitjançant la conjunció de les condicions presents en els seus nodes.

6.3.1 ID3 I C4.5

Un dels primers algorismes d'inducció d'arbres de decisió és l'ID3 (*Iterative Dichotomiser-3*) **[Quinlan, 1986]**, basat en els principis de divisió recursiva del conjunt d'exemples,

explicats al punt anterior. A **[Quinlan, 1993]** es va presentar una versió millorada del ID3, el C4.5, que era més eficient i podia tractar trets numèrics i valors no-definits.

Aquests algorismes construeixen un arbre de decisió que classifica entre exemples i no-exemples d'un concepte determinat. Per cada un dels atributs es mesura el grau de discriminació que permet obtenir, mesurat amb l'entropia, i es selecciona el tret més informatiu. Aquest tret és utilitzat com a node arrel i per cada valor possible introdueix una branca. Per cada una de les branques es repeteix el procés, generant un sub-arbre que classifiqui el subconjunt d'exemples que compleixin les condicions precedents.

Tot i que a **[Schlimmer & Fisher, 1986]** es va suggerir la seva utilització incrementalment per un mètode de *força bruta*: re-calculant un nou arbre cada vegada que es processava un nou exemple **[4.3.1 Incrementalitat Funcional]**, tant l'ID3 com el C4.5 són algorismes *batch*. Afortunadament altres autors van utilitzar l'entorn de l'ID3 per introduir-hi modificacions que els permetessin processar els exemples individualment de manera eficient. Aquestes variants són les que es presenten en el següent punt.

6.3.2 ID4, ID5, ID5R i ITI

A **[Schlimmer & Fischer, 1986]** es va proposar el primer intent de desenvolupar una versió incremental de l'ID3, anomenat ID4. Malgrat ser incremental, aquest algorisme pràcticament reconstrueix l'arbre de decisió cada vegada que rep un exemple. Però te l'avantatge que emmagatzema les distribucions de classes a cada node en una taula que resumeix el nombre d'exemples positius i negatius per cada valor. Cosa que li permet actualitzar eficientment les mesures d'entropia per cada node i cada atribut. A l'hora d'actualitzar l'arbre amb un nou exemple només ha d'actualitzar aquestes distribucions i re-construir un sub-arbre si detecta un tret que sigui millor discriminant.

Una de les característiques de l'ID4 és que presenta una gran sensibilitat a l'ordre en què es reben els exemples. Si en un determinant punt del procés hi ha diferents atributs amb graus similars valors de discriminació, l'ordre precedent dels exemples pot decantar la decisió en una direcció o en una altre. El resultat és que diferents seqüències d'un mateix conjunt d'exemples dona lloc a arbres diferents **[4.5 Dependència de l'Ordre dels Exemples]**. Un altre limitació més important era que, degut a que descartava sub-arbres al substituir el tret d'un determinat node, no era capaç d'aprendre determinats conceptes.

A **[Utgoff, 1988]** va proposar l'ID5, que no descartava sub-arbres, però continuava generant arbres dependents de la seqüència d'entrenament. Poc després, a **[Utgoff, 1989]** es desenvolupa l'ID5R, una versió incremental de l'ID3 que és independent de l'ordre d'entrenament i a més garanteix la generació d'arbres idèntics als generats per l'ID3. Igual que l'ID4 emmagatzema informació estadística a nivell de node, però al detectar un canvi en la discriminalitat dels atributs no substitueix els sub-arbres, sinó que els actualitza recursivament. Les principals limitacions són que no pot tractar trets numèrics, valors no-definits ni tasques multiclasse. Finalment, a **[Utgoff, 1997]** es presenta l'ITI (*Incremental Tree*

Inducer[6]), també incremental i independent de l'ordre, però que a més resolt les mancances del ID5R: pot treballar amb trets numèrics, valors no definits i tasques multiclasse.

6.4 INDUCCIÓ DE REGLES

Les regles i els arbres de decisió són dues representacions equivalents d'un mateix coneixement, i en molts aspectes poden tractar-se com una mateixa família. Per extreure les regles existents en un arbre de decisió n'hi ha prou amb extreure una regla per cada branca de l'arbre. Les condicions de cada node de la branca formen una conjunció de condicions, i el node terminal la conclusió de la regla.

La inducció de regles es basa en la mateixa idea de generalització que s'aplica en els arbres de decisió: analitzar les diferències i similituds entre els exemples per crear representacions més generals que les englobin, memoritzar la distribució de classes per cada generalització i oblidar els exemples individuals.

Les regles poden ser binàries o probabilístiques, en aquest darrer cas inclouen pesos de manera que poden estimar la confiança o marge d'error de les seves decisions. I en qualsevol dels casos ofereixen avantatges similars a l'arbre de decisió: la inspeccionabilitat dels models, que suposa la obtenció de coneixement explícit, i els pocs recursos de memòria necessaris per emmagatzemar un model.

6.4.1 1-RULE

L'algorisme més simple d'inducció de regles és el *1-Rule*: un algorisme que genera regles simples que inclouen una única condició referida a algun dels trets dels exemples. És un mètode molt simple, utilitzat com a línia de referència, que dona resultats molt més bons del que seria previsible.

L'origen d'aquest article es troba a **[Holte, 1993]** on presenta un estudi exhaustiu del comportament d'aquest algorisme en setze conjunts de dades utilitzats habitualment pels investigadors d'AA a l'hora d'avaluar els seus algorismes. Els resultats van ser sorprenentment bons, "fins i tot vergonyosament bons" en paraules de **[Witten & Frank, 2005]**. Totalment comparables amb els de la resta d'algorismes del moment, amb una precisió de només uns pocs punts inferior. A més, els seus models formats per regles simples, eren considerablement més petits. Per això, tot i la seva simplicitat, és interessant utilitzar-lo davant d'un nou problema per mesurar el seu grau de dificultat.

L'algorisme genera una regla per cada un dels atributs i li afegeix una branca per cada un dels valors possibles. A l'hora de classificar un nou exemple s'apliquen totes les regles i per cada una s'obté la classe majoritària de la branca corresponent. Si durant la inducció de les regles s'han comptabilitzat els encerts i errors de cada branca, és senzill determinar les seves

[6] *Inductor d'Arbres Incremental (IAI)*

taxes d'error, i per tant assignar la classe d'aquella regla amb un error menor. A la **Fig. 34** es mostra el pseudo-codi de la versió *batch* d'aquest algorisme.

```
For each attribute,
  For each value of that attribute, make a rule as follows:
    count how often each class appears
    find the most frequent class
    make the rule assign that class to this attribute-value.
  Calculate the error rate of the rules.
Choose the rules with the smallest error rate.
```

FIG. 34: Pseudo-codi per l'algorisme 1Rule. Font [Witten & Frank, 2005].

Adaptar aquest algorisme a un funcionament incremental és senzill, només cal generar dinàmicament una taula de valors per cada un dels atributs, i actualitzar els comptadors de les classes per cada un d'ells. Durant la classificació es pot accedir directament a cada un dels comptadors i seleccionar el que tingui l'error més baix.

6.4.2 TAULES DE DECISIÓ

Les *Taules de Decisió* [7] **[Kohavi, 1995]** són unes representacions molt senzilles que emmagatzemen simplificacions dels exemples, juntament amb les seves etiquetes. La simplificació es realitza seleccionant un subconjunt dels atributs utilitzats per descriure els exemples, i ignorant la resta.

En certa manera són similars a les *Taules de Consulta* **[6.2.1 Taules de Consulta]**, però el fet de que els exemples es modifiquin eliminant alguns dels seus trets fa que tècnicament no pugui considerar-se un sistema '*lazy*' basat en memòria. A més, el procés de simplificació suposa un cert grau de generalització en les representacions obtingudes, i per tant poden interpretar-se com regles.

A **[Kohavi, 1995]** es van realitzar diferents experiments per estudiar les seleccions de trets fetes durant la inducció d'arbres de decisió. Els resultats van mostrar que en molts casos els arbres induïts eren casi complets, és a dir, que les diferents branques representaven pràcticament totes les combinacions dels trets seleccionats. Per això va concloure que, en aquests casos, era possible estalviar-se la inducció de l'arbre i simplement tabular totes les combinacions dels diferents trets-valors, donant lloc a una *Taula de Decisió*.

L'entrenament d'aquests models és directa, només cal simplificar els exemples d'entrenament, eliminant els trets no seleccionats, i emmagatzemar-los en una taula juntament amb la seva classe. Durant la classificació es consulta la taula i s'obtenen totes les coincidències que encaixin amb l'exemple desconegut. Finalment, se li assigna la classe majoritària entre les coincidències; si no hi ha cap coincidència, es retorna la categoria majoritària global. Aquest algorisme pot utilitzar-se incrementalment sense haver de fer cap

[7] *Decision Table Majority (DTM)*

adaptació, ja que tant la simplificació com l'emmagatzemant s'apliquen als exemples individualment.

El punt crític d'aquest algorisme és la selecció dels trets que s'utilitzaran per indexar la taula, la dificultat augmenta si la decisió s'ha de fer de manera incremental. Una solució seria utilitzar un mètode de selecció aleatòria i combinar diferents taules mitjançant un sistema de votació **[7.3 Sistemes de Votació]**.

6.4.3 LLISTES DE DECISIÓ

Les *Llistes de Decisió*[8] **[Rivest, 1987]** són un sistema d'inducció de regles que genera un model format per una llista ordenada de regles conjuntives; aquestes regles s'estructuren en funció d'una relació de condicions *if-else*.

A l'hora de classificar un nou exemple s'avaluen les regles una darrera l'altre, i la primera regla aplicable a l'exemple s'utilitza per assignar-li la classe corresponent. La inducció parteix d'una inicialització aleatòria de la llista de decisió i d'una modificació incremental de les posicions que ocupen les regles. Durant l'entrenament s'apliquen a l'exemple les regles de la llista fins trobar la primera regla que classifiqui erròniament, aleshores es modifica la seva posició baixant-la un nivell. La repetició d'aquest procediment al llarg dels exemples acaba ordenant la llista de manera que maximitza la seva capacitat de classificació. A **[Blum, 1996]** s'inclou una explicació detallada d'aquest algorisme.

En dominis d'alta dimensionalitat i amb pocs exemples, les Llistes de Decisió poden donar millors resultats que els arbres, que tendeixen a fragmentar les dades en excés. Cal tenir en compte que es basen en la hipòtesi de que és possible classificar els exemples mitjançant regles senzilles i independents, una assumpció que no és vàlida per a tots els problemes.

Segons **[Màrquez, 2001]** són el sistema d'inducció de regles més utilitzat en problemes de PLN, i han estat aplicades en tasques com la resolució de l'ambigüitat semàntica de les paraules o en la selecció lèxica en traducció automàtica.

6.4.4 RIPPLE DOWNS RULES

Les *Ripple Down Rules* (*RDR*) **[Horn et al., 1985]** van néixer com una metodologia incremental d'adquisició de coneixement, a partir de l'experiència directa de l'actualització manual d'un sistema expert **[Compton & Jansen, 1990]**. Tot i que la metodologia també és vàlida per la inducció incremental de sistemes de regles.

En essència consisteix en aplicar el sistema de regles a cada nou exemple, si la classificació és correcta el sistema es deixa intacte. Si la classificació és incorrecte, s'afegeix una excepció en la condició que va fallar de tal manera que el nou exemple sigui classificat correctament.

[8] *Decision Lists (DL)*

Aquest sistema d'actualització incremental utilitza modificacions locals que impedeixen la introducció d'efectes no desitjats en altres branques de l'arbre.

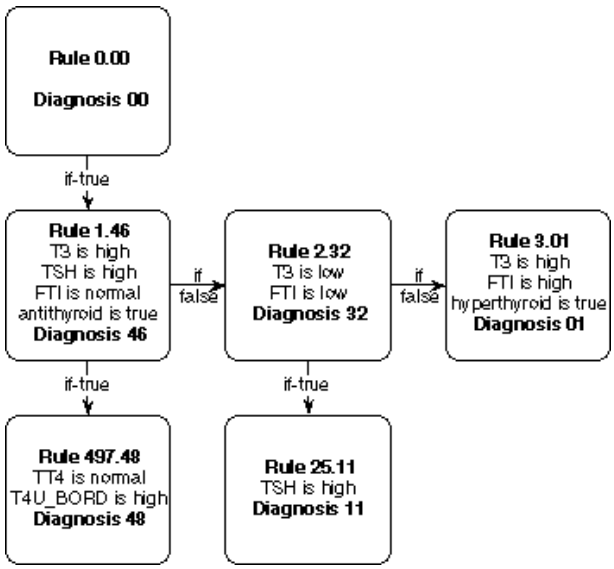

FIG. 35: Exemple de regles RDR en el sistema expert original. Font [Gaines & Compton, 1995].

Les regles s'estructuren de manera similar a un arbre de decisió, on cada node té una condició, una conclusió i dos connexions a altres nodes, de la següent forma:

```
IF (cond1 AND cond2 AND ... AND condN)
THEN conclusion
     EXCEPT ...
ELSE ...
```

Finalment, aclarir que existeix un algorisme d'inducció de regles amb un nom semblant, RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) **[Cohen, 1995]**, que malgrat la utilització de la paraula *incremental* en el seu acrònim, no deixa de ser un algorisme *batch* iteratiu que requereix accedir a la totalitat de les dades per poder entrenar-se.

6.4.5 GEM , AQ11 I AQ15

L'AQ15 **[Hong et al., 1986]** és un algorisme incremental d'inducció de regles fruit de l'evolució de l'algorisme GEM, i anteriorment de l'algorisme AQ11, dels mateixos autors. Aquest algorisme aprèn regles de decisió a partir d'exemples, contra-exemples i, a més, d'altres regles. I això és el que li permet aprendre incrementalment, ja que pot obtenir noves regles a partir de la recombinació del nou exemple amb les regles ja existents.

A l'hora d'aprendre les regles, el programa utilitza: a) coneixement previ en forma de regles i conceptes que el programa ja coneix, b) definicions de les representacions dels exemples amb els seus tipus, i c) un criteri de preferència que avalua hipòtesis candidates i permet triar entre dos hipòtesis competidores. Com en altres algorismes d'inducció, cada exemple d'entrenament és una instància d'un concepte i està associat a la seva classe correcta. I les

regles de decisió generades s'expressen com descripcions simbòliques que impliquen restriccions dels atributs.

A més, les regles són optimitzades mitjançant criteris definits per l'usuari i avaluades automàticament segons la tasca a realitzar. L'usuari, fins i tot, pot especificar hipòtesis inicials que seran introduïdes en el procés d'aprenentatge incremental, de tal manera que l'algorisme les anirà millorant fins que siguin consistents amb els exemples d'entrenament **[Hong et al., 1986]**.

6.4.6 PDL, PDL2 & ILA

Una altra família d'algorismes incrementals d'inducció de regles està formada per diferents algorismes desenvolupats per Giraud-Carrier. El primer algorisme de la sèrie va ser el PDL2 (*Precept-Driven Learning Algorithm*) **[Giraud-Carrier & Martinez, 1994a; 1994b]**, un sistema inductiu que combina coneixement previ, en forma del que anomena *precepts*, amb exemples tradicionals. El PDL2 modela el seu coneixement en forma de xarxa de nodes que va adaptant a mesura que adquireix coneixement mitjançant la incorporació i eliminació dinàmica de nodes. L'aprenentatge és incremental, i tot i que el seu resultat és depenent de l'ordre dels exemples, tant els exemples com els *precepts* són analitzats una sola vegada.

La representació utilitzada segueix el llenguatge clàssic basat en parells d'atribut-valor, amb la incorporació d'un símbol especial pels valors no-definits. Els exemples tenen tots els seus atributs definits amb algun valor i els *precepts* tenen algun tret sense definir. També pot tractar la no-monotonicitat de forma natural, i implementar raonament per defecte gràcies als *precepts*. Les excepcions són emmagatzemades i se'ls hi dóna prioritat durant el raonament, el que li permetent tractar l'herència sense conflictes. Finalment, les situacions d'inconsistència les resolt mitjançant el comptatge d'exemples, amb una procés equivalent a una votació per pluralitat **[7.3.1 Votació sense Pes]**.

A partir d'aquest algorisme va proposar un altre, l'ILA (*Incremental Learning Algorithm*) **[Giraud-Carrier & Martinez, 1995; Giraud-Carrier, 2000]**, que conserva la combinació de característiques d'un sistema basats en regles i d'un sistema basat en exemples. Però en aquest cas hi ha una major similitud entre els processos d'aprenentatge i els processos de classificació. A més, es millora l'eficiència mitjançant un sistema que permet limitar les actualitzacions als nodes seleccionats sense haver d'ampliar-la a la resta. En aquest algorisme la seva capacitat de raonament està més limitada, però es més capaç d'aprendre a partir dels exemples. La generalització dels conceptes la realitza mitjançant hiper-plans en comptes d'hiper-rectangles; i si no és capaç d'induir regles a partir dels exemples el seu funcionament és equivalent a una aprenentatge simplificat basat en memòria.

El seu darrer algorismes, és l'i-AA1* (*Incremental-Adaptative Algorithm 1*) **[Giraud-Carrier & Martinez, 2006]** una versió incremental del seu antecessor AA1* **[Martinez & Vidal, 1988]**. A diferència dels anteriors, aquest algorisme està orientat a la classificació binària de patrons binaris. Però manté moltes de la resta de característiques: combinació de dades

amb coneixement previ, propagació de l'actualització a traves dels nodes, possibilitat d'utilitzar trets com no-definits, etc ... Tot i així, no és clar que pugui adaptar-se fàcilment al tractament de tasques de PLN.

6.4.7 WAVE

El WAVE **[Aseltine, 1999]** és un altre algorisme que indueix regles a partir de la generalització dels exemples d'entrenament, però a diferencia del seu inspirador, CRYSTAL **[Soderland, 1995]**, ho fa incrementalment.

L'algorisme resolt les limitacions de l'aprenentatge incremental mitjançant l'actualització d'una jerarquia de regles de generalització. La utilització d'una jerarquia permet trobar les regles de manera eficient, i accelera la classificació ja que evita haver d'aplicar totes les regles a cada instància. A més permet equilibrar la precisió i cobertura de les regles sense haver-se de re-entrenar. Els experiments van demostrar que els resultats obtinguts eren equivalents als d'altres algorismes *batch*. Les **Fig. 36** i **Fig. 37** mostren els algorismes recursius utilitzats per emmagatzemar un nou exemple i per aplicar-li les regles.

store(A, B)

1. If A does not cover B, return *failure*.
2. Update reliability of each prediction in A based on annotations in B.
3. Recursively store B under the children of A.
4. If B is *not* successfully stored under a child of A:
   (a) Unify B with each child of A.
   (b) Add the valid unifications to the children of A.
   (c) Add B to the children of A.
   (d) Reorganize the children of A to preserve hierarchy invariants.
5. Return *success*.

FIG. 36: Pseudo-codi de l'emmagatzemament d'una nova instància en l'algorisme WAVE. Font [Aseltine, 1999].

extract(R, I)

1. If R does not cover I, return *failure*.
2. For each prediction p in R:
   (a) Calculate E_p, the error estimate for p.
   (b) If $E_p \leq error\ tolerance$, predict p for I.
3. For each child C of R, call $extract(C, I)$.
4. Return *success*.

FIG. 37: Pseudo-codi de l'aplicació de les regles a una nova instància. Font [Aseltine, 1999].

6.5 MODELS ESTADÍSTICS

Històricament aquests models van néixer del camp de l'estadística i de la teoria de la informació, tot i que han acabat integrant-se naturalment en el camp de l'AA.

La principal característica dels models estadístics és que generalitzen el coneixement implícit dels exemples en forma de mesures estadístiques que els descriuen, i de les correlacions entre ells o els seus atributs. Cosa que, com en altres models *greedy*, els permet no emmagatzemar els exemples individuals.

El representant clàssic d'aquests models és el *Naïve Bayes*, un algoritme que, tant ell com les seves variants, permet un entrenament incremental sense dependència de l'ordre de la seqüència d'entrenament.

6.5.1 NAÏVE BAYES

El *Naïve Bayes* és el classificador estadístic més senzill de tots, però malgrat això els seus resultats acostumen a ser comparables amb la resta. Va ser presentat per **[Duda & Hart, 1973]** i, a partir del teorema de Bayes **[Bayes, 1763]**, indueix un model probabilístic que determina la classe més probable a en funció dels valors dels seus atributs, que poden ser tan simbòlics com binaris o numèrics.

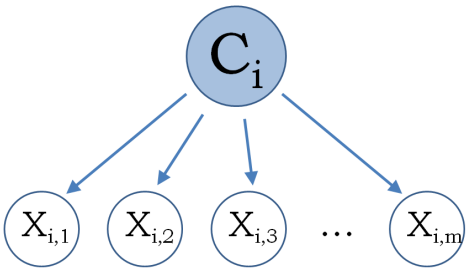

FIG. 38: Node Ci pels atributs Xi

Aquest model està format per un node que representa la classe *C* i un node per cada atribut x_i que representa l'instancia. Per simplificar la matemàtica associada s'introdueix una assumpció d'independència entre els atributs, i es considera que els valors dels atributs es generen a partir de la classe *C* segons les distribucions individuals $P(x_i|C)$. L'entrenament consisteix en obtenir una estimació d'aquestes distribucions; i la predicció de la classe es fa triant aquella que maximitza la probabilitat d'haver generat aquest exemple.

Més detalladament, la classificació d'un exemple es realitza seleccionant la classe que maximitza la seva probabilitat condicionada al conjunt d'atributs que descriuen l'instancia. Sigui el conjunt de classes $\{1 \dots k\}$ i l'exemple $\boldsymbol{x_i}$ amb els valors d'atributs $\{x_{i,1}, \dots, x_{i,m}\}$. Aleshores, la classe seleccionada és aquella que maximitza $P(k|\, x_{i,1}, \dots, x_{i,m})$:

$$arg\ max_k\ P(k|\, x_{i,1}, \dots, x_{i,m}) \approx arg\ max_k\ P(k) \cdot \prod_j P(x_{i,j}|k)$$

Mitjançant l'assumpció d'independència s'aproxima la probabilitat condicionada conjunta com el productori de les probabilitats condicionades individuals, és a dir, $P(k)$ i $P(x_{i,j}|k)$probabilitats individuals estimades a partir de les freqüències relatives induïdes a partir dels exemples d'aprenentatge. En els valors simbòlics, les probabilitats de la classe condicionades als valors d'un atribut és una simple freqüència relativa però *Naïve Bayes* també és aplicable a valors reals, simplement cal estimar les probabilitats a partir de la seva distribució, per exemple, una distribució normal. La possible incrementalitat d'aquest algorisme es deriva del fet que és possible estimar $P(x_i|C)$ incrementalment, a partir del simple compteig de co-ocurrències o, en el cas de valors reals, del càlcul incremental de la mitjana i la desviació estàndard.

Sempre que s'indueixen probabilitats a partir de les freqüències relatives apareix el problema de la dispersió de dades, cosa que fa el model molt sensible als esdeveniments que apareixen poc, o no apareixen, en el corpus d'entrenament. Aquest problema es pot minimitzar amb diferents tècniques de suavitzat, una de les mes senzilles es descriu a **[Ng, 1997a]**:

$$P(x_{i,j}|k) = \frac{P(k)}{N}, \qquad si\ |x_{i,j} \wedge k| = 0$$

És a dir, si un determinat valor no apareix en els exemples d'entrenament d'una classe determinada, s'estima que la seva probabilitat condicionada és la probabilitat de la classe dividida pel nombre d'exemples.

Malgrat la seva aparent simplicitat acostuma a donar resultats força satisfactoris. Aquest model s'ha utilitzat àmpliament en diferents aplicacions del AA i en gran quantitat de tasques de PLN (correcció ortogràfica, etiquetat morfosintàctic, desambiguació de dependències de sintagmes preposicionals, resolució d'ambigüitat semàntica i classificació de documents). A més, aquest model té l'avantatge que facilita la combinació estadística de diferents fonts independents.

Finalment, existeixen diferents variants que intenten millorar el model, com les xarxes bayesianes[9] **[Heckerman et al., 1996]**, un model basat en grafs que codifica les relacions entre variables i pot modelar situacions de dependència, o el *Naïve Bayes* multinomial[10] **[McCallum & Nigam, 1998]**, que permet tractar variables numèriques.

6.6 CLASSIFICADORS LINEALS

Aquesta família d'algorismes va néixer dins el camp de la Teoria d l'Aprenentatge Computacional (*CoLT, Computational Learning Theory*), dedicat a explorar les limitacions teòriques i metodològiques dels processos d'aprenentatge.

[9] *Bayesian Networks*
[10] *Multinomial Naïve Bayes*

Els classificadors lineals **[Nilsson et al., 1965]** són una família de funcions que bàsicament realitzen una combinació lineal dels atributs d'entrada, és a dir, una suma ponderada, el resultat es compara amb un llindar preestablert de manera que s'obtingui una sortida binaria que s'interpreta com una classificació.

L'algorisme d'aprenentatge consisteix a calcular la sortida per cada un dels exemples, i fer correccions en els pesos de manera que es minimitzin els errors. Aquests algorismes han demostrat donar molt bons resultats, especialment en problemes d'alta dimensionalitat, que continguin soroll i presentin una gran dispersió, és a dir, en problemes com els que apareixen en PLN.

El classificador lineal més utilitzat és el Winnow, que com la resta pot separar perfectament qualsevol conjunt d'exemples sempre que existeixi un hiperplà que separi les dues classes. Aquest algorisme, a més, pot modificar-se per continuar classificant raonablement be quan l'hiperplà separador varia lleugerament al llarg de l'entrenament, i per tant pot adaptar-se a derives en les dades d'entrenament.

6.6.1 WINNOW

El Winnow **[Littlestone 1988]** és un algorisme que construeix un classificador binari basat en un separador lineal. El model d'aquest separador consisteix en un hiperplà a l'espai d'atributs que maximitza la separació entre el conjunt d'exemples positius i exemples negatius. Aquest algorisme requereix que les representacions dels exemples estiguin formats per trets binaris, per tant cal binaritzar els atributs nominals i els numèrics.

Aquest algorisme pertany a la família de classificadors lineals coneguts com a *mistake-driven*[11], models que només actualitzen els models quan la classificació feta és incorrecte. L'algorisme general d'aquest principi és el que es mostra a la **Fig. 39**:

1. Initialize $i = 0$, success counter $c_i = 0$, model w_0
2. For $t = 1, 2, ..., T$:
   (a) Receive new example x_t
   (b) Predict $\widehat{y}_t = f(w_i, x_t)$, and receive true class y_t
   (c) If prediction was mistaken:
      i. Update model $w_i \rightarrow w_{i+1}$
      ii. $i = i + 1$
   (d) Else: $c_i = c_i + 1$

FIG. 39: Pseudo-codi dels classificadors *mistake-driven*. Font [Carvalho & Cohen, 2006].

En un classificador Winnow, a partir d'un model entrenat definit pel vector de pesos $\{w_1, \ldots, w_n\}$ que pondera la importància de cada atribut, la classificació per la classe positiva es realitza a partir de la següent funció:

[11] *Guiats pels errors*

$$\sum \omega_f > \theta, \qquad \forall f \in F$$

On ω_f és el pes d'aquells atributs actius, és a dir, d'aquells trets de l'instancia amb un valor binari 1, i θ és un llindar definit prèviament. Per tant, s'assigna la classe positiva si la suma dels pesos associats als atributs actius és superior al llindar θ; en cas contrari s'assigna la classe negativa.

L'algorisme original de Winnow ja és intrínsecament incremental, ja que durant l'entrenament va processant els exemples de forma individual. Per cada un el classifica com a positiu o negatiu, si la classe assignada coincideix amb la proporcionada no modifica el model, si la classificació és errònia modifica els pesos en la direcció que redueix l'error. Les actualitzacions dels pesos es fan mitjançant una paràmetres d'increment i decrement anomenats $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$ respectivament. A continuació es mostra l'algorisme extret de **[Blum, 1996]**:

```
a) W s'inicialitza amb 1s
b) Per cada exemple d'entrenament <x,y>
      1. S'obté una predicció de classificació y' = f(x)
      2. Si és correcta y=y', no es fa res
      3. Si y' és negativa i y era positiva
            1. Els pesos dels atributs actius es multipliquen
               per α (promotion >1), per incrementar el seu
               pes i facilitar la superació del llindar
      4. Si y' es positiva i y era negativa
            1. Els pesos dels atributs actius es multipliquen
               per β (demotion <1), per decrementar el seu
               pes i allunyar-lo del llindar
```

Aquest procés continua per cada exemple conegut i poc a poc el vector $\boldsymbol{w}$ de pesos, que inicialment s'havien inicialitzat amb 1s, convergeix cap a un valor que minimitza els errors de classificació. Una de les virtuts d'aquest algorisme és que convergeix molt ràpidament, especialment en problemes d'alta dimensionalitat on la majoria d'atributs són no rellevants.

6.6.2 VARIANTS WINNOW

L'algorisme Winnow original funciona raonablement be, però en la mateixa línia de classificadors *mistake-driven*[12], s'han proposat algunes variants que aconsegueixen millores en els resultats.

A **[Carvalho & Cohen, 2006]** aconsegueixen millores aplicant a les dades un pre-procés format per una *augmentació* i una *normalització*. L'*augmentació* consisteix a afegir al vector x_t, format per m trets que representen l'instancia, un tret addicional (m+1) conegut com a biaix amb un valor d'1. Una vegada *augmentat*, el vector es *normalitza* de manera que tos els pesos

[12] *Guiats pels errors*

es trobin a l'interval [0,1]. Durant la classificació, la *normalització* es realitza posteriorment a l'eliminació de tots els trets de l'instancia que no estan presents al model del classificador.

A **[Littlestone, 1988; Dagan et al., 1997]** es proposa una altre variant coneguda com a *Balanced Winnow*, format a partir de dos models classificadors Winnow, un positiu i un negatiu. A partir d'aquests dos models es defineix una funció classificadora diferencial:

$$f = sign\ (\langle x_t, u_i \rangle - \langle x_t, v_i \rangle - \theta_{th})$$

La idea és entrenar dos combinadors lineals, un per separar els exemples positius i l'altre els exemples negatius, i utilitzar la diferència de les dues sortides com un criteri conjunt per classificar els nous exemples. Tot i la seva simplicitat conceptual ha donat resultats molt positius en diferents tasques de PLN.

Una altre variant semblant a l'anterior, és la *Modified Balanced Winnow* **[Carvalho & Cohen, 2006]**, equivalent a la *Balanced Winnow* excepte per la introducció de dos petites modificacions. Per un costat s'amplia el criteri de predicció incorrecta, de manera que l'actualització dels pesos es realitza sempre que la separació entre les funcions dels dos models és inferior a un cert marge *M*, concretament:

$$y_t \cdot (\langle x_t, u_i \rangle - \langle x_t, v_i \rangle - \theta_{th}) \leq M$$

La segona diferència consisteix a modificar la regla d'actualització de pesos de manera que la correcció de cada pes sigui proporcional a l'atribut de l'exemple corresponent. A la **Fig. 40** es pot veure el pseudo-codi d'aquest algorisme:

1. Initialize $i = 0$, counter $c_i = 0$, and models u_0 and v_0
2. For $t = 1, 2, ..., T$:
   (a) Receive new example x_t, and add "bias" feature.
   (b) Normalize x_t to 1.
   (c) Calculate $score = \langle x_t, u_i \rangle - \langle x_t, v_i \rangle - \theta_{th}$.
   (d) Receive true class y_t.
   (e) If prediction was mistaken, i.e., $(score \cdot y_t) \leq M$:
      i. Update models. For all feature j s.t. $x_t > 0$:

$$u_{i+1}^j = \begin{cases} u_i^j \cdot \alpha \cdot (1 + x_t^j) & \text{, if } y_t > 0 \\ u_i^j \cdot \beta \cdot (1 - x_t^j) & \text{, if } y_t < 0 \end{cases}$$

$$v_{i+1}^j = \begin{cases} v_i^j \cdot \beta \cdot (1 - x_t^j) & \text{, if } y_t > 0 \\ v_i^j \cdot \alpha \cdot (1 + x_t^j) & \text{, if } y_t < 0 \end{cases}$$

      ii. $i = i + 1$
   (f) Else: $c_i = c_i + 1$

FIG. 40: Pseudo-codi del *Modified Balanced Winnow*.
Font [Carvalho & Cohen, 2006].

Diferents avaluacions comparatives **[Carvalho & Cohen, 2006]** conclouen que, en tasques de PLN, el MBW supera clarament la resta de variants Winnow, i fins i tot de les SVM

[6.6.4 Màquines de Vectors de Suport]. El que coincideix amb la intuïció general de que els algorismes d'actualització multiplicativa es comporten correctament amb problemes d'alta dimensionalitat i baixa densitat. Tot i així, existeixen altres variants menys conegudes basades en la actualització de pesos additiva.

Una d'elles és la que es coneix com a ROMMA (*Relaxed Online Maximum Margin Algorithm*) **[Li & Long, 2002]** que també aprèn incrementalment a separar funcions lineals amb un marge màxim, similar al *Large-Margin Winnow* **[Zhang, 2000]**. L'altre és la més coneguda *Passive-Aggressive* **[Crammer et al., 2003]** que planteja l'actualització dels pesos com un problema d'optimització.

Finalment, a **[Dredze et al., 2008]** es proposa una interessant variant coneguda com a *Confidence-Weighted Linear Classification* (CWLC) que corregeix un dels problemes dels classificadors lineals. Els classificadors lineals només actualitzen el pesos dels trets quan aquests trets apareixen en els exemples, per tant els trets que apareixen amb més freqüència s'actualitzen més sovint, i per tant convergeixen més ràpidament, que els trets menys freqüents. Però aquest fet no es tingut en compte pels classificadors anteriors, en canvi el CWLC incorpora aquesta informació ponderant els trets segons la seva freqüència d'aparició i per tant, segons la fiabilitat de l'estimació del seu pes.

6.6.3 SNOW

L'SNoW (*Sparse Network of Winnows*), desenvolupat per Dan Roth **[Roth & Zelenko, 1998]**, és una arquitectura basada en la combinació de classificadors Winnow. Està formada per una xarxa de baixa connectivitat, de separadors lineals en l'espai d'atributs, on a cada node està constituït per un algorisme Winnow. Aquesta arquitectura permet un tipus d'aprenentatge robust i eficient, manté l'adaptativitat i incrementalitat del Winnow, però permet la classificació multiclasse.

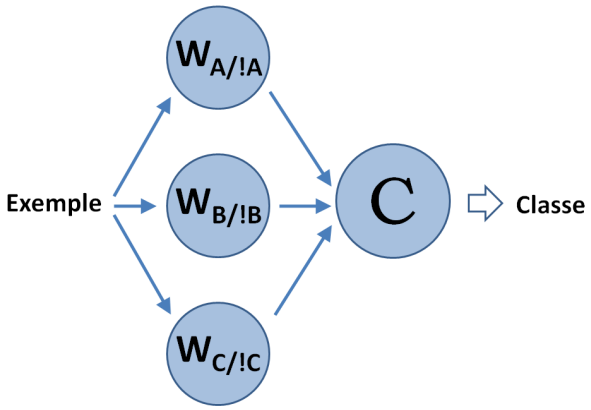

FIG. 41: Esquema de l'arquitectura SNoW: bateria de Winnows, un per classe.

Per construir una arquitectura SNoW, en primer lloc, es crea un classificador Winnow per cada una de les classes existents. Cada un d'ells s'entrena per discriminar entre la seva classe i la resta; de tal manera que els exemples d'una classe específica s'utilitzen com a exemples positius pel seu node i com exemples negatius per tots els altres. A l'hora de classificar un

nou exemple, cada node Winnow multiplica el seu vector de pesos pels atributs de l'exemple, i es tria la classe guanyadora com la corresponent al node amb més activació. Es pot trobar una descripció detallada a **[Golding et al., 1999]**.

El fet que cada node sigui un classificador independent amb una sortida quantitativa, permet normalitzar aquests valors respecte la suma de totes les sortides; cosa que permet obtenir graus de certesa o simplement oferir una classificació ordenada **[Carlson et al., 2001]**.

Cal destacar que cada node Winnow no accedeix a la totalitat dels trets disponibles en els exemples, sinó que només accedeix als atributs actius en els exemples associats a la seva classe. Aquest és el motiu pel que apareix la paraula *sparse* en el seu acrònim. Aquesta diferència, en relació a les xarxes connexionistes fortament interconnectades, fa que sigui una arquitectura força més eficient en tasques d'alta dimensionalitat com les associades al llenguatge.

Aquesta arquitectura s'ha aplicat a moltes tasques de PLN: correcció ortogràfica contextual, etiquetat morfosintàctic, desambiguació de dependències de sintagmes preposicionals, anàlisi sintàctic superficial, classificació de documents i la desambiguació semàntica de paraules.

6.6.4 MÀQUINES DE VECTORS DE SUPORT

Les Màquines de Vectors de Suport[13] (MVS) van ser proposades per Vapnik al 1979, però probablement per motius tecnològics, no es van popularitzar i utilitzar en aplicacions reals fins molt més tard **[Vapnik, 1995; 1998]**.

Aquests classificadors lineals es basen en el principi de Minimització de Risc Estructural[14] (MRE), i pertanyen a la família dels *classificadors basats en el marge*[15]. El principi general és similar al dels Winnow, construir un hiperplà que separi el conjunt d'exemples positius i negatius, però els MVS ho fan maximitzant el *marge*, el que millora la seva generalització. S'anomena *marge* a la distància entre l'hiperplà i els exemples positius i negatius **[Smola et al., 2000]**. La **Fig. 42** mostra un diagrama bidimensional d'un hiperplà separador amb marge màxim.

[13] *Support Vector Machines (SVM)*
[14] *Structural Risk Minimization (SRM)*
[15] *Margin-based Classifiers*

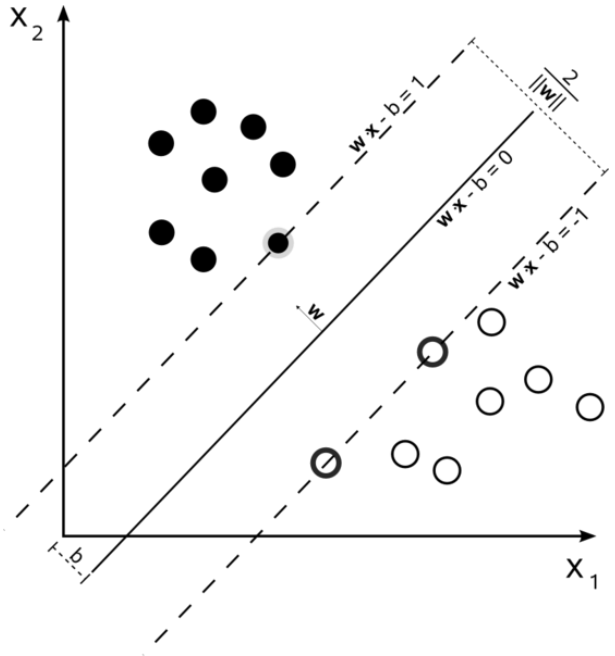

FIG. 42: Hiperplà separador amb marge màxim en una SVM

Tot i que la versió bàsica de les MVS genera classificadors lineals, pot ser ampliat mitjançant les funcions *kernels*. Aquestes funcions permeten projectar l'espai d'atributs d'entrada a un espai de dimensionalitat molt més gran, on poder separar linealment problemes que no ho eren en l'espai original. I aquesta és la principal avantatge d'aquests classificadors que actualment són els classificadors que, tot i el seu cost computacional, estan aconseguint els millors resultats. Es pot trobar més informació a **[Cristianini & Shawe-Taylor, 2000]**. Finalment, s'han desenvolupat diverses implementacions incrementals d'aquests classificadors **[Cauwenberghs & Poggio, 2000]**, i tot indica que els seus resultats són equivalents a la versió *batch*.

6.7 SISTEMES CONNEXIONISTES

Els sistemes connexionistes es caracteritzen per estar constituïts per una gran quantitat d'elements computacionalment simples, i interconnectats entre ells amb un elevat grau de interdependència. L'exemple més representatiu són les xarxes neuronals, la **Fig. 43** mostra un esquema d'un sistema connexionista prototípic.

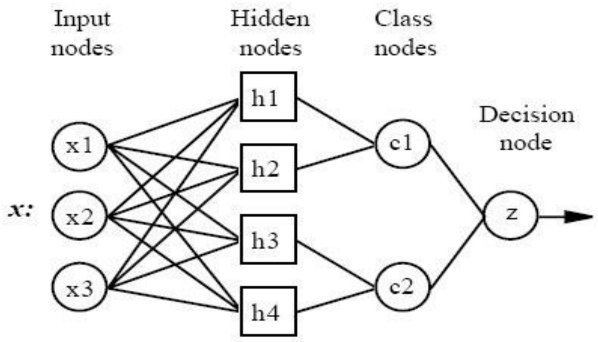

FIG. 43: Esquema d'una Xarxa Neuronal típica.

Les xarxes neuronals estan formades per un conjunt d'unitats simples, anomenades *neurones*, interconnectades entre si segons un conjunt de paràmetres anomenats *pesos*. La sortida de cada neurona, conegut com a nivell d'activació, és funció de les entrades i dels pesos corresponents. Normalment la primera capa de neurones rep la informació corresponent a la descripció de l'exemple que s'ha de classificar. Les seves sortides s'utilitzen com a entrades de les següents neurones i els nivells d'activació es van propagant a les següents neurones de la xarxa. I les neurones finals calculen la funció de classificació.

El procés d'entrenament consisteix a trobar la combinació de pesos que minimitzi l'error de classificació promitjat al llarg de la totalitat dels exemples d'entrenament. Aquests pesos són els que representen implícitament el coneixement adquirit. En altres paraules, donada una topologia determinada *a priori*, aquests models generalitzen el coneixement implícit dels exemples en forma de pesos i, com tots els algorismes *eager*, obliden els exemples individuals.

6.7.1 PERCEPTRÓ

El perceptró va ser proposat per **[Rosenblatt, 1958; 1962]** i representa el model connexionista més simple; de fet és tan simple que no arriba a ser ni una xarxa, és un simple node format per una sola neurona. Comparteix moltes similituds amb el classificador lineal Winnow, tot i que la incorporació d'una funció d'activació sigmoide fa que no sigui estrictament un classificador lineal. En qualsevol cas, la seva aproximació és molt semblant, i també pot separar qualsevol conjunt d'exemples que siguin separables per un hiperplà.

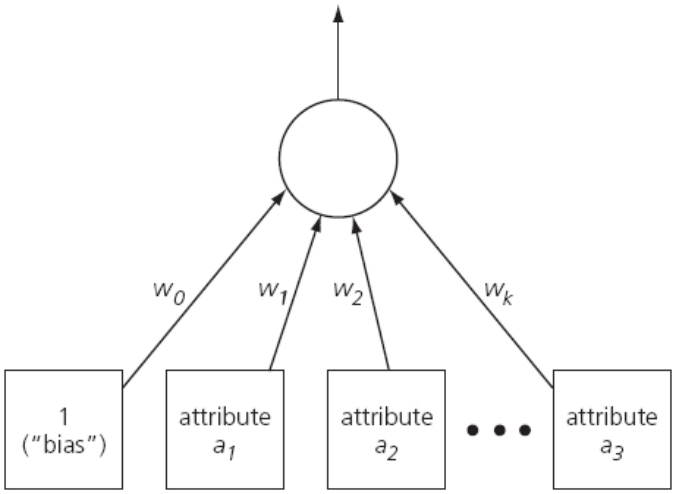

FIG. 44: Esquema d'un perceptró simple. Font [Witten & Frank, 2005]

Està format per una única *neurona* que rep la descripció de l'exemple a classificar. El valor a la sortida de cada neurona ve donada per la següent fórmula:

$$Output = w_0 + w_1a_1 + w_2a_2 + \ ... + \ w_3a_3 = w_0 + \sum_{i=1}^{N} w_ia_i$$

On a_i és el valor d'entrada *i-esim*, w_i és el pes associat a la connexió *i-esima* del perceptró, i w_0 és un valor de desplaçament determinat durant l'entrenament com al resta dels paràmetres dels pesos. A l'hora de calcular l'error, imprescindible per actualitzar els pesos durant l'entrenament, s'utilitza la diferència quadràtica entre el valor de sortida obtingut i l'esperat, una altre similitud entre l'algorisme del perceptró i els classificadors lineal.

L'entrenament, o ajustament de pesos, és possible fer-ho tant en mode *batch* com en mode incremental. En aquest segon cas, l'actualització es realitza després de tractar cada exemple, i l'error, utilitzat com a *feedback* corrector, és la diferència absoluta entre la sortida obtinguda i l'esperada, no la quadràtica.

```
Set all weights to zero
Until all instances in the training data are classified correctly
  For each instance I in the training data
    If I is classified incorrectly by the perceptron
      If I belongs to the first class add it to the weight vector
      else subtract it from the weight vector
```

FIG. 45: Pseudo-codi de l'algorisme d'entrenament d'un perceptró. Font [Witten & Frank, 2005]

6.7.2 XARXA NEURONAL MULTICAPA

Per obtenir un veritable sistema connexionista és necessari interconnectar un conjunt de perceptrons. L'arquitectura més habitual és la que es coneix com a multi-capa, o xarxa *feed-*

forward, on els senyals d'activació es propaguen des de les primeres capes, que tenen accés directe a la descripció de l'exemple, passant per les capes intermèdies que permeten aprendre *conceptes* pont, fins a les capes de sortida que generen la funció final del sistema. Aquesta arquitectura, amb capes ocultes, permet al sistema modelar funcions no-lineals, i per tant superar la limitació dels perceptrons que només poden classificar conjunts linealment separables. A la **Fig. 46** es mostra l'esquema d'una xarxa multi-capa.

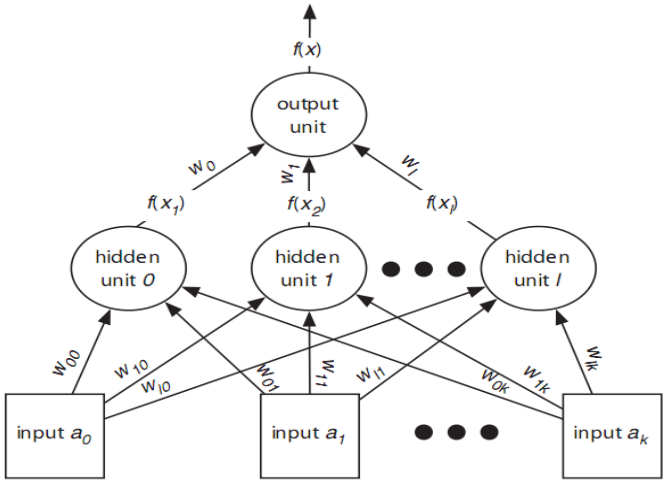

FIG. 46: Diagrama d'un perceptró multicapa amb una capa oculta. Font [Witten & Frank, 2005]

On a_i és el valor d'entrada *i*, i w_{ji} és el pes associat a la connexió entre l'entrada *i* i la neurona de sortida *j*. Hi ha un llindar diferent per cada node de sortida, que s'actualitza durant la fase d'ajustament de pesos.

Aquests sistemes connexionistes s'entrenen mitjançant el reajustament dels pesos durant la presentació iterativa dels exemples d'entrenament. La sortida de la xarxa es comparada amb la sortida desitjada i els pesos són ajustats en la direcció que tendeix a corregir la sortida. Aquest procés es repeteix un numero de vegades arbitrari, definit normalment a partir d'un error objectiu o del seu nivell de reducció.

L'entrenament de les xarxes multicapa es realitza mitjançant l'algorisme conegut com a *back-propagation*. L'aportació d'aquest algorisme és la formalització d'un mètode que permet calcular la proporció de l'error final que ha estat causat per cada una de les neurones de les capes intermèdies, de manera que també es puguin ajustar els seus pesos en la direcció corresponent. Amb l'ajuda d'aquest algorisme, a partir d'una assignació aleatòria de pesos i mitjançant un procediment iteratiu de descens del gradient, es van introduint correccions en els pesos fins que el sistema convergeix en un mínim local. Aquest és precisament un dels problemes tradicionals d'aquests sistemes, al no garantir errors mínims globals, és habitual haver de repetir els entrenament diverses vegades fins a trobar una xarxa que minimitzi l'error.

6.8 MODELS SEQÜENCIALS

Abans d'acabar aquest capítol es vol fer referència a uns models, que tot i no ser estrictament classificadors, poden ser imprescindibles en les tasques de PLN, es tracta dels models seqüencials. Els models seqüencials permeten estimar la probabilitat de que s'hagi produït una determinada seqüència de símbols, i són models imprescindibles en les tasques de desambiguació contextual.

Com ja s'ha explicat, la majoria de problemes de PLN poden plantejar-se com a tasques de classificació, però donada la fal·libilitat dels classificadors automàtics, sovint ens trobem amb elements ambigus i formalment similars a dues o mes classes. Afortunadament la naturalesa del llenguatge permet que el context local pugui resoldre bona part d'aquests ambigüitats, i aquesta és precisament la utilitat dels models seqüencials.

La importància dels següents dos models, els *N-grames* i els *Models Ocults de Markov*, rau en que és possible actualitzar-los incrementalment. Això permet complementar els classificadors incrementals amb un postprocés de desambiguació contextual que també sigui entrenable incrementalment. Una combinació que pot millorar considerablement els resultats del sistema en qualsevol tasca de PLN.

6.8.1 MODELS DE N-GRAMES

Els models d'*n-grames* són els models seqüencials més simples que permeten predir el següent element d'una seqüència o, més habitualment, la probabilitat d'una determinada seqüència amb l'objectiu de desambiguar una classificació prèvia.

Un *n-grama* és un fragment de *n* elements d'una determinada seqüència de símbols. Quan la mida de n és 1, s'anomenen *unigrames*, si és 2, *bigrames*, i si és 3, *trigrames* **[Fig. 47]**.

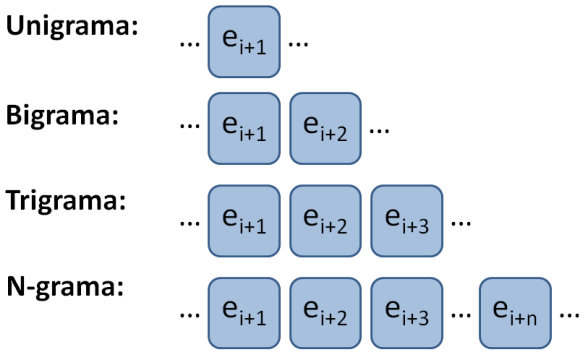

FIG. 47: Diagrama d'un N-Grama

La construcció d'aquests models és trivial, només cal comptabilitzar les ocurrències de cada n-grama. A l'hora d'obtenir la probabilitat d'una seqüència determinada només cal transformar les freqüències absolutes en relatives mitjançant una normalització. La incrementalitat de la construcció d'aquests models és intrínseca, sempre i quan la

relativització de les freqüències es faci en temps d'execució per no perdre la informació de les freqüències absolutes.

6.8.2 MODELS OCULTS DE MARKOV

Els Models Ocults de Markov[16] (MOM) són models estadístics més avançats especialment desenvolupats per aprendre i predir seqüències d'elements. Es poden interpretar com autòmats finits formats per estats, que representen les variables del model, i funcions de transició probabilística, de manera que les transicions entre dos estats tenen associades determinades probabilitats. Al mateix temps, cada un d'aquests estats generen símbols o etiquetes seguint també una distribució probabilística.

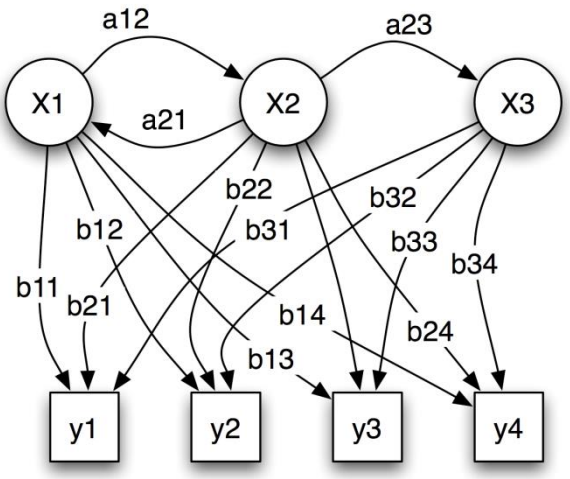

FIG. 48: Diagrama d'un HMM: estats X_i i observacions Y_i.

En cas d'ambigüitat, la predicció de la seqüència correcte es determina trobant aquella seqüència d'estats que maximitza la probabilitat d'haver obtingut una determinada seqüència de símbols. Tot i que sempre es pot fer una cerca exhaustiva, el problema pot resoldre's en un temps lineal gràcies a l'algorisme de Viterbi **[Viterbi, 1967]**.

Els Models Ocults de Markov s'han utilitzat principalment en tasques de baix nivell de PLN com l'etiquetat morfosintàctic i el reconeixement d'entitats nomenades, tot i que el seu èxit indiscutible l'han aconseguit en el camp del tractament de la parla, tant en síntesi com en reconeixement de la parla **[Màrquez, 2001]**.

La incrementalitat d'aquest algorisme neix de la possibilitat de computar incrementalment les probabilitats de transició i de generació, ja que són estimacions realitzades a partir del compteig simple de ocurrències.

[16] *Hidden Markov Models (HMM)*

7 COMBINACIÓ DE CLASSIFICADORS

7.1 INTRODUCCIÓ

Habitualment els classificadors incrementals vistos en el capítol anterior s'utilitzen directament com a classificadors aïllats, però en alguns casos pot ser aconsellable utilitzar arquitectures més elaborades basades en la combinació de classificadors. Afortunadament la majoria d'aquestes arquitectures són aplicables en entorns incrementals, i aquestes són les que veurem en aquest capítol.

La *combinació de classificadors*[1] és una tècnica que permet obtenir millors resultats en una tasca de classificació gràcies a la combinació d'un conjunt de classificadors simples o amb poca precisió. Des del punt de vista funcional una combinació de classificadors es comporta com un únic classificador complex que obté la seva sortida a partir de la combinació de les sortides dels classificadors simples.

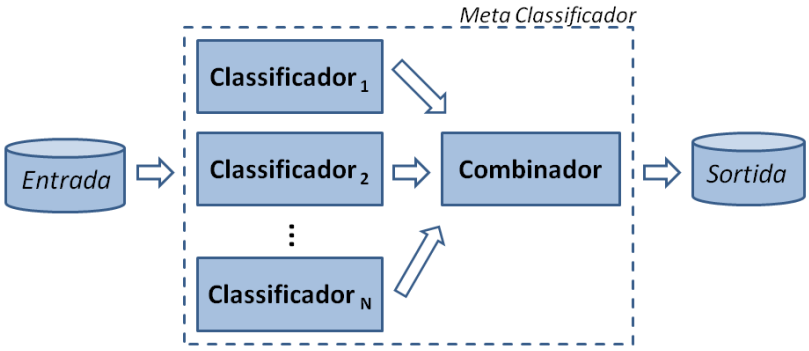

FIG. 49: Esquema general d'una combinació de classificadors.

Les combinacions de classificadors poden aconseguir importants millores en tasques de classificació, reduint significativament els errors, sempre i quan els classificadors simples utilitzats acompleixin dues condicions:

- **Precisió moderada:** els classificadors han de ser més precisos que l'atzar.
- **Diversitat elevada:** els classificadors han de ser el més variats possibles.

Per tant, cal aconseguir diferents classificadors, tots ells amb una precisió superior a l'atzar, que presentin un elevat grau d'independència en els seus errors, és a dir, que no s'equivoquin amb els mateixos exemples, ja que això és el que fa disminuir l'error de la seva combinació. Per exemple, si tenim 21 classificadors amb una precisió del 70% (individualment s'equivoquen un 30% de les vegades), perquè la seva combinació per

[1] *Ensemble of classifiers*

simple majoria s'equivoqui cal que s'equivoquin 11 o més dels classificadors. Suposant que els seus errors siguin totalment independents, la probabilitat que passi és p=2,8% **[Dietterich, 2000]**.

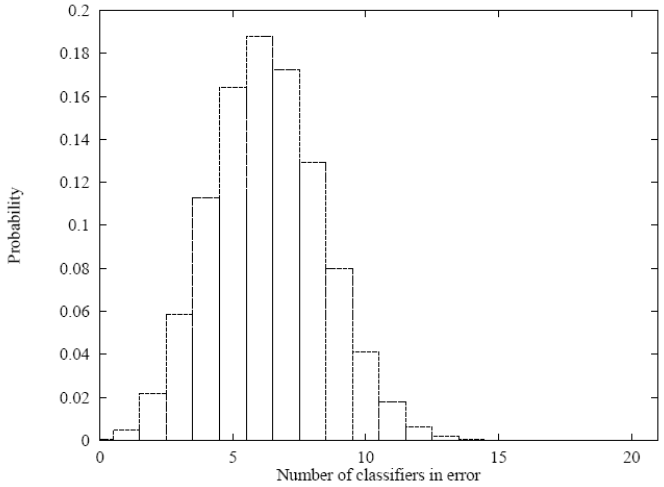

FIG. 50: Distribució de la probabilitat de que s'equivoquin N *experts*, si el seu error individual és del 30%. Font [Dietterich, 2000]

És a dir, el grau de diversitat del conjunt de classificadors està directament associat a la millora obtinguda per la seva combinació, ja que augmenta la independència estadística dels seus errors. Per això s'han provat diferents maneres d'obtenir aquesta diversitat **[7.2 Aconseguir Diversitat de Classificadors]**: utilitzant classificadors basats en diferents algorismes, representant els exemples amb diferents subconjunts dels trets existents o entrenant els classificadors amb exemples diferents del conjunt de dades.

Però abans d'entrar en aquest punt, s'explicaran els motius pel quals aquestes combinacions de classificadors obtenen millors resultats que els classificadors individuals i quines limitacions presenten que puguin desaconsellar la seva utilització en alguns casos específics.

7.1.1 EXPLICACIONS DE LA MILLORA OBTINGUDA

Com s'ha dit, la millora obtinguda al combinar diferents classificadors és un fet demostrat en multitud de treballs; potser no són tant evidents les seves causes, però s'han presentat diferents explicacions que ho recolzen:

> **Motius estadístics:** Perquè una combinació de classificadors s'equivoqui és necessari que diversos classificadors s'equivoquin simultàniament en un mateix exemple. Si els errors que cometen els classificadors són independents entre si, la probabilitat que succeeixi sempre es menor que la probabilitat de cada classificador individual; per tant l'error de la combinació sempre serà inferior a la dels classificadors que el composen.
>
> **Motius numèrics**: Normalment la quantitat de dades d'entrenament és petit en relació a la mida de l'espai d'hipòtesis, i per tant, donat un conjunt de dades

determinat existeixen diferents hipòtesis, o models predictors, compatibles amb les dades i, per tant, igualment correctes. Si els classificadors han estat sobre-entrenats i els seus models són massa fidels a les dades utilitzades, la seva combinació proporciona un major grau de generalització.

Motius representacionals: En alguns casos la representació de les hipòtesis utilitzada pel classificador triat no és capaç de representar la solució correcta. La combinació de diferents classificadors permet obtenir hipòtesis més complexes que cauen fora dels espais d'hipòtesis de cada classificador. Si, a més, els classificadors són prou diferents, especialment des del punt de vista representacional, la combinació de les seves hipòtesis proporciona un major grau d'expressivitat.

Però dels tres factors, el més important és l'estadístic; com ja s'ha dit, perquè una combinació de classificadors cometi un error, cal que la majoria dels seus *experts* s'equivoqui, cosa que sempre és més improbable que l'error d'un classificador individual. Per això és fonamental fer vàlida l'assumpció d'independència entre classificadors. Aquest grau d'independència es pot quantificar amb el que a **[Brill & Wu, 1998]** anomenen *complementarietat*, una mètrica que reflecteix la proporció de vegades que un classificador s'equivoca quan l'altre encerta. Partint d'aquesta mesura es pot determinar el límit superior de la precisió obtinguda mitjançant la combinació dels classificadors corresponents.

Finalment, per il·lustrar els efectes beneficiosos de la combinació de classificadors, fins i tot quan l'única diversitat és la introduïda per l'aleatorietat de les condicions inicials, es mostren els resultats d'un experiment de **[Dietterich, 2000]**. La **Fig. 51** mostra la comparació dels resultats d'un únic classificador C4.5 amb els d'una combinació de 200 classificadors. Cada un dels punts representa un conjunt de dades d'avaluació i la seves coordenades es corresponen als errors de generalització dels dos classificadors. Si el punt cau per sota de la diagonal, significa que per aquest conjunt de dades la combinació de classificadors presenta un error menor que el C4.5.

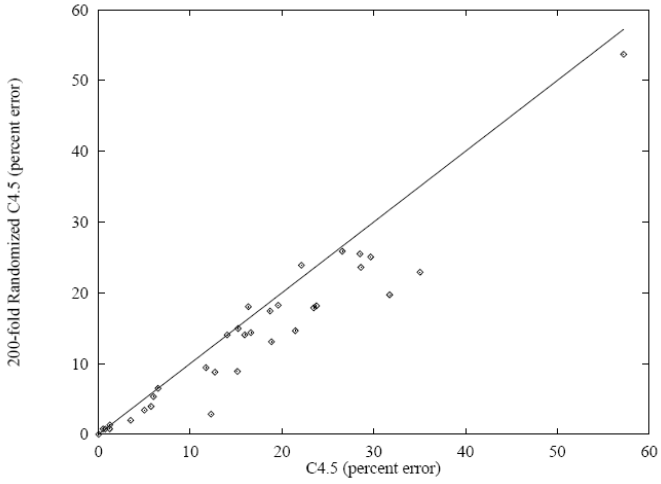

FIG. 51: Comparació d'un classificador C4.5 amb una combinació de 200 classificadors C4.5 amb soroll aleatori. Font [Dietterich, 2000]

7.1.2 LIMITACIONS DE LES COMBINACIONS DE CLASSIFICADORS

El punt anterior permet entendre que una de les limitacions de les combinacions dels classificadors és que no poden utilitzar-se quan els classificadors no compleixen la condició de *diversitat elevada*, és a dir, quan no presenten variacions importants. Cosa que només passa en aquells casos en els quals només es disposa d'un algorisme, les dades d'entrenament tenen pocs atributs o no es té accés a una gran quantitat d'exemples.

Però existeix una altra limitació associada a la condició de *precisió moderada*. Sembla raonable que la precisió individual dels classificadors hagi de ser superior a l'obtinguda per l'atzar, però també pot ser un problema si els classificadors són massa precisos. Si tots els classificadors presenten errors molt baixos, la seva combinació només oferirà una millora marginal, pràcticament nul·la. Però a més, si un dels classificadors és molt més bo que la resta les seves classificacions poden quedar amagades per la resta, fins al punt que la combinació assoleixi una precisió inferior al millor dels seus classificadors.

Aquest fenomen es pot posar de manifest, no només en problemes simples on els classificadors poden resoldre la tasca amb facilitat, sinó en tasques complexes que tinguin accés a corpus d'entrenament massius. El motiu és principalment que la utilització de grans corpus minimitza la variància en les dades d'entrenament i les diferències entre classificadors. A **[Banko & Brill, 2001a; 2001b]** es realitza un experiment en el qual, entre altres coses, es mesura la complementarietat de dos classificadors entrenats amb un corpus de mida creixent. A la **Taula 3** es mostra la disminució de la complementarietat a mesura que augmenta la mida del corpus d'entrenament:

Training Size (words)	Complementarity(L1,L2)
10^6	0.2612
10^7	0.2410
10^8	0.1759
10^9	0.1612

TAULA 3: Reducció de la complementarietat en funció de la mida del corpus d'entrenament. Font [Banko & Brill, 2001a].

És per això, que la utilització de grans corpus que redueixen la complementarietat entre classificadors i, raonablement, augmenten la seva precisió individual, suposa la reducció dels avantatges de la combinació de classificadors. Aquests efectes es poden veure molt clarament a la **Fig. 52**, que mostra els resultats d'un interessant experiment de **[Banko & Brill, 2001a]** on compara la precisió d'una combinació de 3 classificadors amb la precisió del millor d'ells, en una tasca de desambiguació de conjunts de confusió. Es pot observar com per corpus petits ($<10^6$) la combinació és clarament beneficiosa, per corpus mitjans ($<10^7$) els resultats són pràcticament equivalents, i per corpus grans ($<10^8$) la combinació arriba a ser perjudicial aconseguint una precisió inferior al millor dels classificadors.

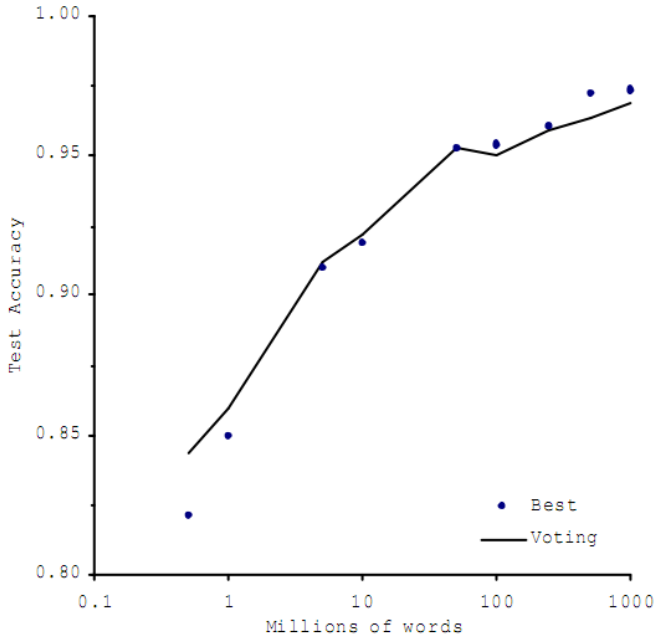

FIG. 52: Comparativa de la precisió conjunta d'un classificador per comitè respecte la del seu millor *expert*. Font [Banko & Brill, 2001a]

És important tenir aquest fenomen en compte a l'hora d'implementar sistemes incrementals basats en la combinació de classificadors. La possibilitat de processar grans corpus i de la millora continuada dels classificadors incrementals facilita que, arribats a un cert punt de la vida del sistema, algun dels algorismes individuals assoleixi una precisió significativament millor que la resta, i que per tant la combinació realitzada representi una pèrdua de precisió. Afortunadament existeix una solució que permet obtenir el millor resultat en cada una de les situacions **[7.4.5 Arbitratge]**.

7.2 ACONSEGUIR DIVERSITAT DE CLASSIFICADORS

Com s'ha dit anteriorment, la combinació de classificadors dona millors resultats quan els classificadors presenten un comportament diferenciat, concretament quan no coincideixen massa en els exemples que classifiquen incorrectament. La forma més directa d'aconseguir-ho és utilitzar diferents algorismes basats en diferents paradigmes, és a dir combinar classificadors *heterogenis*.

Però, fins i tot si es tracta de combinar classificadors basats en un únic algorisme, és possible obtenir classificadors de comportament *heterogeni* mitjançant la manipulació del procés d'entrenament al qual se sotmeten. L'objectiu és utilitzar el mateix algorisme per aconseguir diferents classificadors amb comportaments diversos. En aquest punt es presenten les diferents tècniques existents per aconseguir-ho **[Dietterich, 2000]**.

7.2.1 UTILITZACIÓ D'ALGORISMES NO-DETERMINISTES

La primera opció per obtenir diversitat de classificadors és aprofitar les propietats no-deterministes d'alguns algorismes d'inducció. Hi ha algorismes que utilitzen l'aleatorietat en

alguns punts del seu procés, com per exemple, en la inicialització d'alguns paràmetres del model, que és el cas més habitual.

També es poden crear variacions d'algorismes deterministes introduint soroll explícitament, per transformar-los en no-deterministes, per exemple, modificant l'obtenció d'arbres de decisió perquè seleccioni aleatòriament l'atribut de tall. En qualsevol d'aquests casos, la simple aplicació repetida del mateix algorisme sobre les mateixes dades ja permet obtenir classificadors amb comportaments diferenciats.

7.2.2 MANIPULACIÓ DELS CONJUNTS D'ENTRENAMENT

Una altra possibilitat, especialment si disposem d'una gran quantitat d'exemples, és utilitzar la tècnica de re-mostreig **[7.4.1 Bagging]** i **[7.4.3 Boosting]**. Es poden obtenir diferents classificadors simplement entrenant-los amb conjunts diferents d'exemples.

Aquesta tècnica és especialment útil en algorismes inestables, és a dir, en algorismes molt sensibles a les dades d'entrenament, com els arbres de decisió. En aquest cas la diversitat dels classificadors prové de la variabilitat introduïda en les dades d'entrenament.

7.2.3 MANIPULACIÓ DELS TRETS DELS EXEMPLES

Si no es vol dividir el conjunt d'entrenament, cosa que pot reduir la precisió dels classificadors individuals, i els exemples contenen una gran quantitat d'atributs, és possible entrenar els classificadors amb diferents representacions dels mateixos exemples.

Simplement cal utilitzar diferents subconjunts dels mateixos atributs, o diferents transformacions dels atributs, per entrenar diferents exemples. En aquest cas la diversitat dels classificadors prové de la variabilitat en els atributs descriptius dels exemples.

Aquesta tècnica dóna resultats especialment bons quan hi ha redundància en els trets d'entrada. De fet, es pot utilitzar una variant d'aquesta tècnica per determinar la importància relativa de cada atribut, o per trobar un subconjunt d'atributs òptim.

7.2.4 MANIPULACIÓ DE LES CLASSES DE SORTIDA

Finalment hi ha una quarta estratègia per obtenir comportaments diferenciats entre els classificadors, tot i ser entrenats amb les mateixes dades i atributs. Es tracta de crear un problema intermedi, amb unes noves classes, que permetin fàcilment la seva projecció cap a les classes originals.

Un d'aquests mètodes consisteix a re-agrupar, per cada classificador H_n, les K classes originals en dos grups A_n i B_n , i entrenar cada classificador per discriminar els exemples corresponents als seus grups. Durant la fase de classificació es realitza una votació entre tots els classificadors. Per cada exemple x_i, si la sortida del classificador j és $H_n(x_i)=A_n$, cada classe original inclosa a A_n rep un punt. Es fa el mateix per B_n, i la classe final és la que rep més vots del conjunt de classificadors.

7.3 SISTEMES DE VOTACIÓ

El nucli d'una combinació de classificadors és el *sistema de votació*. Com s'ha explicat anteriorment, la combinació de classificadors més habitual consisteix en una agrupació de classificadors, entrenats amb diferents subconjunts de dades o de trets, que fusiona les seves sortides en una classificació única. Aquesta funció que transforma el conjunt de classificacions independents en un valor homogeni, és el que genèricament es coneix com a *sistema de votació*. Com aquesta transformació es realitza durant la classificació a partir de les sortides dels classificadors, independentment de com hagin estat entrenats, tots els sistemes de classificació són aplicables directament en entorns incrementals.

Tot i que la majoria de les vegades s'utilitza un model de votació simple conegut com *plurality*, existeixen diferents alternatives, algunes més elaborades que altres, que segons el cas poden donar millors resultats **[Van Erp et al., 2002]**. Els diferents sistemes de votació s'agrupen segons el tipus de resposta que combinen, a l'hora de classificar un exemple hi ha classificadors que retornen una única etiqueta, una llista d'etiquetes ordenades per preferència, o una llista d'etiquetes ponderades segons el grau de pertinència.

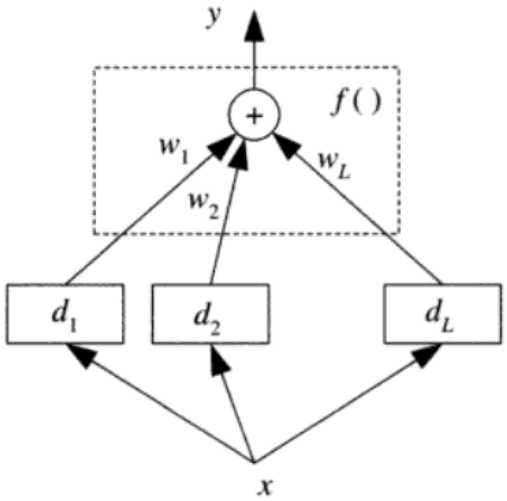

FIG. 53: Esquema d'un sistema genèric de votació. Font [Alpaydin, 2004].

7.3.1 VOTACIÓ SENSE PES

Els mètodes de *votació sense pes* són mètodes en el quals tots els vots emesos pels classificadors tenen el mateix pes, la unitat, i per tant les úniques diferències entre les classes candidates es manifesten en la quantitat de vots que rep cada una. El principal inconvenient que presenten és el risc d'un empat, que en absència de més informació, només es pot resoldre triant-ne un a l'atzar **[Van Erp et al., 2002]**.

Els mètodes de votació sense pes més simples, són el *plurality* i el *majority*. Tots dos són mètodes realitzables en un únic pas, és a dir, on cada classificador és utilitzat una sola vegada:

***Plurality Vote*:** Cada classificador emet un sol vot, la classe més votada és la guanyadora. És un sistema simple que funciona correctament. El principal inconvenient és la possibilitat de guanyar a partir d'un reduït nombre de vots per una minoria dubtosa; el risc augmenta amb el nombre de classes.

***Majority Vote*:** Cada classificador emet un sol vot, la classe que rep més de la meitat de tots els vots és la guanyadora. És un sistema simple amb una baixa taxa d'errors, la classificació final té una elevada precisió però una baixa cobertura, causada per aquells casos on no s'assoleix cap majoria.

Existeixen altres mètodes de votació sense pes que necessiten que la votació es faci en múltiples passes, són l'*amendment*, el *run off* i el *condorcet*. A més, aquests mètodes requereixen que els classificadors siguin capaços de mostrar preferència entre dues classes qualsevol, cosa que els fa més difícils d'aplicar. Per això podria considerar-se que aquests mètodes pertanyen a les votacions amb ranking, però cada una de les passes estan formades indiscutiblement per votacions sense pes.

***Amendment Vote*:** Es comença amb una votació per majoria entre les dues primeres classes, la classe guanyadora s'enfronta contra la següent classe candidata, i es repeteix el procés fins que hi ha una classe guanyadora. El problema d'aquest sistema és la falta de neutralitat, el resultat de la votació té un biaix que afavoreix al darrer candidat.

***Runoff Vote*:** És una votació en dues fases, en la primera ronda s'aplica una votació per pluralitat, les dues classes més votades passen a la segona fase on s'aplica una votació per majoria. Aquest sistema dóna millors resultats que la pluralitat, sempre hi ha classe guanyadora i és improbable que guanyi un candidat minoritari.

***Condorcet Count*:** Tots els candidats són comparats en eleccions aparellades, el guanyador de cada votació acumula un punt, la classe amb major puntuació és la guanyadora. És un sistema més complex, però pràcticament no pateix les limitacions de la resta de votacions.

7.3.2 VOTACIÓ AMB RANKING

Els mètodes de *votació amb ranking* s'utilitzen per combinar les votacions de classificadors que en comptes d'assignar una sola classe, retornen una llista de classes ordenades segons les seves preferències d'assignació **[Van Erp et al., 2002]**. En aquest escenari es disposa de més informació sobre els diagnòstics dels *experts* que en les votacions sense pes, i indirectament permet obtenir la preferència d'un classificador entre dues classes qualsevol.

Un avantatge d'aquests mètodes és que eliminen el perill d'un excés de confiança dels classificadors, evitant la sobreponderació de determinades classes que tot i poder ser preferencials no siguin dominants. La votació amb ranking és especialment útil a l'hora de

combinar graus de confiança que siguin difícils d'escalar, com les distàncies o les probabilitats.

> ***Borda Count:*** Requereix un ranking de preferència complet, de tots els classificadors a totes les classes. La classe guanyadora és la que obté un promig més alt de les posicions assignades en els rankings de cada un dels classificadors. És la variant ordenada de la *Sum Rule* **[7.3.3 Votació amb Índex de Confiança]**.
>
> ***Single Transferable Vote (STV)*:** Es realitza una votació per majoria entre les classes presents a la primera opció de cada classificador. Si alguna classe apareix com a primera opció a més de la meitat dels classificadors, guanya la votació; en cas contrari, s'elimina de la votació la classe que ha aparegut menys vegades com a primera opció. Es repeteix el procés fins que alguna de les classes guanya la votació. Amb aquest mètode, tenir una baixa posició per part d'un únic classificador, no és tant determinant com en el *Borda Count*.

7.3.3 VOTACIÓ AMB ÍNDEX DE CONFIANÇA

Els mètodes de *votació amb índex de confiança* permeten combinar votacions de classificadors que proporcionen graus de preferència per cada candidat. Per fer-ho assignen a cada classe un valor quantitatiu que reflecteix el grau de certesa o de pertinença d'un exemple determinat **[Van Erp et al., 2002]**.

El que és important és que els índexs de confiança dels diferents classificadors estiguin correctament escalats. Alguns graus de confiança típicament utilitzats són les distàncies o el seu invers, el grau de similitud, així com a estimacions de probabilitats. Especialment en aquest darrer cas, aquests sistemes de votacions són simples, robusts, i acostumen a donar molts bons resultats.

> ***Pandemonium*:** La classe que rep el vot amb el grau de confiança més elevat és el guanyador. És el sistema més senzill, però impedeix als classificadors expressar diferències de preferències entre candidats.
>
> ***Sum Rule*:** Cada classificador ha de donar un valor de confiança per cada una de les classes; per cada una se sumen els valors corresponents i guanya la classe amb la suma més alta.
>
> ***Product Rule*:** Cada classificador dóna un valor de confiança per cada una de les classes, per cada una es multipliquen els valors assignats i guanya la classe amb el producte més elevat. Té com a inconvenient que és molt sensible als índexs de confiança baixos.

7.4 COMBINACIONS COMPLEXES

En la secció anterior hem vist les combinacions de classificadors basats en la simple aplicació d'un sistema de votació sobre un conjunt de classificadors. Però existeixen altres arquitectures més complexes que combinen classificadors de manera jeràrquica o multi-nivell. Habitualment en aquestes configuracions les sortides d'un nivell de classificadors limiten les possibles classes, o seleccionen un dels classificadors, en un altre nivell; és a dir, existeixen interdependències entre els diferents classificadors.

En aquest punt es descriuen en primer lloc dues combinacions simples, però populars en el camp de l'AA, basades en la manipulació de les dades d'entrenament: el *bagging*, una tècnica de divisió d'un corpus basada en el re-mostreig amb repetició, i els *comitès solapament*[2]. Després es descriuen quatre arquitectures multinivell: el *boosting* i l'*stacking*, basades en el meta-aprenentatge, la combinació en *cascada*, que permet, i l'*arbitratge*, una arquitectura imprescindible per combinar classificadors incrementals.

7.4.1 BAGGING

El *bagging*, fusió de *bootstrap aggregation*, **[Breiman, 1996]** és el mètode més directe per aconseguir la diversitat d'un grup de classificadors quan es disposa d'un únic conjunt de dades. A partir d'aquest conjunt d'entrenament, format per N exemples, es generen diversos subconjunts seleccionant N exemples mitjançant mostreig amb reemplaçament. Cada un d'aquests subconjunts conté un promig del 63,2% del conjunt original amb diferents exemples repetits una o més vegades. Aquests subconjunts d'entrenament s'utilitzen per entrenar diversos classificadors i les seves classificacions són combinades, normalment, mitjançant un sistema de votació per pluralitat.

Tot i ser desenvolupat, i aplicar-se habitualment, amb algorismes *batch*, és possible obtenir una versió incremental equivalent que no requereix l'emmagatzemament dels exemples. Si es canvia el punt de vista de la partició, i es considera el nombre de vegades que un exemple és seleccionat per cada subconjunt s'arriba que segueix una distribució de Poisson. Per tant, pot emular-se l'assignació de cada exemple un determinat nombre de vegades a cada subconjunt, i utilitzar-lo per entrenar cada un dels classificadors amb un pes variable.

Això és el que es proposa a **[Oza & Russell, 2001]**, a mesura que cada exemple és presentat a la combinació de classificadors, es defineix un valor aleatori K aproximat com *Poisson(1)* que representa el nombre de vegades que aquest exemple apareixerà en el subconjunt del primer classificador, i s'entrena aquest model repetint la presentació K vegades. El procés es repeteix per la resta de classificadors fins que l'exemple ja ha estat completament utilitzat. A la **Fig. 54** es mostra el pseudo-codi de la versió incremental de l'algorisme de *bagging*; la funció de *Poisson(x)* que utilitza per obtenir números aleatoris amb aquesta distribució pot implementar-se mitjançant l'algorisme mostrat a la figura **Fig. 55**.

[2] *Cross-validated committees*

OnlineBagging($\mathbf{h}, L_o, d$)
For each base model $h_m \in \mathbf{h}, m \in \{1,2,\ldots,M\}$,
Set k according to $Poisson(1)$.
Do k times
$h_m = L_o(h_m, d)$.

FIG. 54: Pseudo-codi de l'algorisme de *bagging* incremental. Font [Oza& Russell, 2001].

```
algorithm poisson random number (Knuth):
    init:
         Let L ← e^-λ, k ← 0 and p ← 1.
    do:
         k ← k + 1.
         Generate uniform random number u in [0,1] and let p ← p × u.
    while p > L.
    return k − 1.
```

FIG. 55: Pseudo-codi de la funció *Poisson(x)*. Font [Knuth 1969].

7.4.2 COMITÈS AMB SOLAPAMENT

La combinació mitjançant *comitès amb solapament*, és semblant a l'anterior però utilitza un altre mètode a l'hora de dividir el conjunt d'entrenament. La idea és dividir-lo en diferents subconjunts disjunts i descartar-ne un aleatòriament per cada classificador. D'aquesta manera s'obtenen diferents subconjunts amb solapament que poden utilitzar-se per entrenar els diferents classificadors que formaran el comitè **[Parmanto et al., 1996]**.

L'obtenció d'una variant incremental d'aquesta combinació, que permeti emular l'existència dels diferents subconjunts d'entrenament sense necessitat d'emmagatzemar els exemples, és senzilla. Si tenim un conjunt de classificadors incrementals, amb cada exemple s'entrenen tots els classificadors excepte un, seleccionat aleatòriament cada vegada. El resultat és que tot i que els classificadors comparteixen una fracció important del corpus d'entrenament el solapament no és total i, per tant, obtenen models diferents.

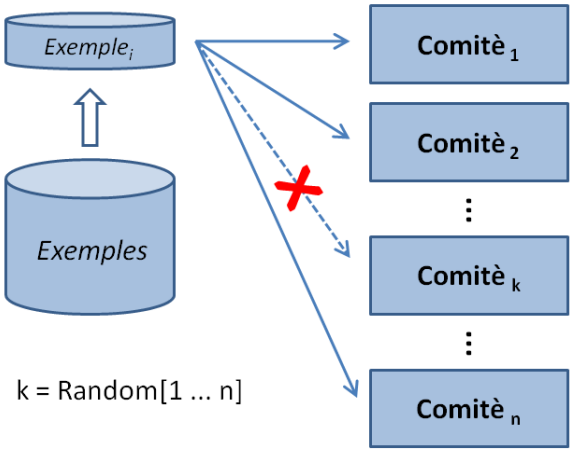

FIG. 56: Divisió incremental del corpus per a un Comitè amb solapament.

7.4.3 BOOSTING

Aquesta aproximació va sorgir dins del camp de la Teoria de l'Aprenentatge **[Freund & Schapire, 1997]** a l'intentar respondre la pregunta teòrica de si era possible obtenir classificadors arbitràriament precisos mitjançant la combinació intel·ligent d'un conjunt de classificadors simples i menys precisos. Una de les implementacions més utilitzada és al que es coneix com a AdaBoost[3] **[Freund & Schapire, 1996; Schapire & Singer, 1999]**.

Com en altres combinacions, la diversitat dels classificadors s'aconsegueix manipulant el conjunt d'entrenament, en aquest cas esbiaixant el mostreig cap als exemples més difícils de classificar. Els classificadors individuals s'entrenen seqüencialment de manera que cada un d'ells utilitza un conjunt d'exemples lleugerament diferent de l'anterior. A diferència del Bagging les composicions dels respectius conjunts d'entrenament no són aleatòries, sinó que es veuen influenciades pels errors comesos pel classificador precedent.

Per cada iteració k, s'aplica l'algorisme sobre el conjunt d'entrenament per obtenir el model M_k que minimitza el seu error ponderat. L'avaluació d'aquest model permet actualitzar la importància de cada exemple: augmentant el seu pes si ha estat classificat erròniament i disminuint-lo si ha estat classificat correctament. El procés es va repetint per obtenir nous classificadors que minimitzin els errors dels conjunts de dades ponderats, que progressivament incorporen aquells exemples que els models precedents no han classificat correctament. La classificació d'un nou exemple es realitza combinant les sortides de tots els classificadors utilitzats, normalment amb una majoria simple, és a dir, seleccionant la classe més votada pel conjunt de classificadors entrenats.

Aquesta tècnica, tal com es va plantejar originalment, és un procés iteratiu pel qual s'entrenen en mode *batch* una sèrie d'algorismes classificadors; i per tant, no és aplicable eficientment en mode incremental. Però potser, podria aplicar-se cada exemple al primer classificador i, segons si el classifica correctament o no, assignar-li una determinada probabilitat de ser presentat al següent classificador. Repetint aquest procés s'acabaria obtenint una sèrie de classificadors entrenats principalment amb aquells exemples que el classificador anterior no va poder classificar correctament. Això és el que també es proposa a **[Oza 2001]**, mitjançant l'algorisme incremental mostrat a la **Fig. 57**.

[3] *Adaptative Boosting*

Initial conditions: For all $m \in \{1,2,\ldots,M\}$,

$\lambda_m^{sc} = 0, \lambda_m^{sw} = 0.$

OnlineBoosting($\mathbf{h}, L_o, (x,y)$)

Set $\lambda = 1$.

For each base model $h_m \in \mathbf{h}, m \in \{1,2,\ldots,M\}$,

Set k according to $Poisson(\lambda)$.

Do k times

$h_m = L_o(h_m,(x,y))$.

If $y = h_m(x)$

$\lambda_m^{sc} \leftarrow \lambda_m^{sc} + \lambda$

$\varepsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}}$

$\lambda \leftarrow \lambda\left(\frac{1}{2(1-\varepsilon_m)}\right)$

else

$\lambda_m^{sw} \leftarrow \lambda_m^{sw} + \lambda$

$\varepsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}}$

$\lambda \leftarrow \lambda\left(\frac{1}{2\varepsilon_m}\right)$

end

To classify a new example with input x, return:

$h_{fin}(x) = \arg\max_{y \in Y} \sum_{m=1}^{M} \log\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right) I(h_m(x) = y).$

FIG. 57: Pseudo-codi de l'algorisme de *boosting* incremental.
Font [Oza & Russell, 2001].

7.4.4 CASCADA

La combinació de classificadors en *cascada* **[Kaynak & Alpaydin, 2000]** és una estratègia d'optimització de recursos que consisteix a ordenar els classificadors segons la seva complexitat computacional o el seu cost de representació, de manera que el classificador d_{j+1} sigui més costós que el d_j. Es tracta d'un sistema multinivell que només utilitza un classificador si cap dels precedents ha pogut realitzar una classificació fiable. Per fer-ho s'associa a cada classificador una mesura de confiança w_j que determina, en cas de superar un determinat llindar θ_j, si el classificador és fiable.

L'entrenament s'inicia presentant el conjunt d'entrenament al primer classificador. Aquells exemples que no puguin classificar-se amb suficient confiança passen a constituir el conjunt d'entrenament del següent classificador. A diferència del *boosting*, no se seleccionen només els exemples classificats incorrectament, sinó aquells que no són classificats amb prou seguretat. La idea és que el primer classificador, el més simple i que utilitza menys recursos, classifica la majoria d'exemples. I que el següent classificador computacionalment més

costós, només s'utilitza per aquelles excepcions o casos que són més difícils de classificar **[Alpaydin, 2004]**.

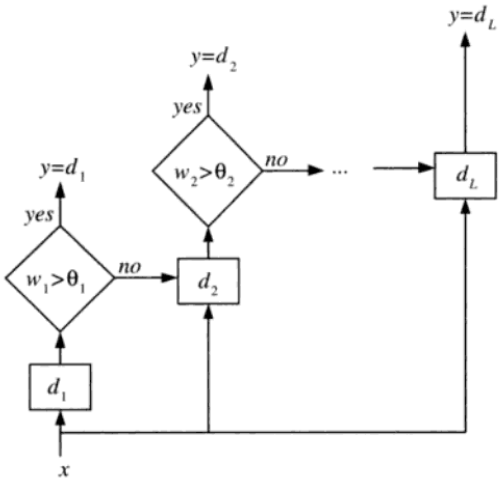

FIG. 58: Esquema d'una combinació de classificadors en cascada. Font [Alpaydin, 2004].

7.4.5 ARBITRATGE

L'*arbitratge* és un sistema molt simple per combinar classificadors però que és molt útil en determinades circumstàncies. En comptes d'intentar combinar els diferents resultats mitjançant un sistema de votació, el que fa és seleccionar el classificador que ha d'utilitzar-se en cada moment.

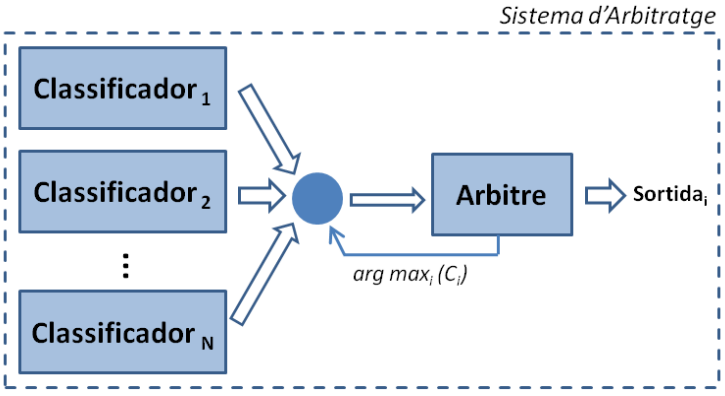

FIG. 59: Esquema d'un sistema combinador per arbitratge.

Una possible estratègia és calcular la precisió i cobertura de cada classificador base i seleccionar en cada moment el millor classificador. En un sistema *batch*, on la precisió dels classificadors ja entrenats és constant, aquest model no té sentit; simplement cal triar el model que després de l'entrenament ha obtingut millors resultats i eliminar la resta.

Però en un sistema incremental, amb classificadors dinàmics que milloren contínuament i a diferents ritmes, la utilització d'un sistema d'arbitratge és imprescindible. Recordem que hi ha classificadors senzills que obtenen bons models amb molts pocs exemples, però per contra de seguida assoleixen un límit en la seva precisió. També hi ha altres classificadors

que aprenen molt més lentament, però que amb el temps poden arribar a obtenir precisions molt més elevades. En aquests casos és interessant utilitzar classificadors amb diferents ritmes d'aprenentatge i deixar que el sistema d'arbitratge, que monitoritza la precisió de cada un, seleccioni el millor model en cada etapa de l'evolució del sistema. La **Fig. 60**, mostra l'evolució de la precisió de tres hipotètics classificadors y la precisió de l'*envolvent* fruit d'una combinació mitjançant un sistema d'arbitratge.

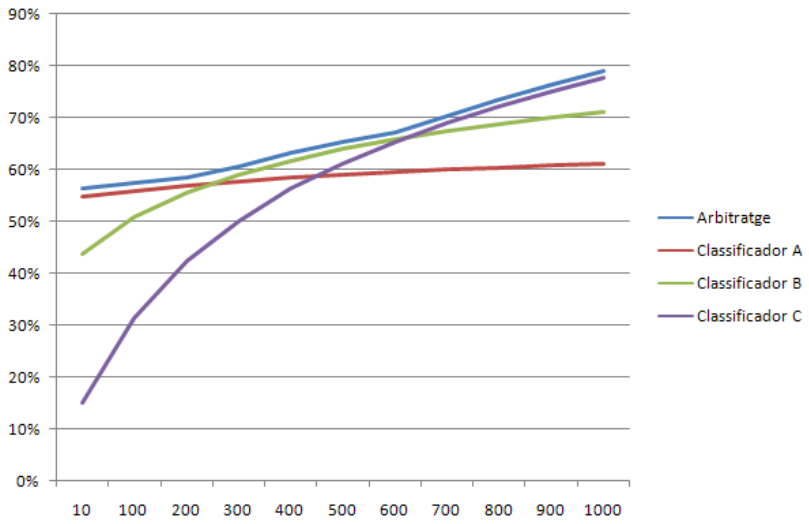

FIG. 60: Evolució de tres hipotètics classificadors i de la combinació fruit d'un arbitratge.

A més, aquest sistema pot aplicar-se simultàniament a qualsevol altre combinació de classificadors. Recordem que qualsevol sistema de votació permet millorar els resultats només si cap dels classificadors bàsics és molt millor que la resta. L'aplicació d'un arbitratge que seleccionés entre les sortides de tots els classificadors i la seva combinació, eliminaria el perill que amb el temps el sistema de votació perjudiqués el millor classificador. La **Fig. 61** mostra l'esquema d'un arbitratge aplicat a un sistema de votació. És important destacar que la facilitat amb la qual es poden avaluar incrementalment les precisions dels models, fa que la seva aplicació en qualsevol arquitectura sigui directa.

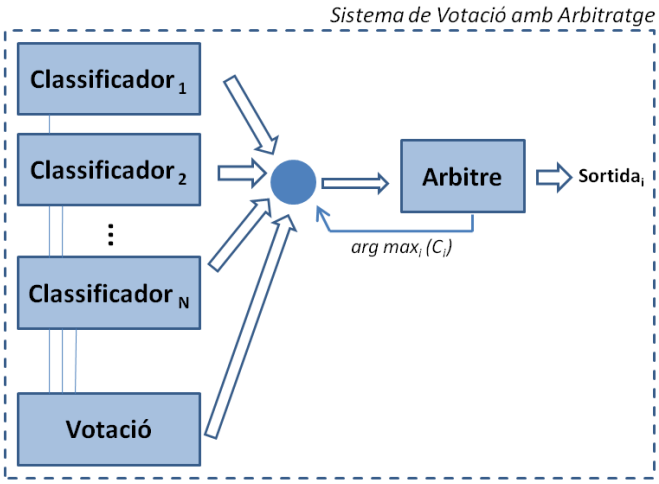

FIG. 61: Esquema d'un sistema de votació amb arbitratge.

7.4.6 STACKING

L'*stacking*[4] és una tècnica proposada a **[Wolpert, 1992]** que millora la idea de votació ampliant la típica combinació lineal, o votació ponderada, per una funció qualsevol que és apresa per un altre sistema combinador que optimitza els seus paràmetres durant l'entrenament. Es tracta d'una tècnica basada en el meta-aprenentatge, on un classificador aprèn a classificar, no a partir de les descripcions dels exemples, sinó a partir de les prediccions fetes per la resta de models. La **Fig. 62** mostra l'esquema general d'aquesta combinació.

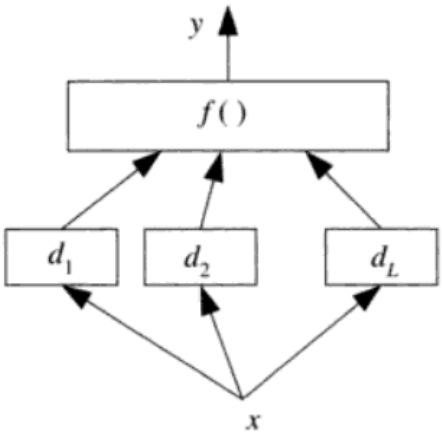

FIG. 62: Esquema d'una combinació *stacking*, basada en meta-aprenentatge. Font [Alpaydin, 2004].

El classificador final és entrenat per obtenir la sortida correcta a partir de les combinacions de les sortides dels classificadors del primer nivell. En principi no és aconsellable entrenar el classificador final amb els mateixos exemples que els classificadors bàsics, ja que es tracta que aprengui quins són els errors de classificació que cometen. Wolpert proposa utilitzar la tècnica *leave-one-out* **[8.5.2 Validació Creuada]**, però a causa del cost computacional que suposa quan es disposa de molts exemples a **[Alpaydin, 2004]** es recomana una simple *validació creuada*.

Una vegada entrenat el sistema, la classificació es realitza en dues fases. En primer lloc els classificadors bàsics són aplicats a l'exemple en qüestió, en segon lloc s'agrupen les prediccions fetes i s'utilitzen com a entrades del classificador final que emet la classe guanyadora. Com que la suposició és que cada classificador s'especialitza en un subconjunt del problema, la combinació de tots ells acostuma a ser més flexible i precisa.

7.5 COMBINACIONS MULTICLASSE

Algunes combinacions de classificadors, a més de per millorar la precisió final, s'utilitzen perquè permeten resoldre problemes multi-classe a partir de classificadors binaris. La idea és replantejar la tasca multi-classe de manera que pugui resoldre's a partir de decisions

[4] També conegut com a '*stacked generalization*'.

simples. Això és el que fa la *binarització simple*, la *binarització amb codis correctors* o la *classificació per parelles*.

7.5.1 BINARITZACIÓ SIMPLE

Una altra manera de combinar diferents classificadors és mitjançant la substitució d'un classificador multiclasse per un conjunt de classificadors binaris. En comptes d'entrenar un únic classificador per identificar totes les classes dels exemples es creen tants classificadors com classes i s'entrena cada un per reconèixer els exemples pertanyents a la classe assignada.

Com que cada classificador continua tenint la mateixa capacitat representacional, però ara la utilitza per resoldre un problema més simple, aquests classificadors especialitzats acostumen a tenir una precisió més elevada. La **Taula 4** mostra la transformació realitzada al problema, en comptes d'utilitzar un classificador per identificar les 4 classes, s'utilitzen 4 classificadors independents que han de retornar 1 si l'exemple pertany a la seva classe i 0 en cas contrari.

Classe	Sortides
1	1 0 0 0
2	0 1 0 0
3	0 0 1 0
4	0 0 0 1

TAULA 4: Equivalències entre les classes originals i les sortides dels classificadors combinats. Font [Witten & Frank, 2005].

A l'hora de classificar un nou exemple, s'apliquen tots els classificadors i, suposant que ofereixin algun *índex de confiança*, es tria la sortida més fiable. I aquest és precisament el principal inconvenient d'aquest sistema; requereix que els classificadors proporcionin un *índex de confiança* de les seves prediccions. Tot i així, és l'única manera de resoldre un problema multiclasse amb classificadors binaris.

7.5.2 BINARITZACIÓ AMB CODIS CORRECTORS

Una millora important respecte la *binarització simple* és la *binarització mitjançant codis correctors d'errors* **[Dietterich & Bakiri, 1995]**, que permet aplicar la mateixa tècnica a classificadors que no proporcionin cap *índex de confiança*. Els resultats proporcionats acostumen a ser tan bons que fins i tot és beneficiós utilitzar-lo en aquells casos on els classificadors poden tractar directament problemes multiclasse **[Witten & Frank, 2005]**.

Un codi corrector d'errors és un codi binari redundant que permet al receptor detectar i corregir una determinada quantitat de bits erronis. S'utilitzen habitualment en sistemes de transmissió d'informació, on el soroll present en el canal pot canviar un bit i modificar el codi. Es basen en la utilització d'*n* bits per representar un nombre de codis molt inferior a

la seva màxima capacitat (2^n de codis). Aquesta *baixa densitat* de codis permet utilitzar criteris de distància *Hamming* mínima per suposar quin era el codi original.

Per resoldre un problema multiclasse, igual que en el cas anterior, es realitza una transformació en les sortides que han d'obtenir els classificadors. Però utilitzant on codi corrector, o el que és el mateix, afegint classificadors redundants que permetin al combinador corregir possibles errors. La **Taula 5** mostra la transformació realitzada, en comptes d'utilitzar un classificador per identificar les 4 classes, s'utilitzen 7 classificadors independents que han de retornar 1 si l'exemple pertany a la seva classe i 0 en cas contrari.

Classe	Sortides	Codi Corrector
1	1 0 0 0	1 1 1 1 1 1 1
2	0 1 0 0	0 0 0 0 1 1 1
3	0 0 1 0	0 0 1 1 0 0 1
4	0 0 0 1	0 1 0 1 0 1 0

TAULA 5: Equivalències entre les classes originals, les sortides simples i les sortides amb codi corrector. Font [Witten & Frank, 2005].

Això permet classificar correctament exemples en els quals un o alguns dels classificadors comet un error. A l'hora de classificar un exemple s'apliquen tots els classificadors i les seves sortides binaries són concatenades per ser interpretades com un únic codi. Si el codi resultant és un codi vàlid, la classe guanyadora es pot obtenir directament. Si algun classificador comet un error i, per tant, el codi resultant no és cap dels vàlids, s'utilitza el criteri de distància mínima per reconstruir el codi i obtenir la classe correcta.

7.5.3 CLASSIFICACIÓ PER PARELLES

La *classificació per parelles*[5] **[Fürnkranz, 2002; Fürnkranz & Hüllermeier, 2003]** és una tècnica conceptualment molt simple que també permet resoldre tasques de classificació multiclasse mitjançant classificadors binaris. Es basa en la combinació mitjançant un sistema de votació dels resultats d'un grup de classificadors que només distingeixen entre parelles de les classes originals.

Si tenim una tasca de classificació entre N classes, la classificació per parelles requereix un total de $\frac{N(N-1)}{2}$ classificadors. Durant l'entrenament, a cada classificador, només se li mostren els exemples corresponents a les seves dues classes. Durant l'etapa de classificació, tots els classificadors s'apliquen al nou exemple, es comptabilitzen els vots assignats a cada classe, i es pren com a classe guanyadora aquella que hagi obtingut més vots.

La **Taula 6** mostra la relació entre el nombre de classes de la tasca i els classificadors necessaris per resoldre-la. Per tasques amb poques classes (2 o 3) el nombre de classificadors és equivalent al d'una binarització simple, per tasques mitjanes (de 4 a 6

[5] *Pair wise classification*

classes) el nombre de classificadors creix moderadament, però en afegir més classes el nombre de classificadors necessaris creix molt ràpidament.

Classes	Classificadors	Classes	Classificadors	Classes	Classificadors
1	-	**6**	*15*	**11**	*55*
2	*1*	**7**	*21*	**12**	*66*
3	*3*	**8**	*28*	**13**	*78*
4	*6*	**9**	*36*	**14**	*91*
5	*10*	**10**	*45*	**15**	*105*

TAULA 6: Relació entre el nombre de classes del problema i el nombre de classificadors necessaris per resoldre'l.

Tot i que pot semblar computacionalment molt costós no ho és gaire, ja que cada un dels classificadors només ha de ser entrenat amb els exemples pertanyents a les seves dues classes. Per això, en el cas que les classes estiguin relativament repartides, el cost computacional del seu entrenament és lineal, i equivalent al d'un únic classificador entrenat amb la totalitat dels exemples **[Witten & Frank, 2005]**. Així doncs, el problema de necessitar un gran nombre de classificadors està provocat per la memòria necessària per emmagatzemar els seus models, que pot ser assumible si s'utilitzen models de mida constant o que creixin linealment amb el nombre d'exemples.

8 TÈCNIQUES AUXILIARS

8.1 INTRODUCCIÓ

A més de la combinació i entrenament d'algorismes, l'Aprenentatge Automàtic (AA) està constituït per una sèrie de processos addicionals que són necessaris per facilitar, optimitzar i avaluar l'aprenentatge dels models. L'AAI també necessita aquestes tècniques auxiliars, però com és lògic han d'estar adaptades a aquest mode de funcionament.

En el paradigma *batch*, on l'entrenament es fa en una fase separada de la pròpia classificació, una part d'aquestes tasques cal fer-les abans de l'entrenament, bàsicament les que impliquen transformacions dels exemples. Altres tasques cal fer-les una vegada finalitzat l'entrenament, l'avaluació i l'optimització del model.

En el paradigma incremental la idea bàsica és la mateixa, especialment en les transformacions dels exemples, que tot i fer-se individualment per cada un d'ells, cal fer-les abans de mostrar-los a l'algorisme. Però al no existir una línia de separació que marqui la fi de l'entrenament, l'avaluació de l'algorisme i l'optimització del model cal fer-les de manera continuada.

FIG. 63: Procés d'aplicació de les tècniques auxiliars.

En aquest capítol es descriu una visió general de les diferències que presenten aquestes tasques a l'haver-se d'aplicar a processos incrementals. El primer punt introdueix quines transformacions poden fer-se a les representacions dels exemples, centrant-nos especialment en la discretització incremental. Després es presenten les tècniques incrementals de selecció de trets i les peculiaritats de l'ajust de paràmetres. I el darrer punt es centra en les tècniques d'avaluació, que permeten quantificar i visualitzar el comportament dels algorismes.

8.2 PREPROCESSAMENT

Abans de realitzar l'entrenament és habitual manipular les dades per modificar les representacions dels exemples. Aquestes modificacions de les dades originals serveixen per adaptar el seu format, per explicitar informació present, per introduir coneixement o per facilitar l'anàlisi a l'algorisme. A més d'aquestes transformacions dels trets, hi ha una tasca de preprocessament molt important: la selecció de trets. En determinats problemes el

nombre de trets pot ser molt elevat, i sovint aquests trets són o redundants o irrellevants, per això pot ser interessant simplificar la representació dels exemples incorporant únicament els trets més útils i significatius.

8.2.1 TRANSFORMACIONS

Les transformacions dels exemples, mitjançant la modificació dels seus trets, és una tasca fonamental per maximitzar la informació present i per facilitar a l'algorisme l'extracció del coneixement. La majoria d'aquestes transformacions són modificacions que afecten individualment cada un dels exemples, per això poden fer-se incrementalment sense requerir accés al conjunt d'entrenament[1].

Les transformacions més bàsiques consisteixen a realitzar modificacions numèriques a trets reals: sumant valors per desplaçar-los, multiplicant per escalar-los, o aplicant transformacions no-lineals com exponencials o logaritmes per deformar l'espai de trets. També poden intercanviar-se determinats valors simbòlics per uns altres o aplicar funcions lògiques als trets binaris.

Un altre grup de transformacions possibles són les que impliquen afegir nous trets, normalment calculats a partir de trets originals. A partir de trets numèrics és possible generar nous trets realitzant operacions entre ells, cosa que acostuma a obtenir relacions numèriques com ràtios o increments. A partir de trets simbòlics és possible recombinar diferents trets en un únic tret complex, producte vectorial dels dos anteriors. I a partir de trets binaris també és possible aplicar qualsevol operació lògica per obtenir nous trets indirectes.

A més d'aquestes transformacions que permeten explicitar informació i facilitar la seva anàlisi per part dels algorismes, hi ha un altre gran grup de transformacions habitualment utilitzat per adaptar les dades a les característiques de l'algorisme. Per exemple, no tots els algorismes poden tractar tots els tipus en les representacions dels exemples. Alguns algorismes no funcionen correctament amb trets simbòlics o amb trets numèrics, o amb trets que no estiguin definits per tots els exemples, en aquests casos la solució més senzilla és eliminar els trets conflictius.

Però això suposa perdre informació que podria ser molt rellevant, així que la solució recomanada és transformar els tipus dels trets, intentant conservar tanta informació com sigui possible. Les dues transformacions més habituals són la *binarització* i la *discretització*.

> **Binarització:** Aquest procés permet transformar trets numèrics i simbòlics a binaris. En el primer cas cada tret numèric és transformat en un únic bit: els valors no definits o amb valor 0, mantenen el seu valor original, la resta de valors

[1] Una important absència en les transformacions incrementals és la normalització; la impossibilitat de conèixer a priori el marge dinàmic d'un determinat tret no permet aplicar l'habitual normalització dels valors, un procés normalment rutinari.

numèrics es transformen en 1s. En el segon cas la transformació no suposa pèrdua d'informació: el tret simbòlic es transforma en un valor binari format per tants bits com símbols contenia el tret original; el bit corresponent val 1 i la resta 0.

Discretització: Aquest procés permet transformar valors numèrics en valors simbòlics, i és de gran importància en algorismes d'inducció de regles o d'arbres de decisió. La idea consisteix a reagrupar els valors numèrics en un reduït nombre d'intervals significatius que permetin generalitzar correctament. Però per fer això de manera eficient cal conèixer la distribució de les classes al llarg d'aquests valors, cosa que requeriria accedir al conjunt de les dades.

8.2.2 DISCRETITZACIÓ

A causa de la importància de la discretització per poder utilitzar models simbòlics amb dades numèriques, en els darrers anys s'han fet alguns avanços importants amb la proposta d'alguns algorismes incrementals que obtenen aproximacions prou fidels a l'hora de discretitzar trets numèrics.

A **[Pinto & Gama, 2005]** es proposa un procés incremental, anomenat PID (*Partition Incremental Discretization*), a partir d'un model de dos nivells. El primer nivell rep els valors dels trets i crea un model de la seva distribució estadística a partir d'un nombre d'intervals més gran del requerit. El segon nivell realitza la discretització final a partir de la distribució obtinguda pel nivell anterior. Aquest procés incremental pot discretitzar valors sense necessitat d'emmagatzemar-los en un temps i espai constant independentment del nombre d'exemples. A **[Pham & Afify, 2005]** es presenta un altre algorisme incremental de característiques similars.

Plantejat inicialment per treballar amb classificadors *Naïve Bayes*, però amb el mateix objectiu a **[Lu et al., 2006]** es presenta l'IFFD (*Incremental Flexible Frequency Discretization*), un algorisme incremental que discretitza els valors d'un tret numèric en una seqüència d'intervals de mida flexible, que permet la seva inserció i fusió incremental.

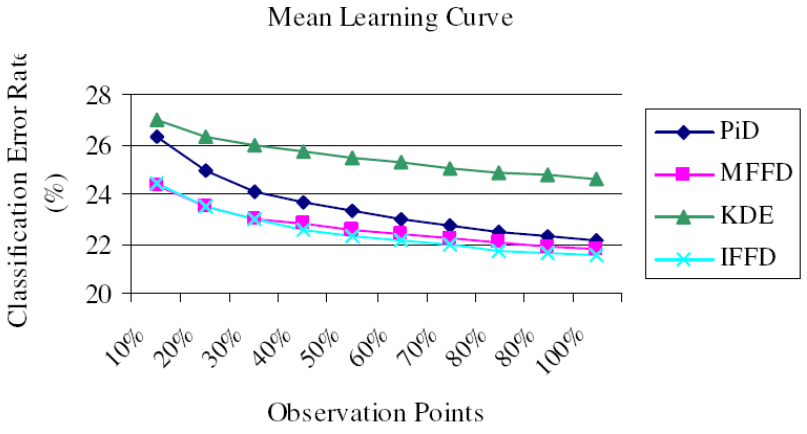

FIG. 64: Evolució dels errors d'un classificador *Naïve Bayes*, segons la discretització incremental utilitzada. Font [Lu et. al. 2006]

La **Fig. 64** mostra l'evolució de l'error d'un classificador *Naïve Bayes* aprenent a partir d'uns exemples discretitzats mitjançant diferents algorismes incrementals, s'observa com l'IFFD permet obtenir els millors resultats. A la **Fig. 65** es mostra una comparativa del temps necessari per cada un dels algorismes per discretitzar els exemples, es pot observar el reduït cost computacional necessari per executar el PID o l'IFFD.

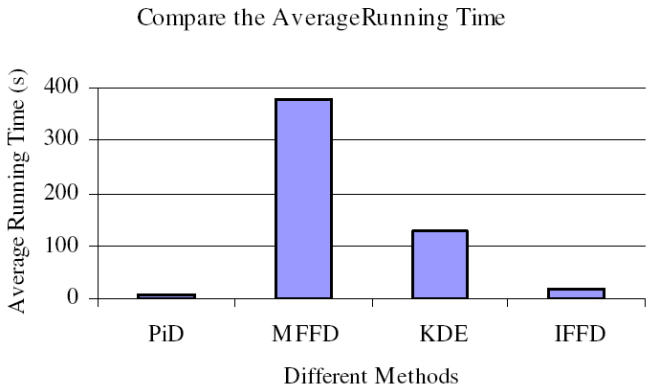

FIG. 65: Comparació del cost computacional dels 4 algorismes de discretització incremental. Font [Lu et. al. 2006]

8.3 SELECCIÓ DE TRETS

La *selecció de trets*[2], també coneguda com a *reducció de la dimensionalitat*[3], és un procés que consisteix en seleccionar un subconjunt dels trets originals més rellevants. El seu objectiu és facilitar l'aprenentatge del classificador gràcies a la reducció de l'espai de cerca i accelerar el procés reduint la seva càrrega computacional.

Els mètodes de selecció de trets poden classificar-se en dos grups, els *filtres*[4] o els *wrappers*. Els *filtres* són mètodes de selecció independent de l'algorisme d'aprenentatge que simplifiquen representacionalment els exemples eliminant trets poc informatius abans que arribin al classificador. Els *wrappers* es basen en el comportament d'un tipus de classificador per avaluar la qualitat del subconjunt **[Fleuret, 2004]**.

En aquells problemes de PLN amb una elevada dimensionalitat, la selecció de trets és una tasca fonamental que tradicionalment s'ha fet en mode *batch*, a partir de mesures informatives estadístiques obtingudes a partir de tot el conjunt de dades. Però aquest plantejament no és vàlid en entorns d'AAI, i cal buscar algorismes que puguin obtenir estimacions progressives de la importància dels diferents trets, sense accedir a priori al conjunt de trets de tots els exemples.

A continuació es presenten dues aproximacions incrementals a la selecció de trets. Tot i així, cal tenir en compte que aquestes tècniques només permeten determinar la utilitat, a

[2] *Feature selection*
[3] *Dimensionality reduction*
[4] *Filters*

l'hora de classificar l'exemple, d'un tret aïllat. Sense tenir en compte que, en ocasions, la combinació de dos trets poc informatius poden ser determinants per la classificació d'algunes classes. És per això que cal valorar amb prudència els pros i els contres de la selecció de trets.

8.3.1 INFORMACIÓ MUTUA

La *informació mútua* (IM) entre dues variables és un valor que mesura la seva interdependència. Intuïtivament, la IM és la informació que comparteixen dues variables aleatòries *X* i *Y*, és a dir, mesura el grau en el qual el coneixement sobre *X* ens permet saber sobre *Y*. En cas de ser dues variables independents la seva IM és zero, ja que conèixer el valor de *X* no ens permet reduir al incertesa de *Y*. Formalment es defineix segons la fórmula:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \cdot log\left(\frac{p(x,y)}{p_1(x)\, p_2(y)}\right)$$

Si s'aplica aquesta mesura a cada tret d'un exemple respecte la seva classe, permet estimar la dependència entre els trets i la classe, cosa que dóna una mesura de la seva capacitat predictiva. Per tant, la IM pot ser utilitzada com un criteri de selecció que permeti triar el subconjunt de trets que siguin més informatius que la resta. A més, el càlcul de les probabilitats condicionades i independents, necessàries per obtenir la IM, poden obtenir-se incrementalment a partir del simple comptatge d'esdeveniments.

Per exemple, a **[Leung & Gong, 2005]**, s'utilitza aquesta mesura de la dependència estadística entre els trets i la classe per reduir la dimensionalitat dels exemples seleccionant una fracció dels trets originals. A **[Fleuret, 2004]** fins i tot s'ha utilitzat per obtenir combinacions de trets que facilitin la classificació. L'algorisme utilitzat es basa en el que anomenen CMIM (*Conditional Mutual Information Maximization*), i consisteix en un procés iteratiu que comença seleccionant el millor tret i, progressivament, va afegint altres trets que, en combinació amb els anteriors, maximitza la seva capacitat de predicció. A **[Estévez et al., 2009]** s'utilitzen algorismes genètics per obtenir d'una manera semblant conjunts de trets significatius.

8.3.2 *EXTREMAL FEATURE SELECTION* (EFS)

Un altre indicador que pot utilitzar-se per detectar incrementalment els trets més informatius és l'EFS (*Extremal Feature Selection*) **[Carvalho & Cohen, 2006]**, basat en l'anàlisi dels pesos assignats per un classificador *Modified Balance Winnow (MBW)* **[6.6.2 Variants Winnow]**.

La idea és ordenar els trets segons la diferència absoluta entre els pesos del classificador positiu i el negatiu d'un *MBW*. Segons els autors, en qualsevol moment *t*, a partir del model positiu u_t i del negatiu v_t, es pot determinar la importància *I* del tret *j* com a:

$$I_t^j = \left|u_t^j - v_t^j\right|$$

Aquesta mesura permet aïllar els trets més significatius, amb pesos més elevats, a la part superior de la llista, i els menys significatius a la cua. Els autors afirmen que aquesta selecció basada en els pesos *extrems* és especialment bona en tasques de PLN i els seus resultats són comparables a les tècniques estàndard (*Chi-Square* o *Information Gain)*.

El fet de poder obtenir aquests resultats incrementalment obre la porta a realitzar una poda també incremental, eliminant del model aquells trets que menys es tenen en compte al calcular la combinació lineal. De totes maneres a **[Carvalho & Cohen, 2006]**, es descriu com una poda massa agressiva empitjora els resultats, i recomanen que tot i que la majoria dels trets seleccionats siguin els més significatius, es reservi una quota del 10% als trets menys significatius. Tot i que el motiu d'aquest fenomen no és clar, hi ha la sospita de que el procés de normalització del *MBW* estigui relacionat.

8.4 OPTIMITZACIÓ DE PARÀMETRES

Qualsevol algorisme d'AA inclou una sèrie de valors, coneguts com a paràmetres, que cal inicialitzar abans de la seva utilització. Aquests paràmetres permeten ajustar el comportament de l'algorisme d'inducció mitjançat la definició de mides màximes, llindars de decisió o ponderacions de diferents variables. Lamentablement, davant d'una tasca concreta, la utilització d'uns valors o uns altres pot determinar totalment la precisió aconseguida de l'algorisme induït.

L'optimització de paràmetres és un procés mitjançant el qual es realitza una cerca que permet obtenir el conjunt de paràmetres que maximitza la qualitat del classificador. En el model *batch* aquest procés acostuma a requerir la creació de diferents models (utilitzant diferents combinacions de valors) i les seves avaluacions corresponents. Tot i que es pot fer una cerca aleatòria, *prova i error*, el més habitual és utilitzar alguna cerca iterativa (algorismes genètics, aproximació per gradient descendent, ...) que permeti convergir en un *mínim local*[5], que maximitzi el rendiment. Però tots aquests mètodes es basen en la minimització de l'error global de predicció, és a dir, el promig de l'error en tot el conjunt de dades; una estratègia inviable amb algorismes incrementals.

A causa del requeriment de no emmagatzemar els exemples, l'única manera d'ajustar els paràmetres d'un algorisme de manera incremental és el desenvolupament en paral·lel de diferents classificadors. Si es creen models, amb diferents valors de paràmetres, i s'entrenen simultàniament a mesura que es van processant els exemples, és possible avaluar la seva precisió en temps real, i, per tant, saber quins són els paràmetres òptims.

[5] Seguint l'analogia en la que una optimització és un recorregut per la superfície de la funció a optimitzar, un *mínim local* és un punt d'aquesta superfície que es troba envoltat de punts amb valors de funció més alts. Però al ser local, no hi ha garanties de que es correspongui amb el mínim absolut de la funció i que, per tant, es tracti de la millor optimització.

Evidentment l'optimització incremental, o més ben dit simultània, consumeix importants recursos computacionals i de memòria. Això restringeix la seva aplicació a petits subespais de cerca, constituïts per unes poques dotzenes de valors; cosa que fa que la seva utilitat sigui molt menor que en l'optimització *batch*. En contrapartida es tracta d'una optimització dinàmica, que pot adaptar-se a l'evolució de les dades i modificar els valors òptims al llarg de la vida del sistema.

Tot i així, en alguns casos pot ser una tècnica vàlida, especialment si s'utilitzen algorismes d'aprenentatge que no consumeixin massa memòria a l'hora d'emmagatzemar els seus models. Per exemple, en el cas dels classificadors Winnow, que tenen models de mida constant, seria viable combinar un centenar de classificadors amb diferents paràmetres i seleccionar el que obtingui la precisió més elevada. Fins i tot podrien utilitzar-se models de creixement lineal, si el nombre de models s'anés reduint progressivament, eliminant els pitjors classificadors, a mesura que es processen més exemples i augmenten les mides dels models.

Finalment, fins i tot els algorismes més ineficients a l'hora d'emmagatzemar els seus models permeten l'optimització dinàmica d'alguns paràmetres. Els algorismes basats en memòria **[6.2.2 *K-Nearest Neighbour*]** necessiten definir el valor òptim de K i triar una determinada funció de distància. El que és interessant és que l'avaluació del classificador, per a diferents combinacions de K i diferents mètriques de similitud, pot fer-se a partir d'un únic model; ja que aquests paràmetres no afecten l'emmagatzemament dels exemples, sinó la seva anàlisi.

En termes més generals, tot i que els algorismes *lazy learners* **[6.2 Aprenentatge Basat en Memòria]** utilitzen molta memòria per emmagatzemar els exemples observats, el fet que no processin els exemples representa un avantatge a l'hora de treballar amb variants del model. Si es té en compte que els paràmetres de l'algorisme són paràmetres *interpretatius*, és a dir que afecten com el classificador analitza el seu model, es pot veure com l'avaluació de diferents variants no suposa un increment de memòria. Això permet que a partir d'un únic repositori d'exemples, s'executin diferents classificadors *virtuals* basats en diferents paràmetres i que s'avaluïn en temps real per triar el més adequat en cada moment. Per tant, la creació i avaluació de diferents variants de *lazy learners* no suposa cap increment de la memòria necessària, únicament un increment lineal del cost computacional a l'hora de fer les classificacions.

8.5 AVALUACIÓ

Un dels aspectes fonamentals de l'aproximació rigorosa a l'aprenentatge automàtic és l'avaluació dels sistemes d'aprenentatge, és a dir, el conjunt de tècniques i metodologies que permeten quantificar amb precisió la bondat d'un model. Les mesures habituals intenten reflectir diferents aspectes dels classificadors, principalment la seva precisió i capacitat predictiva, però també la interpretabilitat de les seves representacions, el cost computacional i els recursos de memòria necessaris **[Aha, 1995]**. En aquesta secció es

presenten les principals mètriques i tècniques, adaptades als algorismes incrementals, utilitzades per descriure quantitativament el comportament d'un classificador.

8.5.1 MÈTRIQUES ESTÀNDARD

Curiosament, en els inicis de la recerca en AA els esforços es van centrar a avaluar i maximitzar la *precisió aparent*, és a dir la precisió de la tasca en relació a les dades d'entrenament. No va ser fins més endavant que alguns investigadors van apuntar al problema de la *sobre-adaptació*[6], situació que es dóna quan el classificador crea un model tan fidel a les dades d'entrenament que perd la capacitat de ser aplicat satisfactòriament a altres dades. Aquesta situació es produeix quan la capacitat expressiva del model, en relació a la quantitat d'exemples, és prou gran com per poder *memoritzar* els exemples incloent excepcions i soroll, cosa que redueix la seva qualitat davant d'exemples desconeguts. Per això es va acabar establint com a norma que l'avaluació dels sistemes calia fer-la respecte la *precisió real*, és a dir, la precisió mesurada respecte a un conjunt de dades diferents a les utilitzades durant l'entrenament **[Aha, 1995]**.

Aquesta divisió entre dades d'entrenament i dades d'avaluació és fonamental per poder parlar d'aprenentatge. Actualment aquesta dada s'anomena *precisió de generalització*[7], i mesura la precisió que assoleix un classificador induït a partir de les dades d'entrenament quan és utilitzat per classificar un conjunt de dades desconegut. A més, no només permet determinar la qualitat de la inducció feta a partir dels exemples d'entrenament, sinó que permet detectar la *sobre-adaptació* quan els dos valors difereixen considerablement.

A **[Witten & Frank, 2005]** es considera que, al no existir una fase d'entrenament diferenciada, en els sistemes incrementals pot evitar-se el problema. Segons l'autor no és necessari separar el corpus en dues parts, una per entrenament i una per avaluació, ja que la predicció es realitza abans de conèixer la resposta. En certa manera tots els exemples poden utilitzar-se com avaluació, abans de mostrar-los la classe correcta, i com a entrenament, després de donar-los la solució. Això permet utilitzar el 100% dels exemples per entrenament i el 100% dels exemples per avaluació.

Però si es vol ser rigorós cal tenir en compte que la precisió d'un classificador incremental no és constant, i que previsiblement l'error de classificació en els primers exemples serà molt superior a l'error després dels primers milers d'exemples. Per això aquestes mètriques de generalització calculades durant l'entrenament són, de fet, els valors promitjats al llarg de la vida del model. Si es volgués comparar la qualitat d'un classificador incremental, en un determinat moment, amb la d'un altre classificador *batch*, seria necessari avaluar tots dos respecte un mateix conjunt d'exemples independent.

---

[6] *Over fitting*
[7] *Generalization accuracy*

Independentment de si l'avaluació es realitza sobre el mateix conjunt d'entrenament o sobre un conjunt de dades separat, les fórmules per calcular les diferents mètriques són les mateixes:

PRECISIÓ I COBERTURA

Les dues mètriques utilitzades habitualment per mesurar la qualitat d'un classificador binari són la *precisió*[8] i la *cobertura*[9].

La *precisió* reflecteix la fiabilitat de les etiquetes assignades, dóna una idea de la proporció d'exemples classificats com a positiu que realment eren positius. La *cobertura* reflecteix la capacitat del classificador a l'hora de trobar els exemples positius, dóna una idea de la proporció d'exemples positius existents que han estat classificats com a positius.

Però abans de mostrar les definicions formals cal presentar la terminologia utilitzada a l'hora de classificar els quatre resultats possibles en una tasca de classificació:

True Positives (tp): Exemples positius classificats correctament com a positius.

False Positives (fp): Exemples negatius classificats errròniament com a positius.

True Negatives (tn): Exemples negatius classificats correctament com a negatius.

False Negatives (fn): Exemples positius classificats errròniament com a negatius.

A partir d'aquesta classificació és possible definir més formalment aquestes mesures:

$$Precisió = \frac{tp}{tp + fp}$$

$$Cobertura = \frac{tp}{tp + fn}$$

MESURA-F

El problema de les dues mesures anteriors és que cap d'elles per separat pot reflectir la qualitat d'un sistema. És senzill aconseguir un classificador amb una precisió del 100%, i una cobertura del 0%, a partir d'un classificador hiper-especialitzat que només detecti un cas. També ho és aconseguir un classificador amb una cobertura del 100%, i una precisió del 0%, a partir d'un classificador que etiqueti tots els exemples com a positius.

Per descriure la bondat d'un classificador cal una mesura que combini els dos valors, i això és el que fa exactament la *mesura-F*[10] **[Chinchor, 1992]**. És una mètrica que reflecteix la qualitat conjunta d'un classificador, mitjançant la mitja harmònica de la precisió i la cobertura:

[8] *Precision*
[9] *Recall*
[10] *F-measure*

$$F = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R}$$

On β és la importància relativa de la cobertura respecte la precisió. En alguns problemes aquest factor és important perquè permet, per exemple, maximitzar la precisió a costa de la cobertura, penalitzant els falsos positius i minimitzant els falsos negatius.

Com que en la majoria de tasques tots dos factors són igual d'importants, s'opta per una mesura equilibrada on β és igual a 1, de manera que s'obté la *mesura-F1*[11]:

$$F1 = \frac{2 \times P \times R}{P + R}$$

DESVIACIÓ ESTÀNDARD

Les mesures anterior acostumen a presentar una important variabilitat al repetir l'avaluació amb diferents dades, diferents paràmetres o algorismes no-deterministes. Per això habitualment no n'hi ha prou amb un únic valor per reflectir la qualitat del classificador, i és preferible conèixer la seva distribució estadística.

La solució més estesa es repetir l'avaluació diverses vegades, amb diferents dades o paràmetres d'inicialització, i expressar la mesura mitjançant el seu valor mitjà i la seva desviació estàndard:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

On x_i és cada una de les mesures obtingudes en els N experiments diferents; com en molts altres processos de mesura l'error acostuma a seguir una desviació normal, on el 95% dels casos cauen en un interval de ±2σ al voltant de la mitjana.

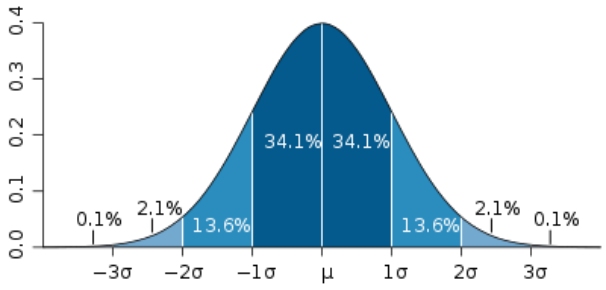

FIG. 66: Distribució de la desviació estàndard

[11] *F1-measure*

8.5.2 VALIDACIÓ CREUADA

Com s'ha explicat en el punt anterior la mesura de la precisió i cobertura d'un classificador presenta una important variabilitat, ja que la inducció d'un model a partir d'exemples és sensible a molts factors, entre ells les decisions estocàstiques que prenen molts algorismes a l'hora de crear el model. Però fins i tot en aquells casos on la inducció és plenament determinista, el model resultant està molt condicionat per les dades utilitzades com a exemples. Si, a més, el que volem fer és quantificar la qualitat del seu funcionament com a classificador, els exemples patró respecte els quals l'avaluem també poden introduir més variabilitat.

Per tant, a l'hora de mesurar el rendiment d'un classificador (ja sigui mitjançant la precisió, cobertura o qualsevol altre mètrica) la variabilitat introduïda per la selecció dels exemples d'entrenament i validació pot ser molt elevada. La solució metodològica a aquest problema és repetir l'entrenament i l'avaluació amb diferents dades i promitjar els resultats, cosa que permet obtenir estimacions més acurades.

A més, en els algorismes incrementals, és important minimitzar la influència de l'ordre de la seqüència d'entrenament, per això una vegada definit un conjunt d'entrenament és necessari reordenar-lo aleatòriament per obtenir diferents seqüències d'entrenament i poder promitjar els seus resultats.

M-FOLD CROSS-VALIDATION

La metodologia més estesa per minimitzar els biaixos introduïts per la selecció dels conjunts d'entrenament i d'avaluació és coneix com a *10-fold-cross-validation*. Consisteix en dividir el conjunt de dades en 10 parts de la mateixa mida i, cíclicament, seleccionar una de les parts com a conjunt d'avaluació i les 9 restants com a conjunt d'entrenament. Els valors promitjats en els 10 cicles són una bona estimació estadística de la qualitat real del sistema inductiu. A més, aquesta tècnica garanteix que tots els exemples del conjunt inicial són utilitzats una vegada com a exemple d'avaluació.

Si es generalitza aquest procés per particions de diferent mida podem formalitzar el procés: si tenim un conjunt D constituït per m dades d'entrenament, podem dividir-lo en K subconjunts disjunts d_i a partir dels quals construir K conjunts d'entrenament D_i on:

$$D_i = \{x \in D: x \notin d_i\}$$

El cost computacional d'aquest procés és proporcional a K vegades el temps necessari per entrenar l'algorisme amb el conjunt de dades principal, més el temps necessari per classificar les dades d'avaluació:

$$O\left(K \left(t_i \frac{m(K-1)}{K} + t_c \frac{m}{K} \right) \right) \approx O(m\, t_i\, (K-1) + m\, t_c) = O\, (m\, (t_i\, (K-1) + t_c)$$

Si considerem que els temps d'entrenament d'una instància és comparable al seu temps de classificació, podem simplificar la fórmula anterior i observar que el temps és proporcional al producte del nombre d'instàncies i el nombre de conjunts:

$$O(\, m(\, t_i(K-1) + t_c) \approx O(\, m\, (t(K-1) + t) = O\, (m\, t\, K)$$

Finalment, en aquells casos en els quals el nombre d'exemples és molt reduït és aconsellable utilitzar el que es coneix com *Leave-One-Out*[12], una variant de la validació creuada en la qual s'utilitza correlativament un dels exemples com a validació i la resta com a entrenament. Aquesta variant és equivalent a una *M-Fold Cross-Validation* en la qual el nombre de divisions *K* es fa coincidir amb el nombre d'exemples *m* **[Kohavi, 1995]**.

VALIDACIÓ CREUADA INCREMENTAL

Un dels problemes de la validació creuada tradicional és que l'algorisme d'aprenentatge s'ha d'executar K vegades, cosa que augmenta en un factor K el temps d'execució. Però a **[Kohavi, 1995]** es presenta una optimització per aquells algorismes *incrementals* que també siguin *decrementals*, és a dir que puguin tant aprendre com desaprendre a partir d'exemples individuals.

En aquest article es proposa una versió incremental de la validació creuada que en comptes d'entrenar i avaluar l'algorisme K vegades amb les diferents parts del corpus, l'entrena una sola vegada amb totes les dades del conjunt, i per cada una de les K parts *desapren* els seus exemples, avalua la qualitat, i torna a aprendre els exemples eliminats. Segons Kohavi, si l'algorisme genera el mateix model en mode *batch* que en mode incremental, és a dir, si és independent de l'ordre dels exemples, la *validació creuada incremental* obté exactament els mateixos valors que la versió tradicional.

En aquest cas, el temps d'execució necessari és el següent:

$$O\big(T + m(t_i + t_c + t_d)\big)$$

on *T* és el temps d'entrenament per totes les dades, *m* és el nombre d'exemples, t_i el temps d'inserció/entrenament d'un exemple, t_c el temps de classificació i t_d el temps d'eliminació/desentrenament. Si *T* és proporcional al nombre d'exemples es pot aproximar per $m \cdot t_i$, i considerem comparables els temps d'aprenentatge, de desaprenentage i de classificació, el temps total d'execució és aproximadament:

$$O\big(m\, t_i + m(t_i + t_c + t_d)\big) \;=\; O\big(m(2t_i + t_c + t_d)\big) \approx O(4mt)$$

És a dir, tot i que el temps d'avaluació és proporcional al nombre d'exemples, ha deixat de ser dependent del nombre de parts *K* en què es divideix el conjunt de dades, i per tant, té el mateix cost computacional avaluar amb deu o amb cent particions, per la qual cosa directament podríem substituir-lo per un *Leave-One-Out*. Cal tenir en compte però que

[12] També conegut com a "*Jackknife*"

alguns estudis **[Zhang, 1992; Shao, 1993]** indiquen que per alguns classificadors els models seleccionats mitjançant *Leave-One-Out* presenten una gran variabilitat i que pot ser preferible utilitzar l'avaluació creuada amb una quantitat moderada (K=10 o K=15) de particions.

8.5.3 CORBES D'EVOLUCIÓ

Els classificadors induïts per algorismes *batch* s'avaluen en funció de la qualitat de les seves classificacions donant per fet que és un propietat estàtica. És a dir, una característica avaluable al finalitzar l'entrenament i representativa del comportament futur del sistema. Però això no és així en un sistema incremental, en el qual l'entrenament és continu i no existeix com a fase diferenciada.

Com que el model evoluciona al llarg de l'entrenament, també ho fa la seva qualitat. Inicialment els seus resultats són pobres, però a mesura que adquireix coneixement a partir dels exemples els seus resultats van millorant progressivament. Si es representen els diferents valors al llarg del temps, s'obté una *corba d'aprenentatge* **[Fig. 67]** que reflecteix l'evolució de la qualitat del sistema al llarg de la seva vida.

A causa d'aquest comportament dinàmic, és habitual que l'avaluació dels classificadors incrementals es visualitzi com una corba, tant a l'hora de representar la precisió, la cobertura, la mesura-F, com l'error de classificació. Com que es tracta de sistemes asíncrons no té sentit mostrar la seva evolució respecte el temps, i la variable independent utilitzada en l'eix horitzontal és el nombre d'exemples d'entrenament. Aquest gràfic permet visualitzar el cost d'anotació necessari per assolir una determinada qualitat.

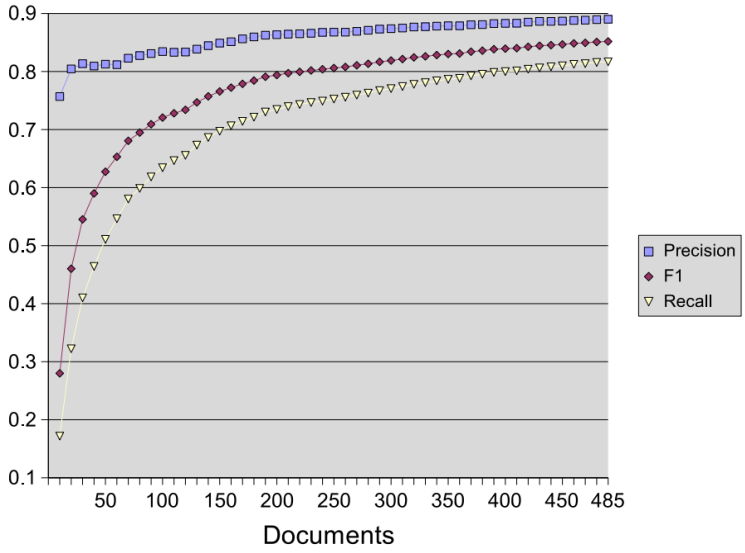

FIG. 67: Evolució de la qualitat d'un sistema d'aprenentatge incremental. Font [Siefkes, 2005].

En sistemes d'Aprenentatge Actiu, on els exemples d'entrenament són una fracció del corpus total, és habitual mostrar l'evolució respecte la mida total del corpus, especialment si es volen comparar els resultats amb els d'un sistema passiu que processa tots els exemples

del corpus. A la **[Fig. 68]** es mostra un exemple d'aquesta comparativa per dos classificadors (Winnow i Basat en Memòria) tant amb entrenament Actiu com Passiu; l'eix horitzontal indica el nombre d'exemples anotats.

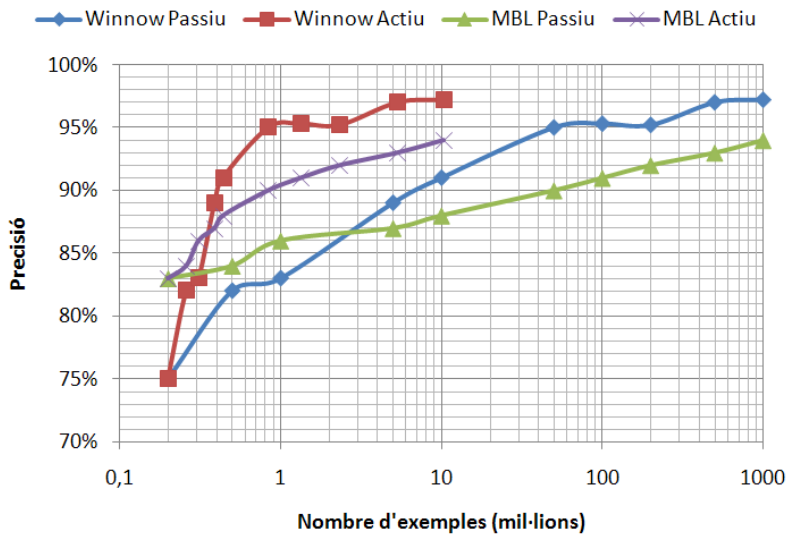

FIG. 68: Precisió de l'aprenentatge Actiu/Passiu en funció dels exemples anotats. Dades [Banko & Brill, 2001].

En altres situacions pot ser interessant veure l'evolució d'altres relacions entre variables, tant de qualitat obtinguda com de recursos utilitzats.

En aquells algorismes que utilitzen models de mida variable, és a dir que no es mantenen constants, és útil visualitzar la dependència entre la mida del model (quantitat de nodes, de regles o de casos) respecte el nombre d'exemples d'entrenament **[Fig. 69]**. I representar la seva evolució directa, per constatar el tipus de creixement (constant, log, lineal, exponencial, ...), o la seva ràtio, interpretable com una mesura de la compressió d'informació realitzada pel model, i detectar en quins punts l'aprenentatge es transforma en *memorització*.

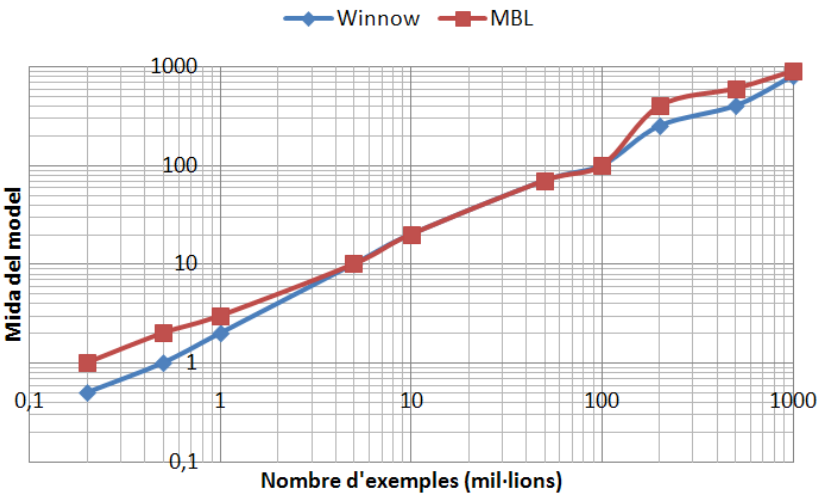

FIG. 69: Grau de compressió: creixement del model segons exemples d'entrenament. Dades de [Banko & Brill, 2001].

També pot ser útil visualitzar gràficament les relacions entre altres variables, com la precisió en funció de la mida del model, cosa que permet obtenir *punts òptims* on es maximitza la precisió en relació als recursos necessaris **[Fig. 70]**. Aquestes gràfiques permeten optimitzar classificadors que hagin de treballar en entorns amb limitacions de recursos, cosa que permet aconseguir la màxima eficiència. O, visualitzar la relació entre la precisió i el nombre d'exemples anotats, per obtenir el *cost* d'anotació necessari per obtenir un determinat nivell de precisió **[Fig. 71]**.

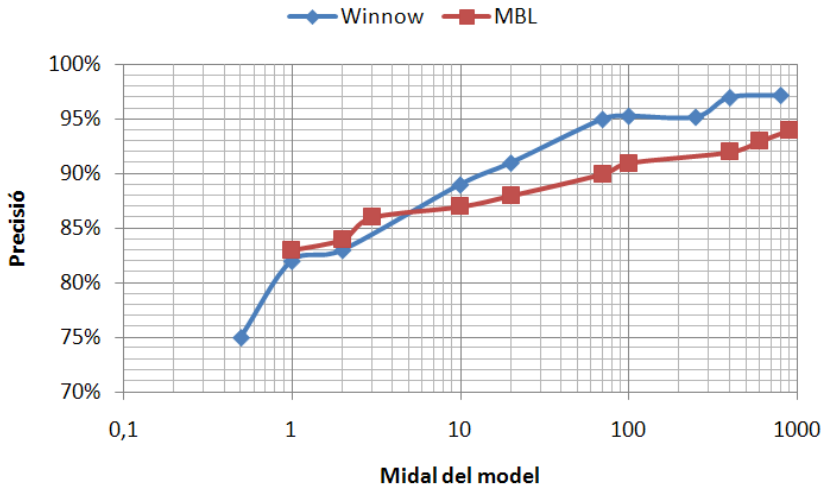

FIG. 70: Precisió segons els recursos assignats al model.
Dades [Banko & Brill, 2001].

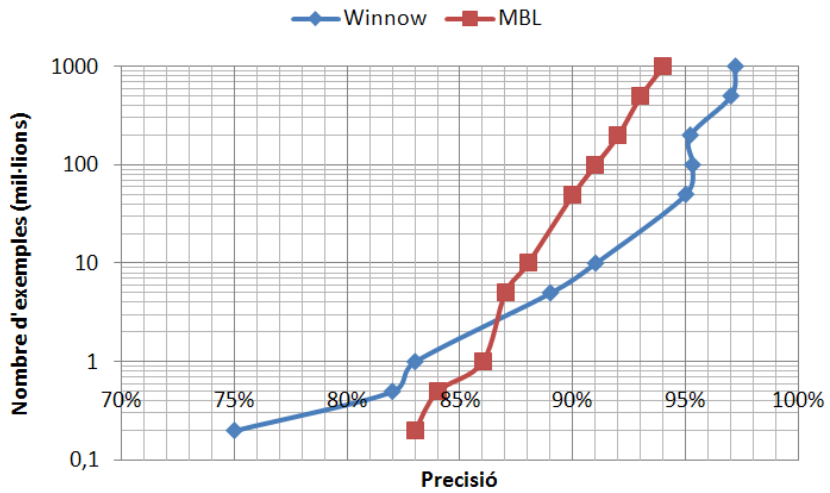

FIG. 71: Exemples necessaris per obtenir una determinada precisió.
Dades [Banko & Brill, 2001].

PART III:

L'APRENENTATGE INCREMENTAL EN PLN: PROVES EXPERIMENTALS

9 ENTORN EXPERIMENTAL

9.1 INTRODUCCIÓ

Una vegada descrit el marc teòric que defensa els beneficis d'aplicar l'Aprenentatge Automàtic Incremental, i després de justificar la seva viabilitat tècnica mitjançant la presentació de l'estat de la qüestió, és realitzaran diferents proves quantitatives que validin o contrastin les tesis proposades al llarg d'aquest treball.

Per portar-ho a terme s'han triat diverses tasques, totes elles representatives del PLN però variades per les seves característiques intrínseques (dificultat, nivell lingüístic, mida de l'etiquetari, complexitat de la representació, ...). Aquestes tasques s'han utilitzat com a base per induir automàticament una sèrie de models classificadors mitjançant diferents tècniques d'entrenament incremental. L'objectiu és obtenir mesures que permetin comparar els resultats i determinar els beneficis d'aquestes tècniques en relació a la precisió assolida pel model i a l'eficiència amb que han utilitzat els exemples d'entrenament.

Aquests resultats haurien de permetre determinar si, mitjançant tècniques d'Aprenentatge Automàtic Incremental, és possible obtenir entorns d'Anotació Inter-Activa més eficients per al desenvolupament de classificadors de PLN. És a dir, entorns que permetin accelerar l'entrenament, reduir el cost d'anotació, i al mateix temps assolir una precisió equivalent.

En aquest capítol es presenta l'entorn experimental que s'ha utilitzat per a dur a terme aquesta validació. En primer lloc es descriu el programari utilitzat en les diferents fases, és a dir, en el processament dels corpus, en la pre-selecció de tasques i en les avaluacions de les diferents tècniques d'entrenament incremental. En segon lloc es descriuen els corpus textuals utilitzats per a generar les tasques de PLN i els diferents processos realitzats per obtenir les corresponents dades d'entrenament .

9.2 PROGRAMARI UTILITZAT

Per portar a terme aquests experiments s'han utilitzat diferents eines de programari, tant per processar els corpus originals i generar les dades d'entrenament de les diferents tasques, com per analitzar-ne la seva validesa i fer-ne una selecció preliminar, com per portar a terme els experiments finals i analitzar en profunditat els resultats.

9.2.1 SCRIPTS PHP (*HYPERTEXT PRE-PROCESSOR*)

El PHP és un llenguatge de programació interpretat utilitzat per crear scripts que s'executen a un servidor o en local a la mateixa línia de comandes. Les seves característiques en velocitat i gestió dinàmica de memòria el fan un bon candidat per processar grans quantitats de text d'una forma àgil i senzilla.

Per aquests motius tot el processament automàtic dels corpus originals, així com els filtres, transformacions, creacions de diccionaris, tractaments de les dades, generació dels corpus de cada tasca i creació dels fitxers amb les corresponents dades d'entrenament, ha estat realitzat mitjançant scripts a mida programats amb aquest llenguatge.

A la secció **[9.3 Corpus Textuals]** s'explica amb detall quins han estat aquest processos i els *scripts* desenvolupats per a aquest projecte.

9.2.2 WEKA (*WAIKATO ENVIRONMENT FOR KNOWLEDGE ANALYSIS*)

El programari conegut com a *Weka* **[Hall et al., 2009]** és un conjunt de llibreries programades en Java i una interfície gràfica dissenyada per experimentar i fer proves amb una amplíssima gama d'algorismes d'aprenentatge automàtic. Tot i que inicialment va ser pensada per a tasques de mineria de dades amb el temps ha acabat incloent la majoria d'algorismes utilitzats en tasques de *clustering*, classificació i regressió. Aquesta eina va ser desenvolupada a la Universitat de Waikato i és àmpliament coneguda a l'entorn acadèmic, fins al punt de que les seves APIs i formats de dades (`.ARFF`) s'han convertit en un estàndard *de facto*.

Aquesta eina ha sigut de gran ajuda a l'hora de fer una pre-selecció de les tasques sobre les quals portar a terme els experiments finals, ja que ha facilitat la realització d'una gran quantitat d'experiments preliminars i la selecció dels trets òptims per representar cada una de les tasques.

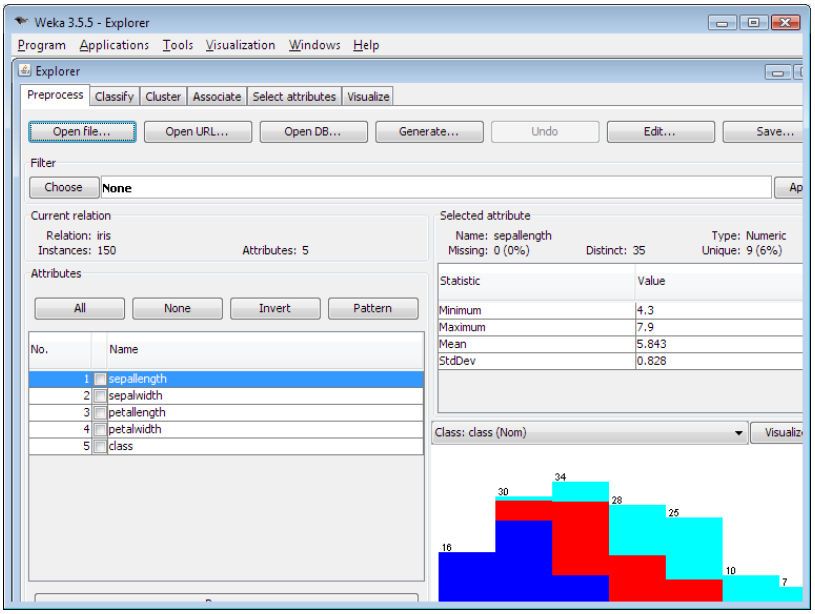

FIG. 72: Captura de la interfície gràfica del *Weka*.

9.2.3 ICE (*INCREMENTAL CLASSIFIER ENVIRONMENT*)

L'*ICE* és un programari desenvolupat pel propi autor en llenguatge C# per l'entorn .Net, dissenyat especialment per avaluar algorismes d'aprenentatge incremental. Està format per una aplicació que s'executa des de la línia de comandes **[Fig. 73]** que gestiona diferents llibreries especialitzades: per a la importació de dades format `.ARFF` i `.CSV`, la gestió automàtica de diferents tipus d'entrenaments (*Batch, Incremental, ActiveLearning*), l'aplicació de diferents tècniques d'avaluació (*Holdout, Sampling, KFold, LeaveOneOut*), i l'obtenció de tota mena de mètriques (*Error, Accuracy, Precission, Recall, F-Measure, ...*) i informes estadístics (*Graphs, Reports, ConfussionMatrix, ...*).

A més defineix una API per desenvolupar algorismes classificadors que puguin utilitzar-se amb aquesta eina. Per a aquest treball s'ha implementat una versió incremental d'un classificador *Naïve Bayes* capaç de tractar representacions mixtes que continguin trets numèrics i trets nominals. Aquest algorisme ha estat validat contra el *Weka* amb diferents conjunts de dades per confirmar que s'obtenien exactament els mateixos resultats.

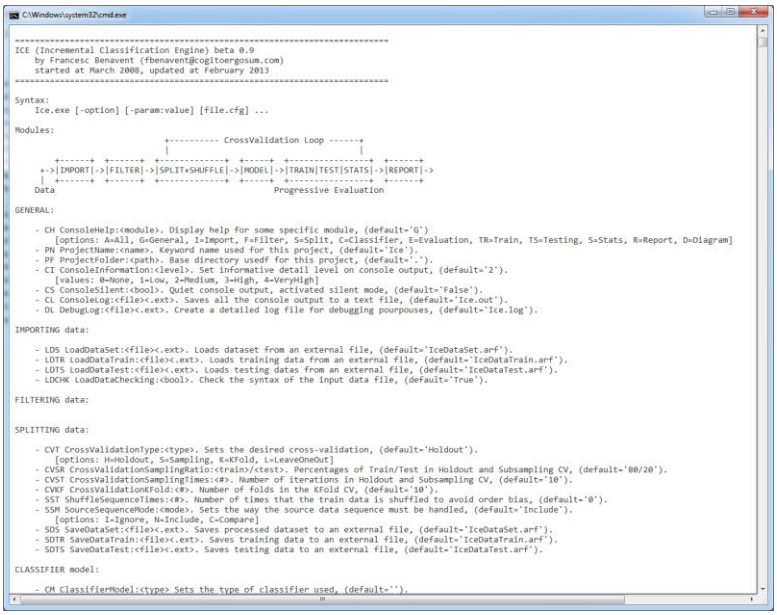

FIG. 73: Captura del resultat a la línia de comandes del programa *ICE*.

Finalment, cal destacar que la utilització d'aquesta eina ha estat imprescindible ja que ofereix algunes funcionalitats fonamentals difícils d'obtenir amb altres:

- Entrenament Incremental, que permet entrenar els models mostrant a l'algorisme els exemples de manera seqüencial.
- Aprenentatge Actiu, que permet entrenar els models únicament amb aquells exemples què el classificador dubta.
- Avaluació Progressiva, que permet avaluar el classificador en diferents moments al llarg de l'entrenament i per tant obtenir corbes d'avaluació.

- Avaluació de la distribució i precisió dels graus de certesa del classificador, que permet conèixer els efectes interns de diferents tècniques d'entrenament.

9.3 CORPUS TEXTUALS

Per generar les tasques de PLN, en forma d'uns conjunts d'exemples d'entrenament, calia partir de corpus textuals anotats amb informació lingüística, estructural i/o tipogràfica. A més, per motius que veurem més endavant, era necessari que els corpus estiguessin formats per parells de corpus equivalents. És a dir, formats per texts formalment diferents però amb anotacions conceptualment equivalents, per exemple, corpus bilingües anotats amb criteris similars. Per això finalment es van triar els dos corpus bilingües que es descriuen a continuació.

9.3.1 CORPUS A: *ANCORA*

L'*AnCora* **[Martí et al., 2008]** és un corpus textual bilingüe català (*AnCora-CA*) i espanyol (*AnCora-ES*). Està constituït per textos periodístics amb un total aproximat de 500.000 paraules per a cada llengua i inclou diferents nivells d'anotació lingüística:

- lema i categoria morfològica
- constituents i funcions sintàctiques
- estructura argumental i papers temàtics
- classe semàntica verbal
- tipus denotatiu dels noms deverbals
- sentits de WordNet nominals
- entitats nombrades
- relacions de coreferència

Per a les tasques previstes n'hi ha hagut prou amb la informació lexicomorfològica, de manera que s'ha processat el corpus per obtenir-ne una versió reduïda amb la informació estrictament necessària per a cada una de les tasques.

	AnCora-CA	AnCora-ES
Llengua	Català	Espanyol
Fitxers	1.550	1.636
Oracions	16.500	17.400
Entitats	13.000	11.000
Tokens	500.000	530.000

TAULA 7: Descripció quantitativa del corpus *AnCora*.

En un primer procés es generen, a partir dels 1.550 arxius de l'*AnCora-CA* i els 1.636 arxius de l'*AnCora-ES*, una sèrie de fitxers amb les dades corresponents a cada llengua

(`Catalan.*` i `Spanish.*`) i a la combinació de totes dues (`Catalan-Spanish.*`) amb les següents extensions:

- **`*.TAG`**: Llistat de les etiquetes morfosintàctiques trobades al corpus i la freqüència absoluta d'aparició. Aquesta llista permetrà detectar errors en l'anotació o limitar el nombre d'etiquetes morfosintàctiques o n-grames als valors més freqüents.
- **`*.LEX`**: Diccionari de formes amb els lemes, etiquetes morfosintàctiques i freqüències absolutes per a cada una de les lectures possibles. Aquest diccionari permetrà determinar la lectura més freqüent per cada forma i utilitzar aquest valor com a etiqueta per defecte.
- **`*.UNK`**: Diccionari de formes desconegudes, és a dir, formes que només apareixen dins de noms propis complexos (*PN multitoken*) i que per tant no disposen d'informació morfosintàctica individual.[1] Aquesta llista serà editada manualment per incorporar el lema i l'etiqueta morfosintàctica d'aquestes formes i ampliar el diccionari de formes amb els fitxers `*.LEX2`.

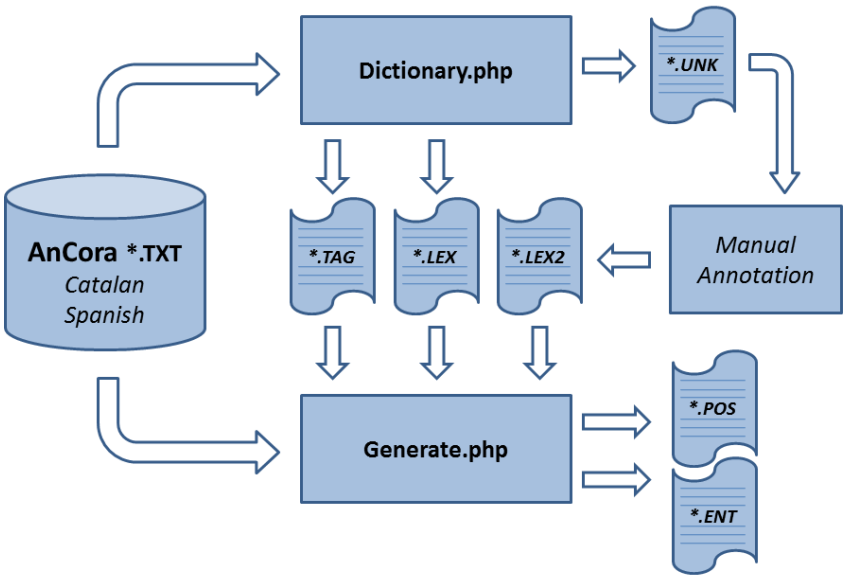

FIG. 74: Diagrama de processament de l'*AnCora* i obtenció dels corpus A1(POS) i A2 (ENT).

La informació morfosintàctica s'ha codificat mitjançant una versió compacte de l'etiquetari original de l'AnCora, formada per 122 etiquetes que inclouen les categories majors, les categories menors i alguns trets discriminants, però se n'han eliminat els trets de gènere, nombre i persona. A l'**[Annex A.1]** es pot consultar el llistat complet de l'etiquetari utilitzat.

[1] Cal tenir en compte que a l'*AnCora* les entitats anomenades (*named entities*) apareixen com un únic element sense tokenitzar.

També s'ha inclòs informació tipogràfica relativa al tipus de caràcters (majúscules, minúscules, dígits, puntuació, ...) i a la longitud de la forma mitjançant l'etiquetari descrit a l'**[Annex A.2]**.

9.3.1.1 CORPUS A1 (POS)

En un segon procés, i utilitzant tant els fitxers originals de l'*AnCora* com els diccionaris i taules de freqüències generats en el pas anterior, es genera el corpus A1. Dos únics fitxers que contenen la totalitat de l'*AnCora-CA* i de l' *AnCora-ES* respectivament, en un format verticalitzat que inclou la informació necessària per realitzar tasques de desambiguació morfosintàctica.

Concretament, per cada *token* (o element unitari de processament) s'inclou l'ordre que ocupa a l'oració, la forma original, la forma normalitzada, informació tipogràfica, el lema i l'etiqueta morfosintàctica (*Part-of-Speech*) per defecte segons el diccionari, el lema i l'etiqueta morfosintàctica real amb què estava anotada i, per acabar, una informació que indica si la lectura assignada per defecte no coincideix amb la real (`[Error]`), o en cas de coincidir si ha estat l'única lectura present al diccionari (`[Only]`) o la més freqüent (`[Default]`).

A la taula següent es mostra un fragment representatiu d'un dels fitxers (`Catalan.pos`) generats a partir de l'*AnCora-CA*:

<#>	<Text>	<Form>	<Typo>	<defLemma>	<defPoS>	<realLemma>	<realPoS>	<Comm>
1	Jordi_Pujol	jordi_pujol	[M9]	jordi_pujol	NP	jordi_pujol	NP	[Only]
2	inaugurarà	inaugurarà	[L9]	inaugurar	VmIF	inaugurar	VmIF	[Only]
3	la	la	[L2]	el	DA	el	DA	[Default]
4	famosa	famosa	[L6]	famós	AQ	famós	AQ	[Only]
5	escultura	escultura	[L9]	escultura	NC	escultura	NC	[Only]
…	…	…	…	…	…	…	…	…

TAULA 8: Fragment del corpus A1 generat a partir de l'*AnCora*.

9.3.1.2 CORPUS A2 (ENT)

Mitjançant un procés similar s'ha generat un segon corpus A2. Dos únics fitxers que també contenen totes les oracions presents a l'*AnCora* en què s'han expandit les entitats anomenades. Concretament, s'han processat tots els tokens complexos (*elements multitoken*) etiquetats com a noms propis (`PN`) i s'han substituït per una seqüencia de tokens individuals anotats segons la lectura per defecte per a cada forma. Finalment s'ha afegit una etiqueta final segons l'anotació *BIO* per identificar la ubicació d'aquestes entitats, segons es tracti d'un token inicial (*[B]egin*), un token intern (*[I]n*) o un token extern (*[O]ut*) a una entitat.

El resultat final A2 és similar al corpus anterior però amb les entitats multitoken expandides i amb els tokens etiquetats segons el seu rol individual. A la taula següent es mostra un fragment representatiu d'un dels fitxers (`Catalan.ent`) generats a partir de l'*AnCora-CA*:

<#>	<Text>	<Form>	<Typo>	<defLemma>	<defPoS>	<realLemma>	<realPoS>	<Comm>
1	Jordi	jordi	[T5]	jordi	NP	jordi	NP	[B]
1	Pujol	pujol	[T5]	pujol	NP	pujol	NP	[I]
2	inaugurarà	inaugurarà	[L9]	inaugurar	VmIF	inaugurar	VmIF	[O]
3	la	la	[L2]	el	DA	el	DA	[O]
4	famosa	famosa	[L6]	famós	AQ	famós	AQ	[O]
5	escultura	escultura	[L9]	escultura	NC	escultura	NC	[O]
…	…	…	…	…	…	…	…	…

TAULA 9: Fragment del corpus A2 generat a partir de l'*AnCora*.

9.3.2 CORPUS B: DE-NEWS

El segon corpus utilitzat és el *German-English Parallel Corpus De-News* **[Koehn, 2000]**. Un corpus bilingüe anglès-alemany format per més de 60.000 oracions en cada idioma, segmentades i tokenitzades. Les oracions han estat extretes de notícies d'una web alemanya i traduïdes a l'anglès manualment.

La versió utilitzada, inclou 9.756 notícies publicades entre l'agost de 1996 i el gener del 2000, i està disponible en tres formats de text: texts plans (`*.TXT`), texts segmentats (`*.PRE`) i texts alineats a nivell d'oració (`*.AL`). Tot i no incloure informació lingüística, el fet d'estar segmentades i tokenitzades ens permetrà definir diverses tasques ortotipogràfiques.

	De-News ENG	De-News GER
Llengua	Anglès	Alemany
Fitxers	1.118	1.118
Oracions	62.475	66.317
Tokens	1.175.526	1.017.064
	Corpus B Eng	**Corpus B Ger**
Oracions	50.271	54.254
Tokens	1.080.000	935.000

TAULA 10: Descripció quantitativa del corpus *De-News*.

El corpus s'ha processat automàticament per descartar els títols i seleccionar únicament les oracions del cos de les notícies acabades amb puntuació final ('.', '?' i '!'). També s'han realitzat algunes transformacions per modificar el format, reconstruir espais inter-tokens i corregir alguns errors de tokenització. El resultat és una llista d'oracions segmentades, tokenitzades de forma no ambigua, i amb marcadors de finals d'oració.

En primer lloc s'han processat els 1.118 arxius de cada idioma (`*.AL`) per generar una diccionari de paraules guionitzades (`*.HYP`), comptabilitzar les freqüències dels tokens presents al final de les oracions (`*.END`) i, especialment, separar els títols de les notícies (`*.TIT`) de les oracions del cos principal (`*.SEN`).

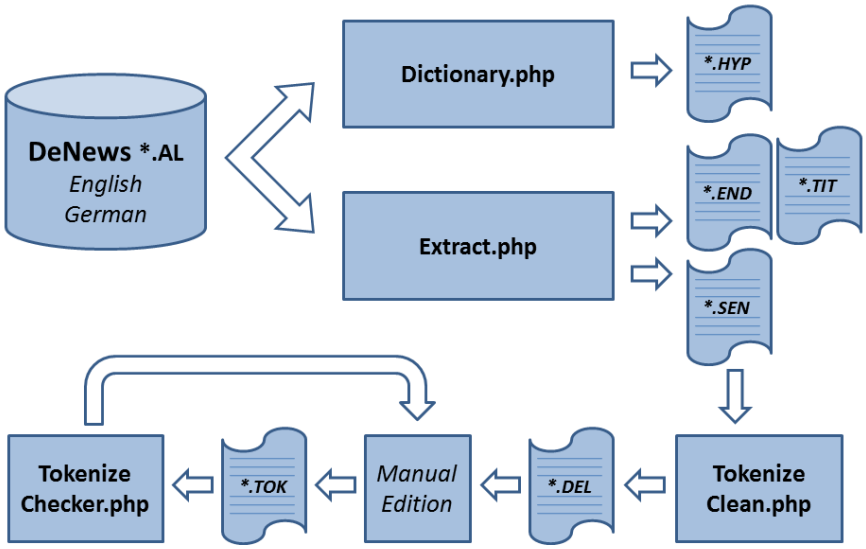

FIG. 75: Diagrama de processament del *DeNews* i obtenció dels corpus B (TOK).

Les oracions del cos principal s'han processat per transformar la tokenització per defecte (`*.SEN`), separada per espais, en una tokenització canònica (`*.DEL`), marcada explícitament, que respecti els espais presents al text original. Aquests fitxers (`*.TOK`) s'han revisat manualment, amb l'ajuda d'un corrector ortotipogràfic *ad-hoc*, en un procés iteratiu fins obtenir els fitxers finals (`English.TOK` i `German.TOK`) que formen el corpus B.

A la taula següent es mostren uns fragments representatius dels fitxers (`English.TOK` i `German.TOK`) del corpus B obtingut a partir del corpus *De-News*:

```
#|<token1>|<token2>|<token3>|<token4>|<token5>|...|<tokenN>|#
#|Priebke| |left| |the| |court| |as| |a| |free| |man|.|#
#|The| |the| |83|-|year|-|old| |presiding| |judge|,| |Mr.| |Quistelli|,| |…
#|Shortly| |after| |3| |p.m.|,| |the| |crisis| |management| |group| |in| |…
…

#|<token1>|<token2>|<token3>|<token4>|<token5>|...|<tokenN>|#
#|Borchert| |betonte|,| |Deutschland| |sei| |BSE|-|frei|.|#
#|Christoph| |Bergner| |(|CDU|)|:| |"|Was| |mich| |neben| |der| |eigenen|…
#|Die| |39|-|jaehrige| |Bause| |tritt| |an| |die| |Stelle| |von| |Ruth| |…
…
```

TAULA 11: Fragments del corpus B generat a partir del corpus *De-News*.

Una vegada obtingut un corpus B bilingüe, segmentat, tokenitzat i normalitzat, el transformarem en tres corpus verticalitzats (anomenats B1, B2 i B3) que inclouran la informació necessària per definir tres senzilles tasques ortotipogràfiques.

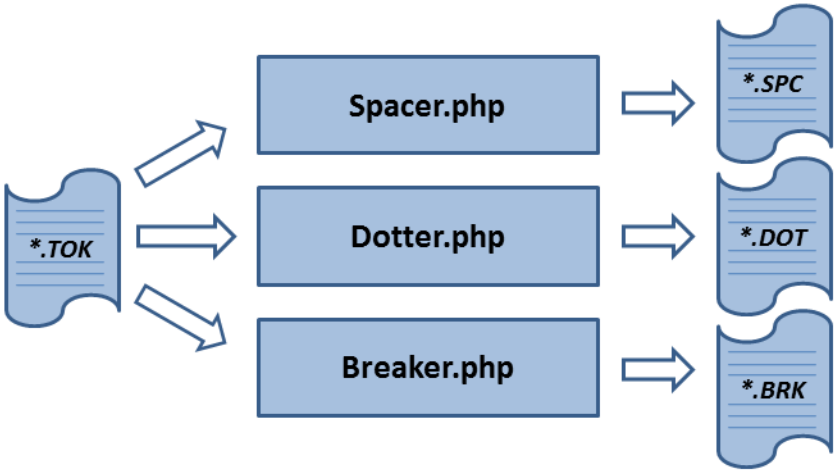

FIG. 76: Diagrama de processament per obtenir els corpus B1(SPC), B2(DOT) i B3(BRK).

9.3.2.1 CORPUS B1 (SPC)

Mitjançant un procés addicional (`Spacer.php`) es verticalitza el text, s'eliminen les línies corresponents als espais i s'incorpora la informació relativa a cada token. Concretament s'inclou el text original, la forma normalitzada, un pseudo-lema que explicita els signes de puntuació, i l'etiqueta tipogràfica corresponent al tipus i quantitat de caràcters (veure **[Annex A.2]**).

La darrera columna és una etiqueta que indica si el token anava seguit d'un espai que el separés del token següent o si es tracta de tokens consecutius. Aquestes dades permetran entrenar un classificador capaç de reconstruir ortotipogràficament els espais eliminats en una seqüència de tokens.

<#>	<Text>	<Form>	<defLemma>	<Typo>	<FollowedBySpace>
1	He	he	he	[T2]	True
2	claimed	claimed	claimed	[L7]	False
3	,	,	[COMMA]	[X1]	True
4	however	however	however	[L7]	False
5	,	,	[COMMA]	[X1]	True
6	to	to	to	[L2]	True
7	have	have	have	[L4]	True
8	acted	acted	acted	[L5]	True
9	in	in	in	[L2]	True
10	an	an	an	[L2]	True
11	emergency	emergency	emergency	[L9]	True
12	situation	situation	situation	[L9]	False
13	,	,	[COMMA]	[X1]	True
14	short	short	short	[L5]	True
15	of	of	of	[L2]	True
16	superior	superior	superior	[L8]	True
17	orders	orders	orders	[L6]	False
18	.	.	[PERIOD]	[X1]	False
…	…	…	…	…	…

TAULA 12: Fragment del corpus B1(SPC) generat a partir del B i *De-News*.

9.3.2.2 CORPUS B2 (DOT)

Amb un procés similar (`Dotter.php`) es verticalitza el text, es respecten els espais i s'incorpora la informació de cada token: el text original, la forma normalitzada, un pseudo-lema que explicita els signes de puntuació, i l'etiqueta tipogràfica corresponent al tipus i quantitat de caràcters (veure **[Annex A.2]**).

La darrera columna és una etiqueta que indica si el token anava seguit d'un delimitador d'oració. Aquestes dades haurien de permetre entrenar un desambiguador de punts, de manera que discrimini entre els punts finals d'oració i els punts interns com els de les abreviatures.

<#>	<Text>	<Form>	<defLemma>	<Typo>	<FollowedByEOL>
1	The	the	the	[T3]	False
2			[SPACE]	[SP]	False
3	judge	judge	judge	[L5]	False
4			[SPACE]	[SP]	False
5	furthermore	furthermore	furthermore	[L9]	False
6			[SPACE]	[SP]	False
7	brought	brought	brought	[L7]	False
8			[SPACE]	[SP]	False
9	in	in	in	[L2]	False
10			[SPACE]	[SP]	False
11	extenuating	extenuating	extenuating	[L9]	False
12			[SPACE]	[SP]	False
13	circumstances	circumstances	circumstances	[L9]	False
14	.	.	[PERIOD]	[X1]	True
15			[SPACE]	[SP]	False
…	…	…	…	…	…

TAULA 13: Fragment del corpus B2(DOT) generat a partir del B i *De-News*.

9.3.2.3 CORPUS B3 (BRK)

Finalment amb un procés molt semblant a l'anterior (`Breaker.php`) es verticalitza el text respectant els espais i s'incorpora la informació de cada token: el text original, la forma normalitzada, un pseudo-lema, l'etiqueta tipogràfica (veure **[Annex A.2]**) i una etiqueta final que indica si el token anava seguit d'un marcador de final d'oració. La principal diferència és que en aquest cas les oracions es reordenen aleatòriament, el que permet recombinar els contexts dret i esquerre dels finals d'oració i, per tant, generar un corpus sintètic molt més gran que l'original. Aquestes dades haurien de permetre entrenar un detector de final d'oració, també conegut com a *Sentence Boundary Detector*, a partir d'una seqüència de tokens sense informació lingüística.

<#>	<Text>	<Form>	<defLemma>	<Typo>	<FollowedByEOS>
1	Priebke	priebke	priebke	[T7]	False
2			[SPACE]	[SP]	False
3	left	left	left	[L4]	False
4			[SPACE]	[SP]	False
5	the	the	the	[L3]	False
6			[SPACE]	[SP]	False

7	court	court	court	[L5]	False
8			[SPACE]	[SP]	False
9	as	as	as	[L2]	False
10			[SPACE]	[SP]	False
11	a	a	a	[L1]	False
12			[SPACE]	[SP]	False
13	free	free	free	[L4]	False
14			[SPACE]	[SP]	False
15	man	man	man	[L3]	False
16	.	.	[PERIOD]	[X1]	True
17			[SPACE]	[SP]	False
…	…	…	…	…	…

TAULA 14: Fragment del corpus B3(BRK) generat a partir del B i *De-News*.

9.4 DADES D'ENTRENAMENT

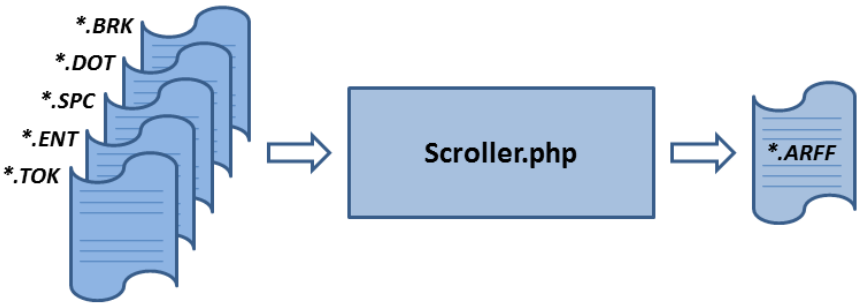

FIG. 77: Diagrama de processament per obtenir els corpus B1(SPC), B2(DOT) i B3(BRK).

Finalment, es processen els corpus verticalitzats per realitzar una doble transformació. Per un costat es recorre la seqüència de tokens verticalitzats i es contextualitza cada un d'ells mitjançant una finestra mòbil de ±3 tokens, és a dir, una finestra que inclou el token classificat, els tres tokens anteriors i els tres tokens posteriors. Cada un d'aquests contexts, juntament amb la seva etiqueta final, constitueix un exemple d'entrenament. I aquests exemples es formaten segons les especificacions dels fitxers `.ARFF` utilitzats pel programari *Weka* **[Hall et al., 2009]** com a dades d'entrenament, dades que seran el punt de partida per definir les tasques experimentals del següent capítol.

<Num$_i$>,
[Info Token$_{i-3}$], [Info Token$_{i-2}$], [Info Token$_{i-1}$],
[Info Token$_i$],
[Info Token$_{i+1}$], [Info Token$_{i+2}$], [Info Token$_{i+3}$],
<Label$_i$>

TAULA 15: Estructura de cada un dels exemples d'entrenament contextualitzats.

9.5 CONCLUSIONS

Per a assolir els objectius d'aquesta tesi i validar les seves hipòtesis és imprescindible realitzar experiments sobre dades reals, experiments que permetin quantificar els beneficis aportats per les diferents tècniques d'entrenament incremental i extrapolar-los a altres tasques de PLN. Aquest capítol s'ha centrat en descriure l'entorn experimental utilitzat per obtenir aquestes dades empíriques.

En primer lloc els diferents programaris utilitzats en cada una de les tres etapes: el PHP en el processament dels corpus per a obtenir les dades d'entrenament, l'entorn *Weka* en la preselecció de les tasques i de la seva representació de trets i, finalment, l'entorn *ICE* per a avaluar les diferents tècniques d'entrenament incremental i l'anàlisi dels seus resultats.

En segon lloc, s'han presentat els dos corpus bilingües triats, l'*AnCora* i el *De-News*, i els processos a partir dels quals s'han creat cinc conjunts de dades d'entrenament adaptats a cinc tasques de PLN: dos de l'AnCora, per a l'anotació morfosintàctica (POS) i detecció d'entitats anomenades (ENT), i tres del De-News, per a la inserció d'espais (SPC), desambiguació de punts (DOT) i segmentació d'oracions (BRK).

10 TASQUES SELECCIONADES

10.1 INTRODUCCIÓ

Una vegada obtinguts els cinc conjunts de dades d'entrenament cal definir exactament com seran presentades les tasques als classificadors. Cal tenir en compte que els conjunts de dades en brut inclouen trets correlats entre si, informativament redundants, i que sovint és possible seleccionar un subconjunt d'aquests trets que obtinguin un millor resultat.

Per a això, per a cada una de les tasques, s'han analitzat diferents combinacions de trets fins aïllar aquell subconjunt que permet al classificador obtenir la màxima precisió a l'hora de classificar els exemples.

Per portar a terme aquests experiments s'ha utilitzat l'entorn *Weka* que permet carregar les dades originals i eliminar fàcilment determinats trets. Per a cada una de les combinacions de trets avaluades, el sistema entrena un model *Naïve Bayes* i avalua la seva precisió en forma d'error de classificació, és aquesta mesura la que s'ha utilitzat per comparar les diferents representacions de cada tasca.

L'objectiu d'aquestes dades d'entrenament és validar el comportament de diferents tècniques d'entrenament incremental i extrapolar els resultats a altres aplicacions de PLN. Però per poder fer-ho amb rigor és necessari que les dades d'entrenament siguin representatives, tant en la forma com en el comportament, de les tasques típiques en aquest camp.

Concretament, s'ha considerat que per les tasques siguin vàlides i permetin extreure conclusions generalitzables, han de complir els dos requeriments següents:

a) La tasca ha de ser *soluble*: és a dir, les dades han de permetre que el classificador indueixi un model amb un error raonablement petit. És considera raonablement petit si és del mateix ordre de magnitud que l'estat de l'art de la tasca corresponent o, significativament inferior a una *baseline* de referència.
b) La tasca no ha d'estar *saturada*:[1] és a dir, les dades han de ser suficientment complexes com per què les dades proporcionades no hagin saturat el model. O, en altres paraules, per poder mesurar els beneficis de les tècniques incrementals, cal que el model es situï en un punt de la corba d'aprenentatge on encara sigui sensible a una variació en la quantitat d'exemples observats.

[1] Es considera que un model està *saturat* quan la presentació de més exemples durant una entrenament addicional, no proporciona cap millora observable en la seva precisió ni redueix significativament el seu error.

En aquest capítol es descriuen, per a cada una de les tasques, les diferents combinacions de trets avaluades i els resultats obtinguts, així com la representació seleccionada per a les posteriors avaluacions de les tècniques d'entrenament.

També es mostren els resultats de les comprovacions del compliment dels dos requeriments anteriors: que la tasca sigui *soluble*, verificant que la precisió final sigui superior a la de determinades *baselines*, i que no estigui *saturada*, verificant que la precisió disminueix a mesura que ho fa la mida del corpus d'entrenament.

10.2 MODEL NAÏVE BAYES

Com s'ha explicat al llarg del **[Capítol 6. Algorismes Incrementals]** els models estadístics, i concretament el *Naïve Bayes*, són un bon model sobre el qual realitzar els experiments incrementals. Recordem que tant la robustesa i precisió dels models induïts, la simplicitat d'implementació d'una versió incremental com, finalment, la possibilitat de quantificar el grau de certesa amb el que el model assigna una etiqueta determinada, fan que sigui idoni per portar a terme aquesta validació **[6.5.1 Naïve Bayes]**.

Així doncs, inicialment s'ha realitzat un estudi preliminar utilitzant l'entorn *Weka* per mesurar l'error final dels models induïts per diferents combinacions de trets i triar el que proporcioni millors resultats. Aquesta validació es realitza mitjançant la tècnica coneguda com *10-fold cross validation* **[8.5.2 Validació Creuada]** en la qual l'entrenament i avaluació del classificador es repeteix 10 vegades per obtenir la mitjana dels resultats; prèviament les dades s'havien dividit en 10 subconjunts i en cada iteració s'utilitza un d'aquests subconjunts com a corpus d'avaluació i els 9 restants com a corpus d'entrenament.

Cal tenir en compte que l'entorn *Weka* està dissenyat principalment per treballar algorismes d'aprenentatge automàtic *batch*, per tant, independentment que s'utilitzi un algorisme *batch* o incremental (*updateable*, segons la seva terminologia) l'entorn realitza una única avaluació al finalitzar l'entrenament i proporciona les dades finals.

Però més enllà de no tenir accés a l'evolució del model, en el cas de l'algorisme de *Naïve Bayes* no representa cap problema, ja que el model final és independent de l'ordre de la seqüència d'entrenament i coincideix amb el que s'obté mitjançant la versió *batch* de l'algorisme. Per això els resultats finals proporcionats pel *Weka* són perfectament extrapolables als obtinguts amb l'*ICE*.

Finalment, donat que la implementació de l'algorisme *Naïve Bayes* de l'*ICE*, només accepta la utilització de trets numèrics i nominals, s'ha hagut d'adaptar la representació transformant el que podrien haver estat trets *bag of words* en trets nominals amb un nombre finit de valors. A efectes pràctics això vol dir que aquells trets que presentaven un elevat nombre de valors, per exemple els lemes, les formes o els tri-grames, s'han hagut de reduir a els N valors més freqüents i englobar la resta en un valor *catch all* (`[*]`).

10.3 TASCA 1: PART-OF-SPEECH TAGGING (POS)

10.3.1 DESCRIPCIÓ

La tasca T1 (POS) basada en les dades d'entrenament del corpus A1 (POS) es correspon a una tasca típica de PLN, l'etiquetació morfosintàctica segons etiquetes de *Part-of-Speech*. El mig milió d'exemples disponibles es distribueixen segons la taula següent:

Tokens	POS-Catalan	POS-Spanish
Tagged	497,000 (100%)	530,000 (100%)
Single PoS	218,000 (43.9%)	286,000 (53.9%)
Ambiguous	278,000 (56.0%)	242,000 (45.8%)
- Default	258,000 (51.9%)	222,000 (41.9%)
- Error	20,300 (4.09%)	20,500 (3.87%)

TAULA 16: Descripció quantitativa del corpus de la Tasca 1 (POS).

És a dir, per al corpus català i espanyol respectivament, el 43,9% i 53,9% dels exemples corresponen a formes no ambigües, amb una única lectura, i per tant trivialment etiquetables. El 51,9% i el 41,9% dels exemples corresponen a formes ambigües però etiquetades amb la lectura per defecte, és a dir, amb la més freqüent segons el diccionari. I només el 4,09% i el 3,87% dels exemples corresponen a lectures minoritàries.

Tenint en compte que els trets amb índex `[0]` es corresponen al token a etiquetar i que els índex negatius i positius es corresponen al context esquerre i dret respectivament, els trets disponibles a les dades són:

- **`Typo[-3,+3]`** Informació tipogràfica de tots els tokens de la finestra.
- **`Lemma [-3,+3]`** Lema per defecte de tots els tokens de la finestra.
- **`PoS[-3,-1]`** Categoria gramatical per defecte del context esquerra.
- **`PoS[0]`** Categoria gramatical per defecte del token central.
- **`PoS[+1,+3]`** Categoria gramatical per defecte del context dret.
- **`+ngrams`** Bigrames i trigrames immediatament al voltant.
- **`+minigrams`** Igual que l'anterior però només amb categories majors.

10.3.2 RESULTATS (ETIQUETADOR POS)

La primera prova va consistir a aplicar una anotació trivial per obtenir una *baseline* de referència a partir dels errors obtinguts pels algorismes *ZeroRule*[2] i *OneRule*[3]:

[2] *ZeroRule*: classificador que aplica a tots els exemples l'etiqueta globalment més freqüent.
[3] *OneRule*: classificador que selecciona el tret més informatiu i assigna l'etiqueta més freqüent per cada un dels possibles valors d'aquest tret.

Model (Overall)	**Error CAT**	**Error SPA**
ZeroRule: all [NC]	82.07%	86.67%
OneRule: *f*(Lemma[0])	27.13%	28.47%
OneRule: *f*(PoS[0])	**4.09%**	**3.91%**

TAULA 17: Errors de les *baselines* de la Tasca 1a (POS), corpus global.

S'observa com degut a la naturalesa multietiqueta de la tasca, l'assignació de la categoria més freqüent (el 17,9% dels tokens són noms comuns) és insuficient i obté un error del 82%. Això obliga a utilitzar una *baseline* una mica més sofisticada que, en el cas de basar-se en el lema del token a etiquetar obté un error d'un 27,13% però que si té accés a l'etiqueta per defecte es redueix fins a un 4,09%. Cal observar que aquest valor és exactament el tant per cent de tokens ambigus amb lectures diferents a les més freqüents.

Donada la influència del tret `PoS[0]`, que és l'etiqueta per defecte assignada pel diccionari, es va voler provar fins a quin punt era necessària per poder realitzar la tasca. Es realitzen diferents experiments sense incloure aquest tret i s'obtenen els resultats següents:

Features (+Typo[] +Lemma[])						**Error CAT**	**Error SPA**
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams		
200	Yes	-	Yes	-	-	16.66%	19.00%
200	Yes	-	Yes	Yes	-	18.33%	20.52%
200	Yes	-	Yes	-	Yes	19.57%	21.81%
1000	Yes	-	Yes	-	-	14.27%	-
1000	Yes	-	Yes	Yes	-	14.74%	-
1000	-	-	-	Yes	-	**12.84%**	-

TAULA 18: Errors de les proves preliminars de la Tasca 1a (POS), sense PoS[0].

El valor N correspon al nombre màxim de valors diferents per a aquells trets amb una llista oberta de possibles valors (text, forma, lema, ...). S'observa com en augmentar aquest valor es redueix l'error. La combinació de trets que més redueix l'error és la utilització dels trets bàsics, la utilització dels n-grames per sobre dels *mini-grames*[4] i, curiosament, l'eliminació de les categories gramaticals, tant del context dret com l'esquerre. És possible que el fet que aquesta informació ja estigui disponible indirectament en els n-grames contradigui excessivament la premissa bàsica del *Naïve Bayes* en relació a la independència dels trets. En qualsevol cas l'error obtingut és molt superior a la *baseline* i per tant, de moment, no és una tasca vàlida.

Demostrada la importància del tret `PoS[0]`, es repeteixen els experiments però incloent aquesta informació entre els trets disponibles, els resultats són els següents:

[4] El que en aquest treball s'anomena mini-grames no és més que els n-grames d'una versió simplificada de l'etiquetari de PoS, és a dir, els n-grames formats únicament per les categories majors.

Features (+Typo[] +Lemma[])						Error CAT	Error SPA
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams		
200	Yes	Yes	Yes	-	-	5.37%	5.32%
200	Yes	Yes	Yes	Yes	-	6.92%	7.1%
200	Yes	Yes	Yes	-	Yes	7.88%	8.07%
200	-	Yes	-	Yes	-	4.45%	-
1000	Yes	Yes	Yes	-	-	4.67%	-
1000	Yes	Yes	Yes	Yes	-	6.47%	-
1000	-	Yes	-	Yes	-	**4.29%**	**5.21%**

TAULA 19: Errors de les proves preliminars de la Tasca 1a (POS), amb PoS[0].

S'observa una clara millora, tots els experiments obtenen errors dos vegades més petits, la millor combinació de trets continua sent la mateixa, però lamentablement els resultats (4,29%) són lleugerament superiors a la *baseline* de referència (4,09%).

10.3.3 RESULTATS (DESAMBIGUADOR POS)

Es conclou que el problema es troba en el fet què el model del classificador està sent entrenat amb la totalitat dels exemples, no només en els casos on existeix ambigüitat, i que probablement els tokens que tenen una única lectura estan interferint en la inducció del model desambiguador. Per tant, s'opta per induir un desambiguador en comptes d'un etiquetador. En una aplicació real, les formes serien consultades al diccionari, en el cas de tenir una única lectura seria la que s'assignaria i en el cas de tenir més d'una lectura s'assignaria l'etiqueta proporcionada pel model estadístic induït.

Així doncs, es selecciona un subcorpus format per aquells tokens ambigus als quals el diccionari adjudica més d'una lectura, i s'utilitza per experimentar la inducció de diferents models, començant per les *baselines* de referència:

Model Subcorpus (+Typo[] +Lemma[])	Error CAT	
	Ambiguous	Overall
ZeroRule: all `[NC]`	82.17%	46.02%
OneRule: *f*(`Lemma[0]`)	52.36%	29.32%
OneRule: *f*(`PoS[0]`)	**7.34%**	**4.11%**

TAULA 20: Errors de les *baselines* de la Tasca 1b (POS), subcorpus ambigus.

Tot i que els errors absoluts semblen pitjors, cal tenir en compte que és l'error validat en el subconjunt de tokens ambigus i que com que aquests representen un 56% del total, els errors globals són aproximadament la meitat, és a dir, un 4,11% en comptes del 4,09%, una *baseline* totalment equivalent.

Finalment es repeteixen els experiments amb les millors combinacions de trets, però entrenant els models amb el subcorpus de tokens ambigus i obtenim uns resultats molt millors. Si apliquem la mateixa correcció per obtenir l'error global a partir de l'error sobre

el corpus d'entrenament, obtenim un error del 2,61%, clarament inferior al 4,09% de la *baseline*, és a dir, obtenim un classificador capaç d'etiquetar correctament el 97,39% de tokens, un taxa d'error habitual en aquestes tasques de PLN **[Martí et al., 2008]**.

Features (+Typo[] +Lemma[])					**Subcorpus**	**Error CAT**	
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams	**Ambiguous**	**Overall**
1000	Yes	Yes	Yes	-	-	6.19%	3.47%
1000	Yes	Yes	Yes	Yes	-	6.19%	3.47%
1000	-	Yes	-	Yes	-	**4.67%**	**2.61%**

TAULA 21: Errors de les proves preliminars de la Tasca 1b (POS), subcorpus ambigus.

A partir d'aquesta combinació òptima de trets es realitza una darrera prova per conèixer la sensibilitat del model a la mida del corpus, és a dir, observar la variació de l'error a mesura que es redueix la quantitat d'exemples d'entrenament. Els resultats de la taula següent mostren com la tasca és clarament sensible a la reducció del nombre d'exemples:

Corpus	**0.1%**	**1%**	**10%**	**70%**	***Baseline***
Ambiguous	46.6%	28.8%	9.31%	4.67%	*7.34%*
Overall	26.1%	16.1%	5.21%	**2.61%**	*4.11%*

TAULA 22: Proves preliminars de *saturació* a la Tasca 1b (POS), subcorpus ambigus.

Així doncs, la tasca final està formada pel subconjunt d'exemples amb dos o més lectures. I la representació que obté els millors resultats està constituïda pels trets generals (`Typo[-3,+3]`, `Lemma[-3,+3]`), la `PoS` del token a etiquetar i la utilització dels n-grames immediats com a substituts de les `PoS` individuals del context dret i esquerra. Segons els resultats obtinguts aquesta tasca queda validada per ser una tasca soluble i no saturada.

<Typo[-3,+3]>	**<defPoS>**	**<Lemma[-3,+3]>** …
[],[],[],[T2],[M9],[X1],[U2],	DA,	[],[],[],el,[*],[PAR],[*],[*], …
[],[T2],[M9],[X1],[U2],[X1],[L2],	FBo,	[],el,[*],[PAR],[*],[PAR],haver, …
[M9],[X1],[U2],[X1],[L2],[L9],[L2],	FBc,	[*],[PAR],[*],[PAR],haver,confirmar,el,…
[X1],[U2],[X1],[L2],[L9],[L2],[L8],	VxIP,	[PAR],[*],[PAR],haver,confirmar,el,[*],…
[X1],[L2],[L9],[L2],[L8],[L1],[L6],	DA,	[PAR],haver,confirmar,el,[*],a,quatre,…
[L2],[L9],[L2],[L8],[L1],[L6],[L4],	NC,	haver,confirmar,el,[*],a,quatre,any,…
[L9],[L2],[L8],[L1],[L6],[L4],[L1a],	SP,	confirmar,el,[*],a,quatre,any,de,…
[L2],[L8],[L1],[L6],[L4],[L1a],[L9],	DN,	el,[*],a,quatre,any,de,[*],…
[L1a],[L9],[L8],[L1],[L3],[L5],[L2],	CC,	de,[*],especial,i,un,[*],de,…
[L9],[L8],[L1],[L3],[L5],[L2],[N3],	DI,	[*],especial,i,un,[*],de,[*],…
[L1],[L3],[L5],[L2],[N3],[L7],[L2],	SP,	i,un,[*],de,[*],milió,de,…
…	…	…

TAULA 23: Fragment de les dades d'entrenament per a la Tasca 1 (POS), meitat esquerra.

	-Bigram,	+Bigran,	-Trigram,	+Trigram,	<POS>
…	[*],	NP-FBo,	[*],	NP-FBo-NP,	DA
…	DA-NP,	NP-FBc,	[*],	[*],	FBo
…	FBo-NP,	VxIP-VmZP,	NP-FBo-NP,	VxIP-VmZP-DA,	FBc
…	NP-FBc,	VmZP-DA,	FBo-NP-FBc,	VmZP-DA-NC,	VxIP
…	VxIP-VmZP,	NC-SP,	[*],	NC-SP-DN,	DA
…	VmZP-DA,	SP-DN,	VxIP-VmZP-DA,	SP-DN-NC,	NC
…	DA-NC,	DN-NC,	VmZP-DA-NC,	DN-NC-SP,	SP
…	NC-SP,	NC-SP,	DA-NC-SP,	NC-SP-NC,	DN
…	NC-AQ,	DI-NC,	SP-NC-AQ,	DI-NC-SP,	CC
…	AQ-CC,	NC-SP,	NC-AQ-CC,	NC-SP-Z,	DI
…	DI-NC,	Z-NC,	CC-DI-NC,	Z-NC-SP,	SP
…	…	…	…	…	…

TAULA 24: Fragment de les dades d'entrenament per a la Tasca 1 (POS), meitat dreta.

10.4 TASCA 2: NAMED ENTITY DETECTION (ENT)

10.4.1 DESCRIPCIÓ

La tasca T2 (ENT) basada en les dades del corpus A2 (ENT) es correspon a una altra tasca habitual de PLN, la detecció d'entitats anomenades multitoken mitjançant l'assignació d'una de tres etiquetes possibles (`[B]`, `[I]` i `[O]`) segons el rol de cada token. El mig milió de tokens de cada idioma es distribueixen segons la taula següent:

Tokens	ENT-Catalan	ENT-Spanish
Tagged	523,000 (100%)	546,000 (100%)
Begin+In	38,600 (7.34%)	29,000 (5.34%)
Begin	13,000 (2.46%)	11,000 (2.01%)
In	25,600 (4.89%)	18,000 (3.30%)
Out	484,000 (92.6%)	517,000 (94.7%)

TAULA 25: Descripció quantitativa del corpus de la Tasca 2 (ENT).

És a dir, pel corpus català i espanyol respectivament, només el 7,34% i 5,34% dels exemples formen part d'entitats anomenades i, per tant, la majoria dels exemples (el 92,6% i el 94,7%) no formen part de cap entitat, ja que només hi ha disponibles 38.600 exemples positius.

Per representar els exemples disposem de la mateixa informació de què disposàvem a la tasca anterior:

- **`Typo[-3,+3]`** Informació tipogràfica de tots els tokens de la finestra.
- **`Lemma [-3,+3]`** Lema per defecte de tots els tokens de la finestra.
- **`PoS[-3,-1]`** Categoria gramatical per defecte del context esquerre.
- **`PoS[0]`** Categoria gramatical per defecte del token central.

- **`PoS[+1,+3]`** Categoria gramatical per defecte del context dret.
- **`+ngrams`** Bigrames i trigrames adjacents als dos costats.
- **`+minigrams`** Igual que l'anterior però només amb categories majors.

10.4.2 RESULTATS

La primera prova també va consistir a obtenir una *baseline* de referència a partir dels errors obtinguts pels algorismes *ZeroRule* i *OneRule*:

Model (Overall)	**Error CAT**	**Error SPA**
ZeroRule: all [O]	7.35%	5.34%
OneRule: *f*(Lemma[0])	**6.63%**	**4.78%**
OneRule: *f*(Lemma[-1])	6.74%	4.94%

TAULA 26: Errors de les *baselines* de la Tasca 2 (ENT).

S'observa com l'existència d'una categoria clarament majoritària (el 92,6% dels tokens són `[Out]`) fa que la simple assignació d'aquesta etiqueta obtingui un error del 7,35%. Valor que es pot reduir lleugerament utilitzant una *baseline* una mica més sofisticada basada en el valor del lema del propi token (6,63%) o del precedent (6,74%).

A partir d'aquesta referència es realitzen diferents experiments variant els trets utilitzats i s'obtenen els resultats següents:

Features (+Typo[] +Lemma[])						**Error CAT**	**Error SPA**
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams		
200	Yes	Yes	Yes	-	-	5.94%	3.55%
200	Yes	Yes	Yes	Yes	-	7.76%	-
200	Yes	Yes	Yes	-	Yes	7.99%	-
200	-	Yes	-	Yes	-	5.76%	-
1000	Yes	Yes	Yes	-	-	5.71%	3.46%
1000	Yes	Yes	Yes	Yes	-	7.77%	-
1000	-	Yes	-	Yes	-	5.73%	-
1000	-	Yes	-	-	-	**4.29%**	-

TAULA 27: Errors de les proves preliminars de la Tasca 2 (ENT).

Sembla que, una vegada proporcionada la informació tipogràfica i de lema, la inclusió de més informació morfosintàctica (`PoS[]`, `n-grams`, `mini-grams`, ...) no millora els resultats. L'error mínim s'obté, doncs, eliminant tot el context morfosintàctic i deixant únicament la PoS del token corresponent. Es repeteixen les proves eliminant precisament aquest tret amb l'objectiu de determinar-ne la importància, i els resultats obtinguts són els següents:

Features (+Typo[] +Lemma[])						Error CAT	Error SPA
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams		
200	Yes	-	Yes	-	-	6.14%	3.89%
200	Yes	-	Yes	Yes	-	7.76%	-
200	Yes	-	Yes	-	Yes	8.39%	5.71%
200	-	-	-	Yes	-	**4.11%**	-
1000	Yes	-	Yes	-	-	5.84%	-
1000	Yes	-	Yes	Yes	-	8.00%	-
1000	-	-	-	Yes	-	5.96%	-

TAULA 28: Errors de les proves preliminars de la Tasca 2 (ENT), sense PoS[0].

Sembla que, en general, els resultats empitjoren lleugerament excepte per a la combinació on el `PoS[0]` ha estat substituït pels n-grames adjacents que el millora amb un 4,11%. És possible que, com en el cas anterior, la redundància de la informació morfosintàctica sigui contraproduent.

La sospita que a mesura que eliminem trets, amb informació potencialment redundant, es milloren els resultats porta a fer més proves en les quals s'ha eliminat la informació del `Lemma[]`. Previsiblement es torna a observar la complementarietat entre els trets `PoS[]` i els n-grams, però de totes maneres s'obtenen pitjors resultats, suggerint que la informació proporcionada pels lemmes sí és important:

Features (+Typo[] –Lemma[])						Error CAT	Error SPA
N	PoS[-3,-1]	PoS[0]	PoS[+1,+3]	n-grams	mini-grams		
200	Yes	Yes	Yes	-	-	6.12%	3.71%
200	Yes	Yes	Yes	Yes	-	8.01%	-
200	Yes	Yes	Yes	-	Yes	8.28%	5.40%
200	-	-	-	Yes	-	-	-
1000	Yes	Yes	Yes	-	-	**6.01%**	-
1000	Yes	Yes	Yes	Yes	-	8.20%	-
1000	-	-	-	Yes	-	**6.02%**	-

TAULA 29: Errors de les proves preliminars de la Tasca 2 (ENT), sense Lemma[].

Finalment, es fa una darrera prova en la qual s'elimina tota la informació morfosintàctica, proporcionant al classificador únicament informació tipogràfica i el lema central. Curiosament, els millors resultats s'aconsegueixen reduint encara més la finestra dels trets tipogràfics:

Features (+Lemma[0])					Error CAT	Error SPA
N	Typo[-3]	Typo[-2,+2]	Typo[+3]	PoS[-3,+3]		
1000	Yes	Yes	Yes	-	3.74%	-
1000	-	Yes	-	-	**3.57%**	2.22%
1000	-	Yes	-	Yes	4.28%	3.15%

TAULA 30: Errors de les proves preliminars de la Tasca 2 (ENT), Lemma[0] i Typo[-2,+2].

Aquest error final del 3,57% és inferior al 7,35% de la *baseline* de referència, i tot i que el corpus presenta una distribució no equilibrada podem considerar que el model ha induït un model prou vàlid i per tant la tasca es pot considerar soluble.

Si amb aquesta combinació òptima de trets es realitza la prova per conèixer la sensibilitat del model a la mida del corpus s'obtenen els següents resultats:

Corpus	**0.1%**	**1%**	**10%**	**70%**	***Baseline***
Error CAT	7.37%	6.29%	6.03%	**5.71%**	*7.35%*

TAULA 31: Proves preliminars de *saturació* a la Tasca 2 (ENT).

I per tant, la tasca final formada per tots els exemples del corpus representats únicament pel `Lemma[0]` i els `Typo[-2,+2]`, és una tasca soluble i no saturada.

<Typo-2>,	**<Typo-1>**	**<Typo>**	**<Typo+1>**	**<Typo+2>**	**<Lemma>**	**<BIO>**
…	…	…	…	…	…	…
[],	[],	[T2],	[L8],	[L2],	el,	O
[],	[T2],	[L8],	[L2],	[L1a],	director,	O
[T2],	[L8],	[L2],	[L1a],	[T6],	de,	O
[L8],	[L2],	[L1a],	[T6],	[T8],	el,	O
[L2],	[L1a],	[T6],	[T8],	[X1],	[*],	B
[L1a],	[T6],	[T8],	[X1],	[T6],	[*],	I
[T6],	[T8],	[X1],	[T6],	[T7],	[COMMA],	O
[T8],	[X1],	[T6],	[T7],	[X1],	lluís,	B
[X1],	[T6],	[T7],	[X1],	[L3],	[*],	I
[T6],	[T7],	[X1],	[L3],	[L6],	[COMMA],	O
[T7],	[X1],	[L3],	[L6],	[L9],	que,	O
…	…	…	…	…	…	…

TAULA 32: Fragment de les dades d'entrenament per a la Tasca 2 (ENT).

10.5 TASCA 3: SPACE NORMALIZATION (SPC)

10.5.1 DESCRIPCIÓ

La tasca T3 (SPC) basada en les dades del corpus B1 (SPC) es correspon a una tasca de pre-processament que, tot i no ser massa habitual, ens permetrà treballar amb un elevat nombre d'exemples: la normalització o inserció d'espais entre tokens, consistent a inserir o no un espai entre dos tokens consecutius. Els corpus corresponents tenen al voltant d'un milió d'exemples i, per tant, la mateixa quantitat de transicions distribuïts segons la taula següent:

Transitions	**SPC-English**	**SPC-German**
Tokens	1,080,000 (100%)	935,000 (100%)
Space Separated	895,000 (82.9%)	759,000 (81.2%)
Joined/Non-Separated	185,000 (17.1%)	176,000 (18.9%)

TAULA 33: Descripció quantitativa del corpus de la Tasca 3 (SPC).

És a dir, tant per al corpus anglès com per a l'alemany, la majoria de transicions impliquen una separació mitjançant espai (el 82,9% i el 81,2% respectivament), i només el 17,1% i el 18,9% de les transicions les formen tokens tipogràficament units.

Aquest corpus no inclou cap mena d'anotació lingüística i, per tant, en aquesta tasca només disposem d'informació tipogràfica i textual, representada mitjançant els trets següents:

- **`Canon[-3,+3]`** Representació normalitzada de tots els tokens de la finestra.
- **`Typo[-3,+3]`** Informació tipogràfica de tots els tokens de la finestra.
- **`+ngrams`** Bigrames i trigrames de les etiquetes tipogràfiques.

10.5.2 RESULTATS

Com en els altres casos, primer es determina una *baseline* de referència a partir dels errors obtinguts pels algorismes *ZeroRule* i *OneRule*:

Model (Overall)	**Error ENG**	**Error GER**
ZeroRule: all [Space]	17.12%	18.94%
OneRule: *f*(Canon[+1])	**2.02%**	**1.85%**

TAULA 34: Errors de les *baselines* de la Tasca 3 (SPC).

L'assignació del cas majoritari, assumir que sempre hi ha espai, proporciona uns errors del 17,12% i del 18,94% en els corpus anglès i alemany. Al permetre que l'algorisme seleccioni un sol tret a partir del qual fer la predicció, obtenim que basant-se en la forma canònica del token posterior, l'error de classificació baixa fins al 2,02% i l'1,85%, en principi, una *baseline* prou baixa.

A partir d'aquesta referència es realitzen diferents experiments variant els trets utilitzats. S'observa que la reducció de la mida de la finestra millora els resultats i, per tant, es proven altres variants reduint la mida del context, els resultats són els següents:

Features					**Error ENG**	**Error GER**
N	Text[]	Canon[]	Typo[]	n-grams		
1000	-	[-3,+3]	[-3,+3]	Yes	1.69%	1.37%
1000	-	[-3,+3]	-	Yes	1.62%	1.42%
1000	-	[-3,+3]	[-3,+3]	-	1.14%	1.47%
1000	-	[-2,+1]	[-2,+1]	-	0.53%	1.00%
1000	-	[-1,+1]	[-1,+1]	-	**0.47%**	0.58%
1000	-	[-1,+1]	[-1,+1]	Yes	1.27%	-
1000	-	-	[-1,+1]	Yes	2.30%	-
1000	-	[-1,+1]	-	Yes	0.96%	-
1000	-	-	[-1,+1]	-	2.33%	-
1000	-	[-1,+1]	-	-	0.57%	0.56%

TAULA 35: Errors de les proves preliminars de la Tasca 3 (SPC).

Sembla que la millor combinació de trets és la que utilitza únicament la informació tipogràfica i la forma canònica dels dos tokens implicats en la transició, sent aconsellable ignorar la resta del context. Aquest model obté un error final del 0,47% que és clarament inferior al 2,05% de la *baseline* de referència, i tot i que la tasca no sembla massa complexa queda clar que el model resol la tasca molt correctament i la tasca és soluble.

Seguidament es realitza la prova preliminar reduint la mida del corpus d'entrenament i s'observa que l'error és sensible a aquest canvi:

Corpus	**0.1%**	**1%**	**10%**	**70%**	***Baseline***
Error ENG	1.74%	0.91%	0.57%	**0.51%**	*2.05%*

TAULA 36: Proves preliminars de *saturació* a la Tasca3 (SPC).

Per tant, la tasca final formada per tots els exemples del corpus representats únicament pel `Canon[-1,+1]` i els `Typo[-1,+1]`, és una tasca soluble i no saturada.

<Typo-1>,	<Typo+1>	<Canon-1>	<Canon+1>	<Space>
…	…	…	…	…
[L8],	[L5],	military,	court,	True
[L5],	[L2],	court,	in,	True
[L2],	[T4],	in,	[*],	True
[T4],	[X1],	[*],	[PERIOD],	False
[X1],	[],	[PERIOD],	[],	False
[T3],	[L9],	the,	[*],	True
[L9],	[L5],	[*],	[*],	True
[L5],	[X1],	[*],	[COMMA],	False
[X1],	[T2d],	[COMMA],	mr.,	True
[T2d],	[T9],	mr.,	[*],	True
[T9],	[X1],	[*],	[COMMA],	False
…	…	…	…	…

TAULA 37: Fragment de les dades d'entrenament per a la Tasca 3 (SPC).

10.6 TASCA 4: PERIOD DISAMBIGUATION (DOT)

10.6.1 DESCRIPCIÓ

La tasca T4 (DOT) basada en les dades del corpus B2 (DOT) es correspon a una tasca clàssica de pre-processament: la desambiguació dels punts al final d'oració. Com és sabut, tot i que els punts acostumen a indicar el final d'oració, també apareixen a l'interior de determinats tokens o després de les abreviatures, així doncs, la idea d'aquesta tasca consistia a entrenar un classificador per desambiguar els punts entre final d'oració o interns.

Transitions	DOT-English	DOT-German
Sentences	50,271	54,254
Total Dots	55,339 (100%)	58,403 (100%)
Middle Dots	5,030 (9.09%)	4,114 (7.04%)
EndOfSentence Dots	50,309(90.9%)	54,289 (93.0%)

TAULA 38: Descripció quantitativa del corpus de la Tasca 4 (DOT).

Els corpus corresponent està format per un total aproximat de 55.000 punts, segons la distribució de la taula anterior. La majoria dels punts (90,9% i 93,0% respectivament) són punts finals d'oració i només un 9,09% i un 7,04% dels punts són interns.

La representació mitjançant trets tipogràfics i textuals és la mateixa de la tasca anterior:

- **`Canon[-3,+3]`** Representació normalitzada de tots els tokens de la finestra.
- **`Typo[-3,+3]`** Informació tipogràfica de tots els tokens de la finestra.
- **`+ngrams`** Bigrames i trigrames de les etiquetes tipogràfiques .

10.6.2 RESULTATS

Com s'ha fet a les tasques anteriors, es determina una *baseline* de referència a partir dels errors obtinguts pels algorismes *ZeroRule* (9,33% i 7,17%), *OneRule* basant-se en la informació tipogràfica del token immediatament anterior al punt (1,51% i 0,87%), i *OneRule* basant-se en el trigrama tipogràfic anterior al punt (0,98% i 0,56%):

Model (Overall)	**Error ENG**	**Error GER**
ZeroRule: all [EoS]	9.33%	7.17%
OneRule: *f*(Typo-1)	1.51%	0.87%
OneRule: *f*(-Trigram)	**0.98%**	**0.56%**

TAULA 39: Errors de les *baselines* de la Tasca 4 (DOT).

A continuació es realitzen diferents experiments per provar diferents combinacions de trets:

Features					Error ENG	Error GER
N	Text[]	Canon[]	Typo[]	n-grams		
1000	-	[-3,+3]	[-3,+3]	Yes	0.67%	**0.52%**
1000	-	[-3,+3]	[-3,+3]	-	**0.64%**	0.75%
1000	-	-	-	Yes	0.72%	0.57%
1000	-	-	[-3,+3]	-	0.73%	0.79%
1000	-	[-3,+3]	-	-	0.83%	1.06%
1000	-	[-2,+2]	[-2,+2]	-	0.70%	0.68%

TAULA 40: Errors de les proves preliminars de la Tasca 4 (DOT).

Els resultats presenten poca variabilitat en funció dels trets seleccionats i, fins i tot els millors casos, presenten un errors relativament similars a la *baseline* de referència. Es determina que tot i que el corpus estava format per més d'un milió de tokens, la reduïda quantitat d'exemples (55.000 punts) i la desequilibrada proporció entre les dues categories (1:10) fan que el nombre d'exemples de la categoria minoritària (5.000 punts interns) sigui insuficient. Cal recordar que l'objectiu és validar l'eficiència de determinades tècniques d'entrenament, i que això només té sentit quan l'anotació de tot el corpus no és viable.

Per tant la Tasca 4 queda descartada, però es decideix superar els seus inconvenients mitjançant la definició d'una tasca similar i la creació sintètica d'un conjunt de dades artificials d'una mida molt superior.

10.7 TASCA 5: SENTENCE BOUNDARY DETECTION (BRK)

10.7.1 DESCRIPCIÓ

Una vegada descartada la tasca anterior es decideix preparar a partir del corpus B3 (BRK) una tasca similar, la T5 (BRK), de detecció de finals d'oració però que permeti crear un conjunt de dades amb una gran quantitat d'exemples. Aquesta tasca es planteja com un classificador de transicions entre tokens, és a dir, per a cada transició entre dos tokens consecutius, el classificador ha de determinar si es tracta d'una transició entre dues oracions diferents o entre dos tokens de la mateixa oració.

La millora respecte la tasca anterior consisteix a utilitzar les 50.000 oracions disponibles per recombinar-les aleatòriament i generar un corpus 20 vegades més gran. Cal tenir en compte que tot i que les diferents oracions apareixen repetides al corpus final, el fet de no aparèixer en el mateix ordre, garanteix la creació d'un elevat nombre de transicions diferents d'oració i, per tant, un nombre molt elevat de contextos diferents a esquerra i dreta de la transició d'oració.

La generació d'aquest corpus sintètic parteix del corpus B3 format per notícies reals, amb la següent distribució de transicions:

Corpus Original	BRK-English	BRK-German
Total Transitions	2,031,000 (100%)	1,752,000 (100%)
Token Transitions	1,980,000 (97%)	1,698,000 (97%)
EndOfSentence Transitions	50,300 (3%)	54,300 (3%)

TAULA 41: Descripció quantitativa del corpus original de la Tasca 5 (BRK).

Bàsicament es reordenen aleatòriament les oracions i es genera un nou corpus que tot i contenir les mateixes oracions, proporciona 50.000 noves transicions entre oracions. La repetició d'aquest procés 20 vegades dóna lloc a un corpus ampliat amb les següents característiques:

Corpus Sintètic Ampliat	BRK-English	BRK-German
Total Transitions	40,620,000 (100%)	35,040,000 (100%)
Token Transitions	39,600,000 (97%)	33,960,000 (97%)
EndOfSentence Transitions	1,006,000 (3%)	1,086,000 (3%)

TAULA 42: Descripció quantitativa del corpus ampliat de la Tasca 5 (BRK).

Finalment, per reduir la desproporció entre exemples positius i negatius (3%/97%) que perjudica els resultats, es remostreja el corpus per obtenir una proporció més raonable (20%/80%). Concretament es conserven totes les transicions d'oracions i es seleccionen un 10% de les transicions entre tokens. Les característiques quantitatives del corpus final per aquesta tasca són:

Corpus Sintètic Equilibrat	BRK-English	BRK-German
Total Transitions	4,966,000 (100%)	4,482,000 (100%)
Token Transitions	3,960,000 (80%)	3,396,000 (76%)
EndOfSentence Transitions	1,006,000 (20%)	1,086,000 (24%)

TAULA 43: Descripció quantitativa del corpus re-equilibrat de la Tasca 5 (BRK).

Cal dir que per poder realitzar amb el *Weka* les proves preliminars d'aquesta tasca, s'ha hagut d'utilitzar una versió reduïda (1:8) del conjunt de dades, ja que el programari presenta inestabilitats al treballar amb conjunts d'entrenament massa grans. De totes maneres, experiments posteriors fets amb la totalitat de les dades han confirmat la validesa dels resultats obtinguts en aquestes proves preliminars.

La representació mitjançant trets tipogràfics i textuals és la mateixa que per a la resta de tasques del corpus B:

- **`Canon[-3,+3]`** Representació normalitzada de tots els tokens de la finestra.
- **`Typo[-3,+3]`** Informació tipogràfica de tots els tokens de la finestra.
- **`+ngrams`** Bigrames i trigrames de les etiquetes tipogràfiques.

10.7.2 RESULTATS

Les *baselines* de referència obtinguts a partir de l'algorisme *ZeroRule*, que assigna sempre la categoria més freqüent, són pitjors en el corpus re-equilibrat, ja que a l'augmentar la proporció de la categoria minoritària, augmenten els errors corresponents. Però si s'utilitza una *baseline* lleugerament més sofisticada, com la *OneRule* basant-se en la forma canònica del token precedent, s'observa com el corpus re-equilibrat permet entrenar millor el model classificador:

Model	Error ENG		Error GER	
	Original	Balanced	Original	Balanced
ZeroRule: all [Typo]	2.451%	20.51%	3.272%	24.12%
OneRule: *f*(Canon-1)	0.283%	**0.231%**	0.270%	**0.204%**

TAULA 44: Errors de les *baselines* de la Tasca 5 (BRK), original i re-equilibrat.

A continuació es mostren els resultats dels diferents experiments, tant amb el corpus original com amb el re-equilibrat, realitzats per seleccionar els trets utilitzats en la tasca final:

Features					Error ENG		Error GER	
N	Text[]	Canon[]	Typo[]	n-grams	Original	Balanced	Original	Balanced
1000	-	[-3,+3]	[-3,+3]	Yes	0.055%	**0.028%**	0.069%	**0.064%**
1000	-	[-3,+3]	[-3,+3]	-	0.142%	0.042%	0.192%	0.210%
1000	-	-	-	Yes	2.595%	2.150%	0.211%	2.434%
1000	-	-	[-3,+3]	-	0.144%	0.198%	0.070%	0.262%
1000	-	[-3,+3]	-	-	0.127%	0.097%	0.128%	0.098%
1000	-	[-2,+2]	[-2,+2]	-	0.078%	0.063%	0.184%	0.096%

TAULA 45: Errors de les proves preliminars de la Tasca 5 (BRK), original i re-equilibrat.

En aquest cas la utilització de tota la informació disponible és l'opció que proporciona millors resultats. Aquest model obté un error final del 0,028% que és clarament inferior al 0,231% de la *baseline* de referència, i per tant la tasca és soluble.

També es va realitzar una prova preliminar per determinar la sensibilitat a la mida del corpus, que va donar aquests resultats:

Corpus	0.1%	1%	10%	70%	*Baseline*
Error ENG	0.521%	0.212%	0.65%	**0.029%**	*0.231%*

TAULA 46: Proves preliminars de *saturació* a la Tasca5 (BRK), re-equilibrat.

Per tant, es conclou que la tasca final formada per tots els exemples del corpus sintètic re-equilibrat, representats per la totalitat dels trets `Canon[-3,+3]`, els `Typo[-3,+3]`, i els bigrames i trigrames anteriors i posteriors, és una tasca soluble i no saturada.

<Typo[-3,+3]>	**<Lemma[-3,+3]>** …
[SP],[L8],[SP],[L4],[SP],[L3],[SP],	[SP],[*],[SP],from,[SP],the,[SP], …
[L8],[SP],[L4],[SP],[L3],[SP],[U3],	[*],[SP],from,[SP],the,[SP],fdp, …
[U3],[aL1],[SP],[L6],[SP],[L2],[SP],	fdp,s,[SP],demand,[SP],to,[SP], …
[SP],[L2],[SP],[D4],[X1],[SP],[SP],	[SP],as,[SP],[*],[PERIOD],[SP],[SP], …
[L2],[SP],[D4],[X1],[SP],[SP],[SP],	as,[SP],[*],[PERIOD],[SP],[SP],[SP], …
[SP],[T3],[SP],[U3],[X1],[U3],[SP],	[SP],the,[SP],cdu,[SLASH],csu,[SP], …
[X1],[U3],[SP],[L3],[SP],[L9],[SP],	[SLASH],csu,[SP],has,[SP],[*],[SP], …
[L6],[SP],[L3],[SP],[L6],[SP],[L9],	reform,[SP],not,[SP],become,[SP],[*], …
…	…

TAULA 47: Fragment de les dades d'entrenament per a la Tasca 5 (BRK), meitat esquerra.

	-Bigram,	+Bigran,	-Trigram,	+Trigram,	<BRK>
…	[L]-[SP],	[SP]-[L3],	[SP]-[L]-[SP],	[SP]-[L3]-[SP],	False
…	[SP]-[L],	[L3]-[SP],	[L]-[SP]-[L],	[L3]-[SP]-[U3],	False
…	[aL1]-[SP],	[SP]-[L2],	[U3]-[aL1]-[SP],	[SP]-[L2]-[SP],	False
…	[L2]-[SP],	[X1]-[SP],	[SP]-[L2]-[SP],	[*],	False
…	[SP]-[D],	[*],	[L2]-[SP]-[D],	[*],	True
…	[T3]-[SP],	[X1]-[U3],	[*],	[X1]-[U3]-[SP],	False
…	[U3]-[SP],	[SP]-[L],	[X1]-[U3]-[SP],	[SP]-[L]-[SP],	False
…	[SP]-[L3],	[L]-[SP],	[L]-[SP]-[L3],	[L]-[SP]-[L],	False
…	…	…	…	…	…

TAULA 48: Fragment de les dades d'entrenament per a la Tasca51 (BRK), meitat dreta.

10.8 CONCLUSIONS

En aquest capítol s'han presentat els resultats de les proves preliminars fetes als conjunts de dades d'entrenament. Aquestes proves tenien com a objectiu determinar si les dades eren prou representatives com per a poder generalitzar les conclusions obtingudes de les avaluacions de les diferents tècniques d'entrenament.

Després d'un llarg procés s'han seleccionat quatre tasques de PLN amb diferents característiques, diferent grau de dificultat, i diferents quantitats d'exemples. Aquesta varietat queda evidenciada en la taula següent que resumeix les característiques de les tasques seleccionades:

#	Tasca	Descripció	Etiquetes	Trets	Exemples	Error
T1	**POS**	Part of Speech Tagging	88	19	278,000	2.61%
T2	**ENT**	Named Entity Detection	3	6	523,000	3.57%
T3	**SPC**	Space Normalization	2	4	1,080,000	0.47%
T5	**BRK**	Sentence Boundary Detection	2	18	4,966,000	0.028%

TAULA 49: Característiques preliminars de les tasques seleccionades: POS, ENT, SPC i BRK.

Les diferents proves exploratòries han permès definir per a cada una el subconjunt de trets més adequat per entrenar un classificador *Naïve Bayes*, i determinar que totes quatre són tasques solubles i no saturades, de manera que poden utilitzar-se perfectament per a avaluar els avantatges de les diferents tècniques d'entrenament incremental.

En els següents capítols es mostraran els resultats de diferents experiments a l'hora d'entrenar un conjunt de classificadors *Naïve Bayes* mitjançant diferents estratègies d'entrenament incremental.

11 ENTRENAMENT INCREMENTAL

11.1 INTRODUCCIÓ

Per a poder comparar les diferents tècniques d'entrenament incremental i verificar si permeten assolir els mateixos objectius de manera més eficient, és a dir, si indueixen models igual de precisos però utilitzant menys exemples, és necessari comparar l'evolució dels models prenent com a referència l'entrenament incremental estàndard.

En aquest capítol es descriuen les corbes d'avaluació, eina bàsica per entendre l'evolució d'un model al llarg del seu entrenament, i les fases típiques que presenten: transició, entrenament i saturació. També es presenta amb més detall el concepte de *saturació* i es proposa un índex que permet quantificar i comparar el grau de dificultat d'una tasca, determinat pel seu univers de representació, en relació al nombre d'exemples d'entrenament. Finalment es mostren les corbes d'avaluació de referència, amb entrenament incremental estàndard, de les quatre tasques seleccionades en el capítol anterior. Aquestes corbes aporten informació sobre la dificultat de la tasca, però sobretot ens defineixen les línies de referència respecte a les que haurem de comparar les altres tècniques d'entrenament incremental.

11.2 CORBES D'AVALUACIÓ

Un classificador incremental és un model classificador que ha estat induït incrementalment a partir de la presentació seqüencial d'exemples coneguda com a *entrenament*. El fet que el model sigui funcional en qualsevol moment de l'entrenament permet avaluar el classificador i observar com evoluciona al llarg del procés. D'aquesta manera podem avaluar-lo periòdicament i obtenir una seqüència de mètriques que, suposadament milloraran a mesura que avanci l'entrenament, és a dir, que se li mostrin més exemples.

Per això, a diferència dels classificadors *batch* que poden sintetitzar el seu comportament amb els valors escalars d'unes mètriques determinades (error, precisió, cobertura, ...), per visualitzar l'evolució d'un classificador incremental és preferible representar aquesta seqüència de valors **[Ciravegna et al., 2002b; Siefkes, 2005; Culotta et al., 2006]** en un gràfic que anomenarem *corbes d'avaluació*.

Tot i que aquestes corbes poden fer-se amb qualsevol de les mètriques que existeixen, si hem de resumir-ne qualitat i triar una única corba que reflecteixi la qualitat global del classificador, el més habitual és fer-ho amb l'exactitud del classificador, definit com la proporció de classificacions correctes respecte el total d'exemples classificats.

Al gràfic següent es mostra un exemple genèric de la corba d'avaluació d'un classificador fictici, l'eix vertical representa el tant per cert d'encerts i l'eix horitzontal, en escala logarítmica, la quantitat d'exemples amb què s'ha entrenat el classificador. El motiu per treballar amb un gràfic semilogarítmic és que la velocitat a la qual convergeix el model no és constant, per la llei de rendiments decreixents, i cada vegada necessita més exemples per continuar millorant:

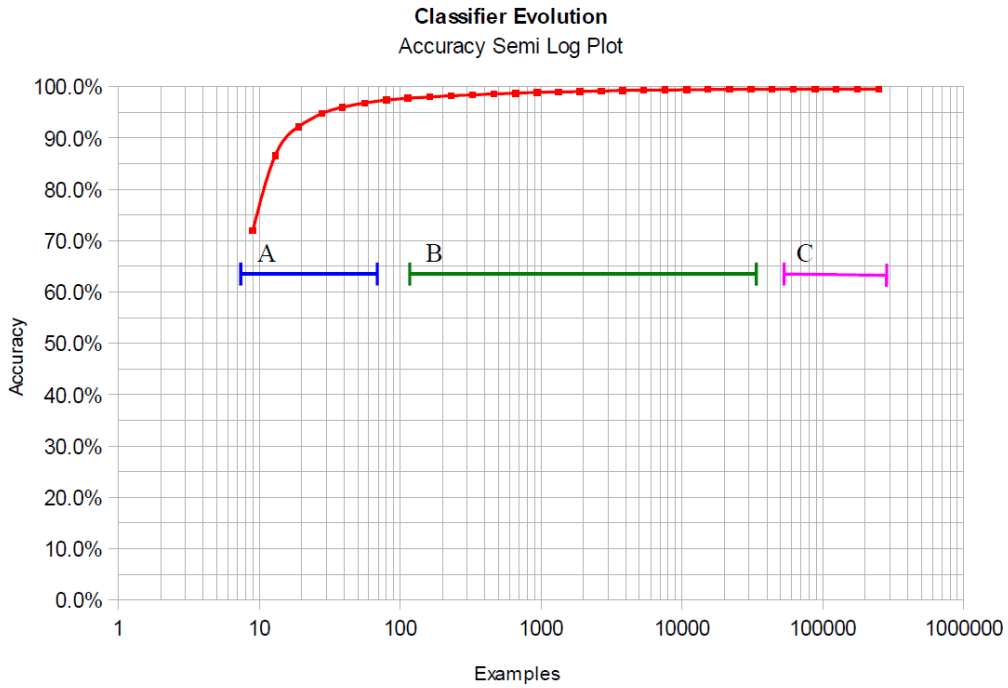

FIG. 78: Evolució de l'exactitud d'un classificador en representació semilogarítmica.

Aquest exemple **[FIG. 78]** mostra les tres etapes habituals de l'entrenament d'un classificador. En primer lloc una etapa (**A**) que podem anomenar *transicional*, on el model millora molt ràpidament a mesura que processa els primers exemples. Aquesta etapa no és representativa ja que els models, especialment els estadístics, mostren comportaments inestables fins que no han pogut analitzar un mínim d'exemples. En segon lloc s'inicia una etapa (**B**) que és l'*entrenament* pròpiament dit en què el sistema millora progressivament de forma logarítmica, és a dir, necessitant cada vegada més exemples per obtenir una millora equivalent; la durada d'aquesta etapa depèn de la dificultat intrínseca de la tasca, especialment de la complexitat de la representació. I finalment, una tercera etapa (**C**) que podem anomenar de *saturació*, en què el model s'ha estabilitzat i un augment en el nombre d'exemples no proporciona millores significatives.

En els casos en què el model classifica correctament la gran majoria dels exemples, a les corbes d'avaluació, és preferible substituir l'exactitud per l'error, és a dir, representar la proporció d'exemples classificats incorrectament, especialment quan es tracta de comparar corbes similars. El gràfic següent **[FIG. 79]** mostra la mateixa evolució però representat el tant per cent d'errors obtinguts, evidentment en aquest cas la corba és decreixent.

El següent pas, recomanable quan l'error evoluciona a través de valors de diversos ordres de magnitud, és transformar l'eix vertical també a una escala logarítmica **[FIG. 80]**, on cada divisió principal es correspon amb un error deu vegades més petit. Aquesta representació és la que s'utilitzarà en aquest capítol i tots els següents.

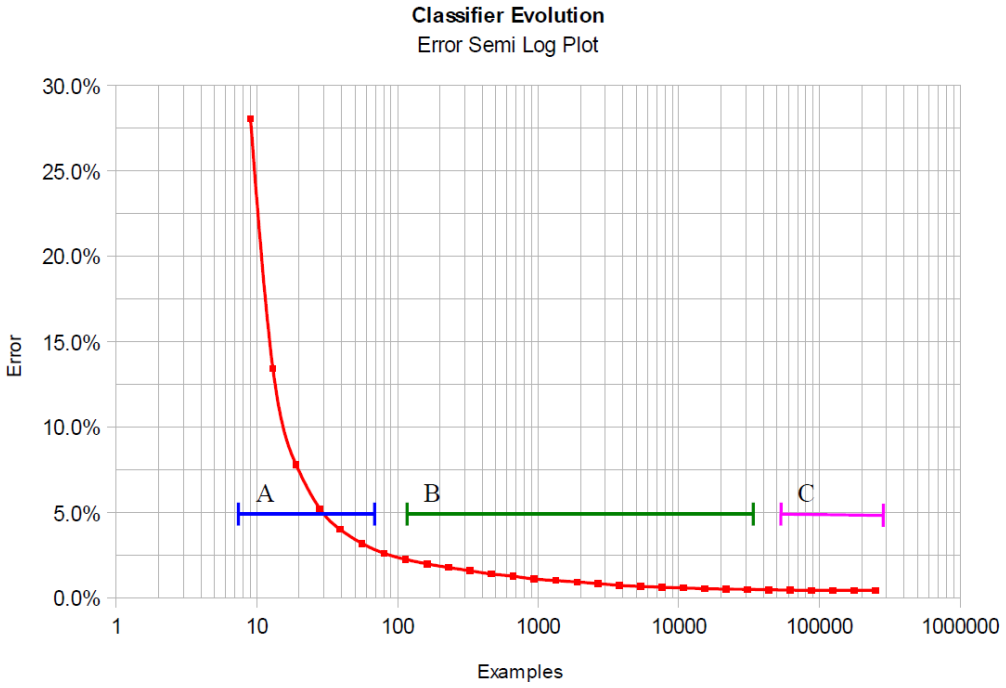

FIG. 79: Evolució de l'error d'un classificador en representació semilogarítmica.

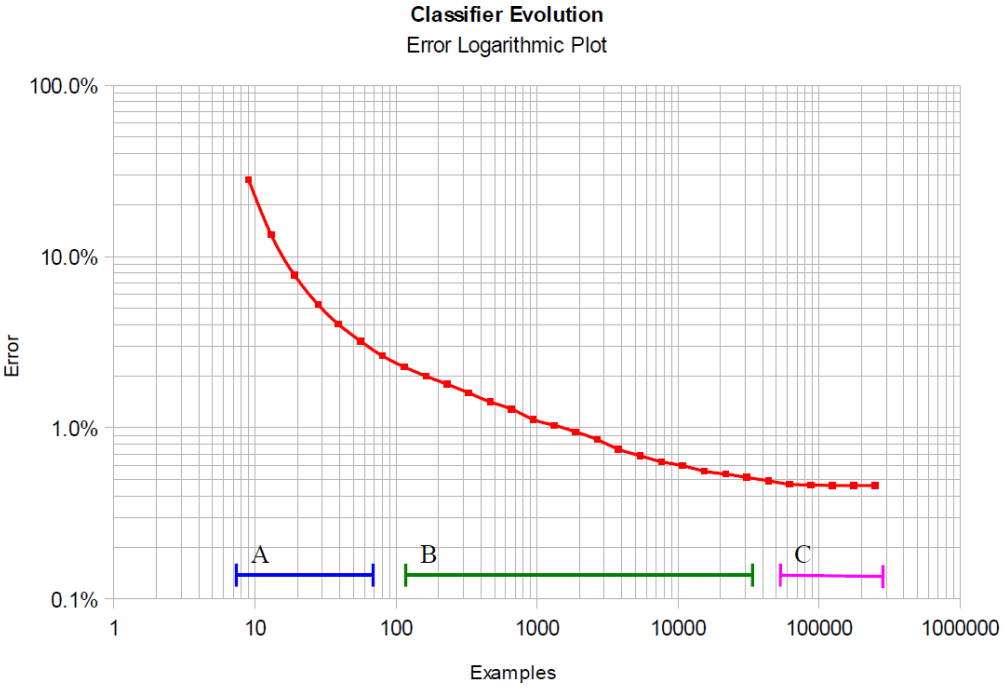

FIG. 80: Evolució de l'error d'un classificador en representació logarítmica.

11.3 DENSITAT I SATURACIÓ

Com s'ha vist anteriorment, la inducció d'un model paramètric acostuma a proporcionar una corba d'error asimptòtica, on a partir d'una determinada quantitat d'exemples el classificador no millora significativament. La causa d'aquest comportament és múltiple, però hi ha dos motius clars: per un costat qualsevol aproximació estadística segueix la llei dels grans nombres i els estimadors convergeixen a mesura que augmenta la mida de la mostra, i per un altre costat, els corpus són redundants amb una distribució que segueix la llei de Zipf i, per tant, la proporció d'exemples informatius també decreix a mesura que s'analitzen més exemples.

En qualsevol cas aquest fenomen succeeix i és important tenir-lo en compte a l'hora d'entrenar un model de forma eficient. Per això és interessant mirar d'entendre una mica, ni que sigui amb un model intuïtiu, quins paràmetres condicionen la *saturabilitat* d'un model.

Una manera és entendre que un model paramètric és una funció amb un valor d'entrada pertanyent a un espai multidimensional i amb un valor de sortida categòric o nominal. La mida d'aquest espai multidimensional, format per l'univers d'exemples possibles, depèn directament de la representació utilitzada, com més trets utilitzi i més valors puguin prendre aquests trets, més gran serà l'univers i més varietat d'exemples podrem representar.

$$|Univers| = f\left(\prod_{i=0}^{n}|feature_i|\right) \cong \prod_{i=0}^{n}|feature_i|$$

És a dir, podem aproximar la mida d'aquest univers, almenys en el cas de trets nominals, com una funció del producte dels valors possibles per cada un dels trets utilitzats.

A més, sembla desitjable que el corpus d'entrenament estigui format per exemples representatius i idealment haurien de distribuir-se uniformement. També seria desitjable que els exemples es repartissin equitativament entre les diferents categories entre les quals el classificador ha de discriminar. A partir d'aquests criteris podem definir un índex que podem anomenar *índex de densitat*, segons la fórmula següent:

$$índex\ de\ densitat \cong \log\left(\frac{|exemples|}{|Univers| \cdot |categories|}\right)$$

Més enllà del valor exacte, aquest índex quantifica la proporció entre la mida del corpus i la complexitat de la representació. La hipòtesi és que si aquesta proporció és baixa, l'entrenament no proporcionarà prou dades com per induir un model final; si la proporció és elevada, l'entrenament proporcionarà prou dades i obtindrà una aproximació raonable al model final; però si la proporció és excessiva, l'entrenament arribarà a la zona de saturació i no obtindrà cap millora.

Aquest indicador és un simple formalisme que intenta capturar les intuïcions que, independentment de la dificultat intrínseca de la tasca,

a) augmentar la quantitat d'exemples, facilita l'entrenament i millora el classificador obtingut;
b) augmentar la complexitat de la representació, afegint més trets o trets més rics, dificulta l'entrenament i empitjora els resultats;
c) augmentar la granularitat de l'etiquetari, també dificulta l'entrenament i empitjora els resultats obtinguts.

Per tant, aquest índex de densitat, hauria d'estar correlacionat amb el risc que un entrenament determinat, amb una tasca i un corpus específic, arribi a una fase de saturació o minimitzi l'error obtingut.

Per a entendre millor el seu funcionament podem aplicar aquestes fórmules a les tasques seleccionades anteriorment. A partir del subconjunt final de trets es pot determinar la mida de l'univers de representació dels exemples, que juntament amb el nombre d'etiquetes i la mida de les dades d'entrenament, permeten obtenir la seva densitat i l'índex corresponent com es mostra a la taula següent:

#	Tasca	Exemples	Etiquetes	Nº Trets	|Univers|	Densitat	$I_{densitat}$
T1	**POS**	278,000	88	19	$\approx 8\times10^{48}$	$\approx 4\times10^{-46}$	-45
T2	**ENT**	523,000	3	6	$\approx 1\times10^{13}$	$\approx 2\times10^{-11}$	-11
T3	**SPC**	1,080,000	2	4	$\approx 1\times10^{10}$	$\approx 5\times10^{-08}$	-7
T5	**BRK**	4,966,000	2	18	$\approx 1\times10^{46}$	$\approx 2\times10^{-40}$	-40

TAULA 50: Univers, densitat i índex de densitat per les tasques de referència.

Interpretant els ordres de magnitud d'aquest indicador, podem dir que la tasca T1(POS), amb la proporció més baixa entre la mida del corpus i la complexitat de la representació, és la que presenta menys risc de saturació, seguida per la tasca T5(BRK) amb una proporció similar. Per contra la tasca T2(ENT) presenta una proporció més alta, i encara més la tasca T3(SPC), que és la que presenta la proporció més alta de totes. Per tant, T2 i T3 serien les tasques en que hi hauria més risc que el model entri en zona de saturació.

11.4 EVOLUCIONS DEL MODELS

Més enllà de les especulacions teòriques, el que és fonamental és observar i analitzar les dades reals. En aquesta secció s'analitza l'evolució de quatre classificadors *Naïve Bayes* entrenats incrementalment. Les dades es corresponen als valors mitjans de deu experiments en què s'ha utilitzat el 90% dels exemples per induir al classificador i el 10% restant per mesurar l'error de classificació. En els següents apartats es mostren les corbes d'avaluació, error mitjà i valors màxims i mínims, i es comenten les principals conclusions que es poden extreure d'aquestes representacions gràfiques.

EVOLUCIÓ DE T1 (POS)

L'evolució d'aquest classificador [**Fig. 81**] mostra com en un primer moment l'error és superior al 50%, ja que es tracta d'una classificació multietiqueta i amb una representació molt complexa. Però l'error es redueix progressivament, de manera pràcticament lineal (lineal en escala logarítmica) fins arribar a poc més del 4% al final de l'entrenament. També s'observa com la variació entre experiments és bastant baixa, i les corbes del pitjor i millor cas evolucionen properes i paral·leles a l'error mitjà.

La característica més destacada que es pot observar és que al finalitzar l'entrenament el model es troba molt lluny d'haver saturat, el pendent de la corba encara és pronunciada, i sembla evident que en cas d'haver continuat l'entrenament amb més exemples l'error s'hauria pogut reduir sensiblement.

[Espai intencionadament en blanc per alinear el text amb les figures.]

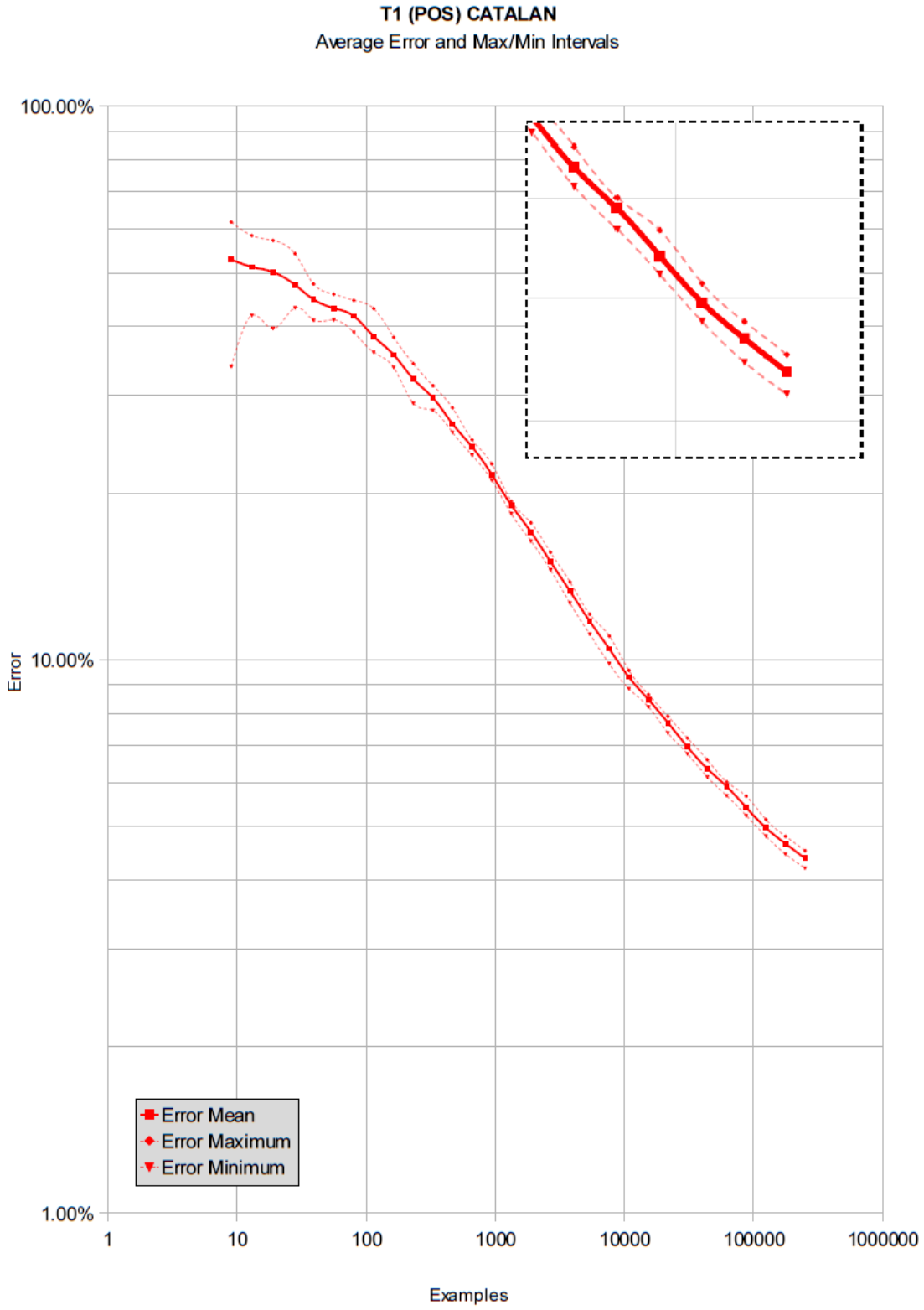

FIG. 81: Evolució de l'error de T1(POS) a un entrenament incremental estàndard.

EVOLUCIÓ DE T2 (ENT)

En aquest cas el gràfic corresponent **[Fig. 82]** mostra com l'error inicial és inferior al 7%, cal recordar que existeix una etiqueta majoritària `[O]` pel 93% dels tokens, però que la reducció de l'error és molt més lenta, fins arribar al voltant del 3,5% al final de l'entrenament. A diferència del cas anterior, el pendent al final de l'entrenament és pràcticament pla i queda clar que un augment en el nombre d'exemples no sembla que pogués reduir l'error, el model ha arribar clarament a un punt de saturació amb una reducció de l'error molt pobre.

[Espai intencionadament en blanc per alinear el text amb les figures.]

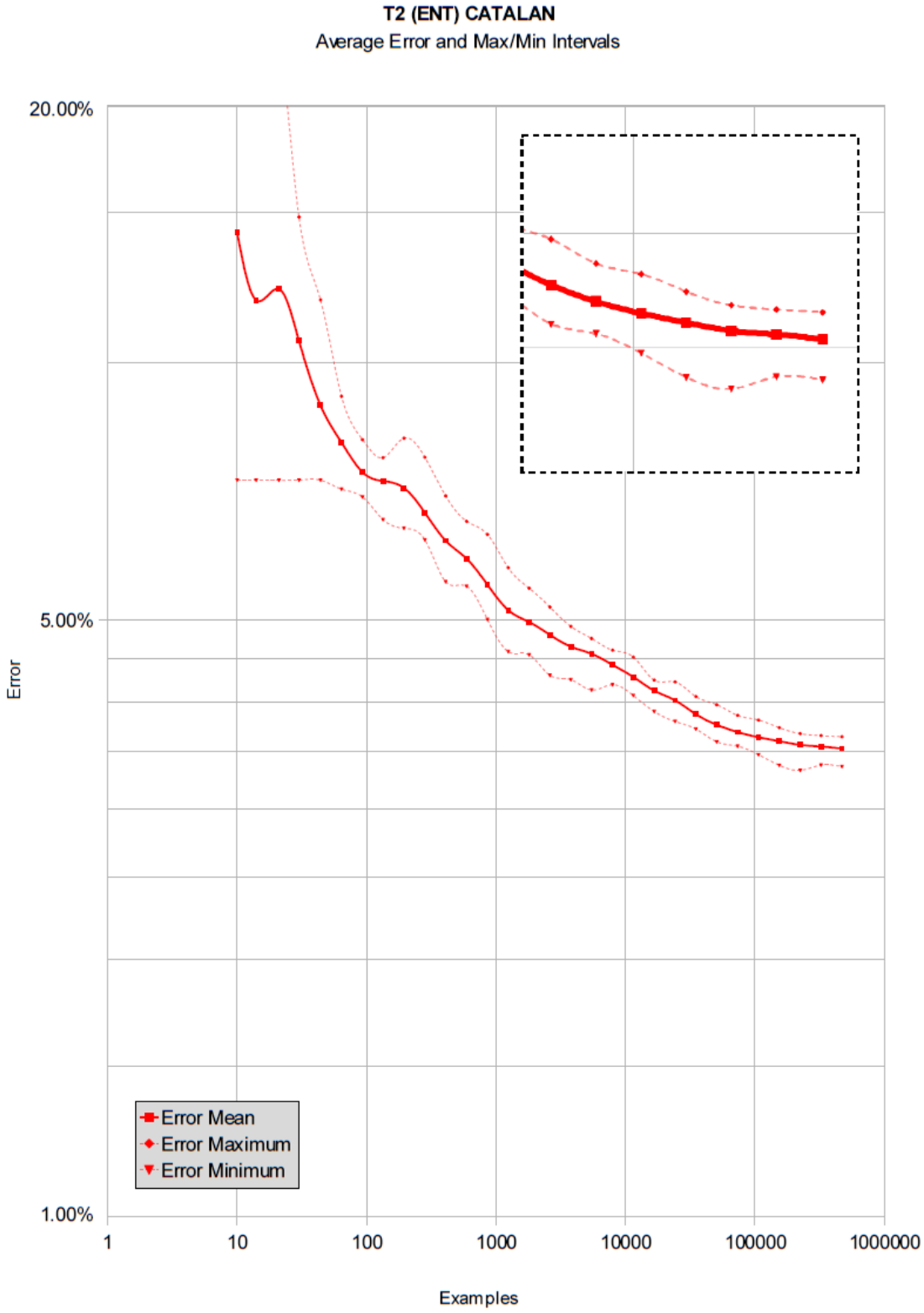

FIG. 82: Evolució de l'error de T2(ENT) a un entrenament incremental estàndard.

EVOLUCIÓ DE T3 (SPC)

La corba d'aquesta tasca **[Fig. 83]** comença amb un error proper al 20% que es redueix molt ràpidament fins a un valor inferior al 0,5% en finalitzar l'entrenament. En aquest cas el pendent de la corba també suggereix indicis de saturació i no sembla que proporcionar-li més exemples aconseguís reduir l'error més d'unes dècimes. En aquest cas el model sí ha après a resoldre la tasca raonablement bé, però ha assolit la saturació.

[Espai intencionadament en blanc per alinear el text amb les figures.]

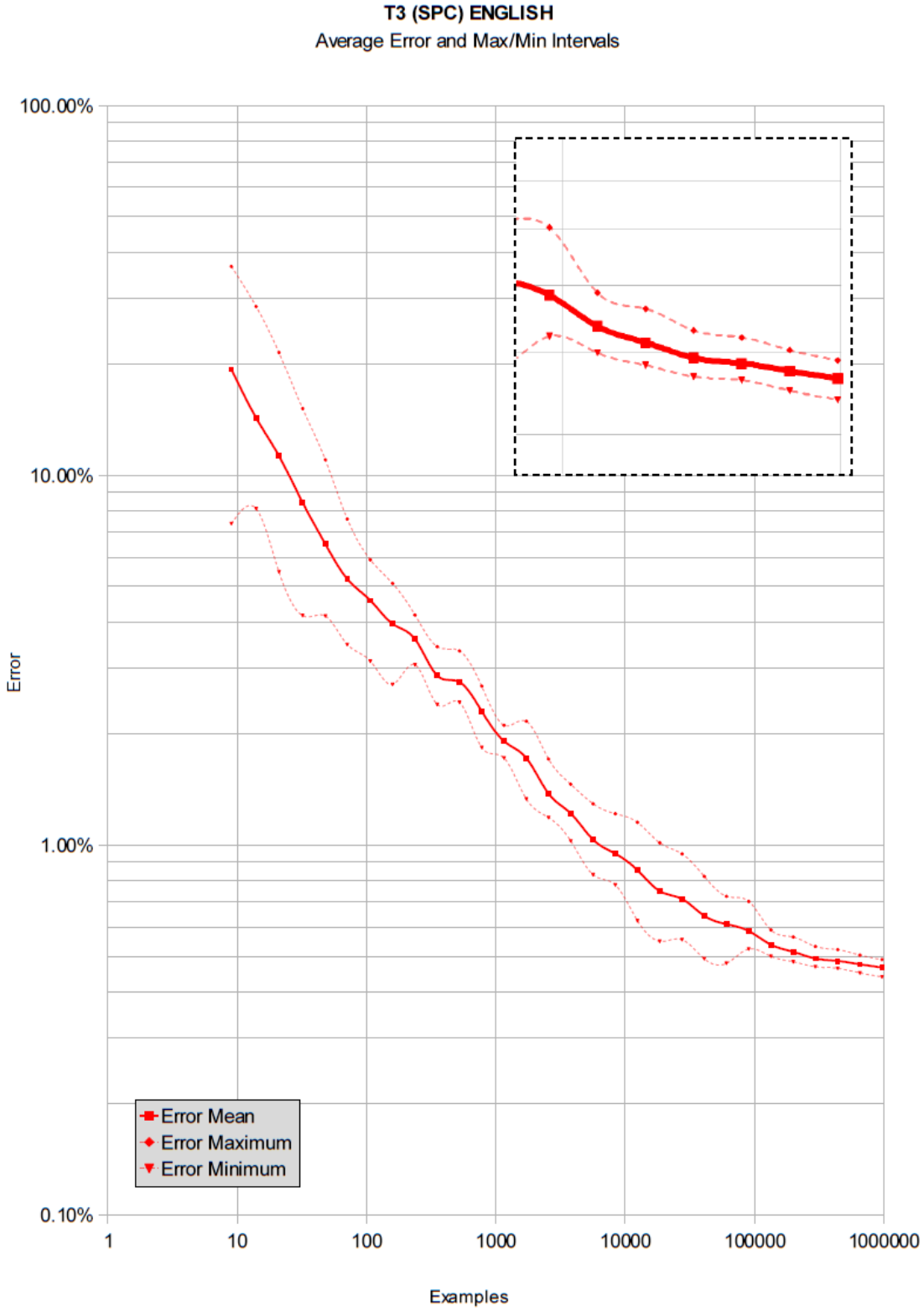

FIG. 83: Evolució de l'error de T3(SPC) a un entrenament incremental estàndard.

EVOLUCIÓ DE T5 (BRK)

Finalment, la corba de la tasca T5 **[Fig. 84]** mostra un error inicial aproximat del 11% i, tot i evolucionar més esglaonadament, assoleix una espectacular reducció, fins arribar a un error inferior al 0,03%. A més, no mostra indicis d'haver assolit cap mena de saturació i, per tant, en cas de disposar de més dades encara l'hauria pogut reduir més.

[Espai intencionadament en blanc per alinear el text amb les figures.]

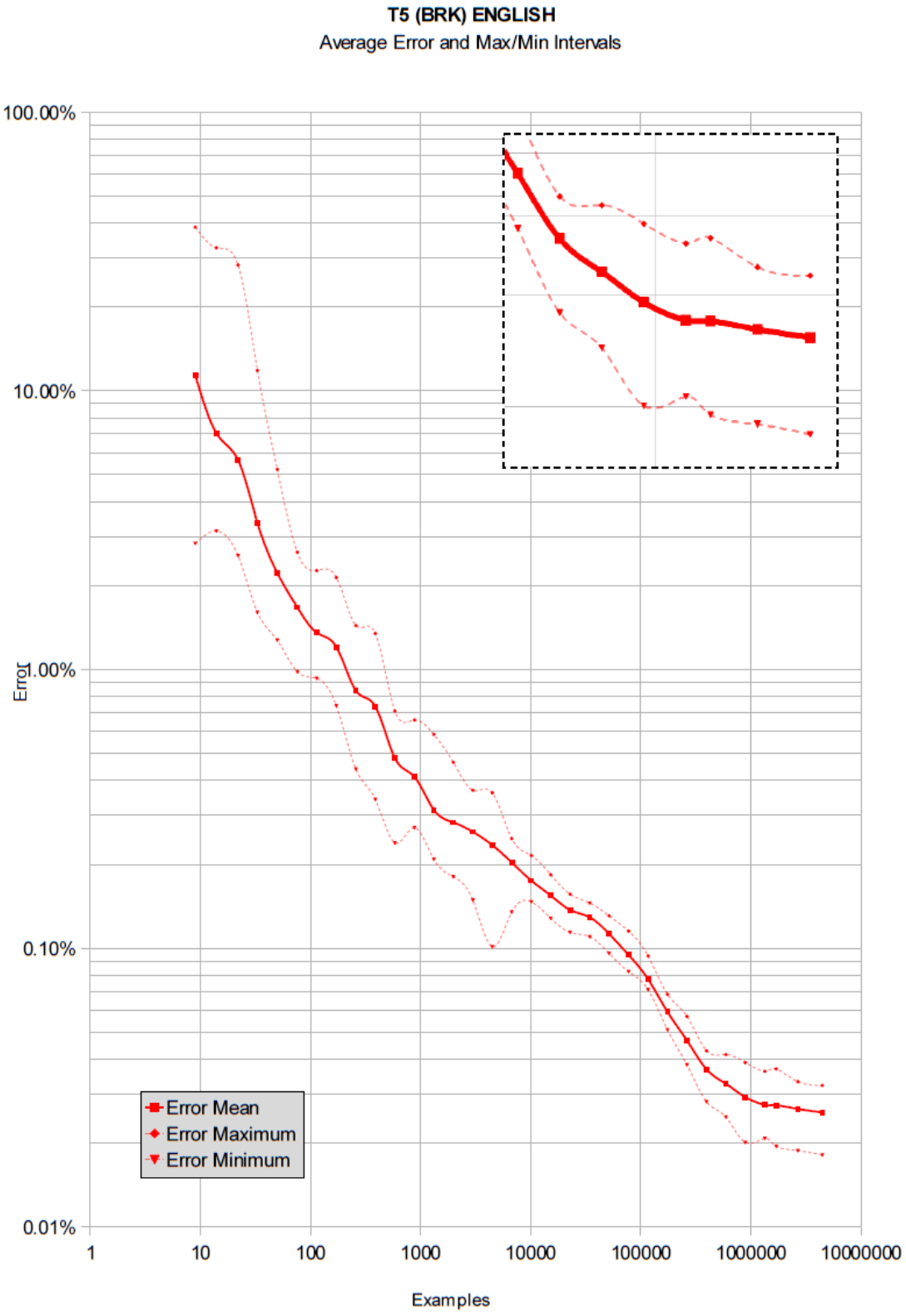

FIG. 84: Evolució de l'error de T5 (BRK) a un entrenament incremental estàndard.

11.5 CONCLUSIONS

En aquest capítol s'han introduït les corbes d'avaluació, una representació de l'evolució de l'entrenament que és utilitzada intensivament en la resta de capítols.

També s'ha proposat relacionar el concepte de saturació amb la proporció entre la quantitat d'exemples disponibles i la mida de l'univers de representació de la tasca. Seguint aquest punt de vista s'ha proposat la creació d'un *índex de densitat* que pugui utilitzar-se com a indicador del risc de saturació. Tot i que caldria una recerca més profunda per poder confirmar la seva validesa, és interesant observar que les corbes que presenten indicis de saturació es corresponen precisament amb les tasques que tenien els índex de densitat més elevats.

Però la conclusió més important d'aquest capítol és que les quatre tasques seleccionades no només representen una amplia gama de tasques de PLN, amb diferents nivells lingüístics i diferent quantitat d'etiquetes i diferents mides de corpus, sinó que també mostren una considerable diversitat en el grau de saturació, els valors absoluts de l'error assolit i la ratio de la seva reducció.

A la taula següent **[Taula 51]** es mostra un resum de la diversitat de característiques de les tasques seleccionades. Caldrà tenir en compte aquestes dades a l'hora d'interpretar els resultats dels propers experiments i explicar les diferències entre tasques:

#	Tasca	Etiquetes	Exemples	Error final	Saturació	Reducció Error
T1	**POS**	88	278,000	2.61%	No	10:1
T2	**ENT**	3	523,000	3.57%	Si	2:1
T3	**SPC**	2	1,080,000	0.47%	Inicial	20:1
T5	**BRK**	2	4,966,000	0.028%	No	400:1

TAULA 51: Característiques finals de les tasques seleccionades: POS, ENT, SPC i BRK.

12 UTILITZACIÓ DE PRE-ENTRENAMENT

12.1 INTRODUCCIÓ

En aquest capítol es presenta la primera de les tècniques d'entrenament incremental que poden proporcionar avantatges en relació a l'entrenament incremental estàndard: el pre-entrenament del model amb un corpus auxiliar.

En primer lloc es descriu en què consisteix aquesta tècnica i quins beneficis pot aportar. Seguidament es presenten els corpus auxiliars que s'han utilitzat per a les quatre tasques seleccionades i es comparen amb els entrenaments basats en els corpus principals.

En segon lloc es presenta el paràmetre `Weight`, que determina el pes relatiu entre el corpus auxiliar i el principal, i s'analitzen els seus efectes teòrics a l'aplicar aquesta tècnica. Tot seguit es mostren els resultats dels primers experiments on es compara l'entrenament incremental estàndard amb la utilització de pre-entrenament per diferents valors d'aquest paràmetre.

Finalment s'analitzen els resultats dels quatre experiments, cosa que obliga a introduir la variable de *similitud entre corpus* per obtenir una explicació consistent amb tots ells. De totes maneres els resultats de les quatre tasques no presenten cap dubte: la utilització de pre-entrenament permet reduir l'error a la fase inicial de l'entrenament i una selecció curosa del paràmetre `Weight` permet aconseguir-ho sense perjudicar l'error final.

12.2 PRE-ENTRENAMENT

Un dels avantatges que presenta l'entrenament incremental d'un classificador és la possibilitat d'entrenar-lo de manera discontínua, en diverses etapes i amb diferents corpus. Aquesta característica permet aprofitar la disponibilitat d'un corpus similar que tot i no correspondre's a la tasca final tingui certa similitud amb ella.

Per exemple, en el cas d'haver d'entrenar un corpus en un idioma minoritari del qual no es disposa de corpus d'entrenament, es pot preentrenar el classificador amb un corpus equivalent però d'una altra llengua de la mateixa família. O fer el mateix amb dos corpus del mateix idioma però de diferent registre o temàtica, com preentrenar un classificador amb textos periodístics i, posteriorment, continuar amb textos acadèmics.

La idea subjacent és que un corpus determinat pot induir un model que, tot i no ser òptim per a la tasca desitjada, sigui millor que un model en blanc; en certa manera s'utilitza un corpus auxiliar com a *prior* estadístic del corpus principal. Tot i que amb un classificador

batch també es podrien recombinar diferents corpus, la diferència es troba en què amb un classificador incremental podem ajustar de forma dinàmica els pesos relatius del corpus auxiliar i del corpus principal. És a dir, donar menys pes al corpus de pre-entrenament i més pes al corpus definitiu de forma que al començament, quan el classificador encara no ha vist prou dades de la tasca final, es recolzi en les dades del pre-entrenament, però a mesura que es van analitzant més dades de la tasca final, aquestes vagin agafant preponderància i el model vagi oblidant les del pre-entrenament.

12.3 CORPUS AUXILIARS

Recordem **[9.3 Corpus Textuals]** com a l'hora de triar els corpus es va optar per buscar corpus bilingües, de manera que cada un d'ells estigués format per dos conjunts de texts en un idioma diferent. El que premeditadament es buscava era disposar d'uns corpus auxiliars amb els quals pre-entrenar els classificadors de les tasques seleccionades.

És important tenir en compte que, perquè dos corpus puguin utilitzar-se per induir incrementalment un mateix model, cal que les anotacions fetes comparteixin etiquetari i que els dos conjunts de dades d'entrenament utilitzin la mateixa representació.

El primer que es va fer va ser pre-entrenar quatre classificadors amb els corpus auxiliars, corresponents a un altre idioma, i obtenir les seves corbes d'avaluació. L'objectiu era comparar l'evolució dels classificadors amb els corpus auxiliars i els corpus principals, i comprovar que el comportament era similar. En els propers punts es mostren les gràfiques superposades de l'entrenament principal i de l'entrenament auxiliar, per cada una de les taques.

CORPUS AUXILIAR DE T1 (POS)

El gràfic de la **[Fig. 85]** mostra l'evolució de l'error de la tasca T1 entrenada amb el corpus principal en català, línia de color vermell, i el d'un entrenament equivalent però amb el corpus auxiliar en espanyol, línia de color marró. Es pot observar com les dues corbes evolucionen de manera pràcticament paral·lela, cosa que suggereix que la desambiguació de PoS presenta la mateixa dificultat en els dos idiomes.

CORPUS AUXILIAR DE T2 (ENT)

El gràfic de la **[Fig. 86]** mostra l'evolució de l'error de la tasca T2 entrenada amb el corpus principal en català, línia de color vermell, i el d'un entrenament equivalent però amb el corpus auxiliar en espanyol, línia de color marró. S'observa una important diferència al llarg de tot l'entrenament. Finalment l'error assolit en el corpus en català, aproximadament un 4%, és significativament més alt que l'assolit en el corpus en espanyol, inferior al 3%. Per tant sembla que la detecció d'entitats és lleugerament més difícil en el corpus en català.

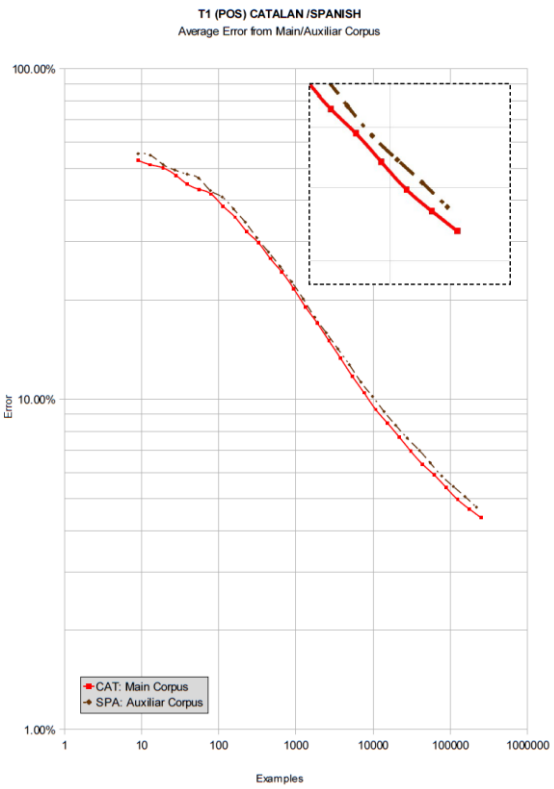

FIG. 85: Comparativa amb la tasca auxiliar de T1(POS): corpus Català vs Espanyol.

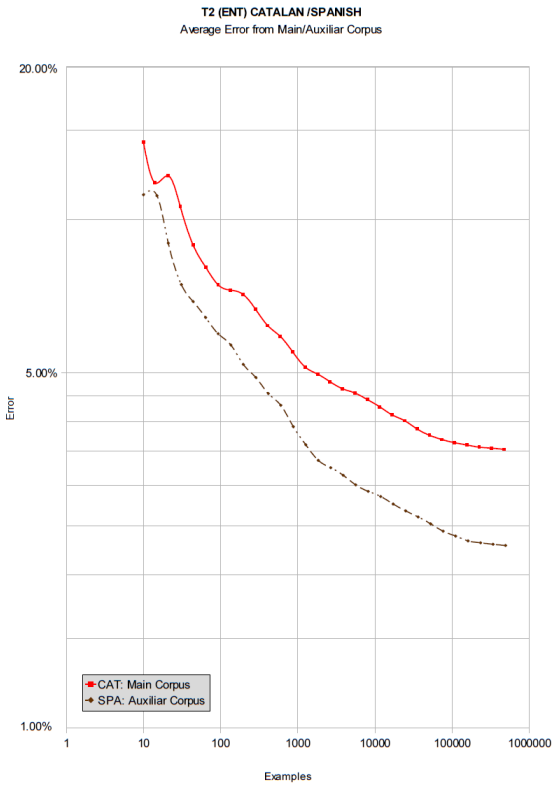

FIG. 86: Comparativa amb la tasca auxiliar de T2(SPC): corpus Català vs Espanyol.

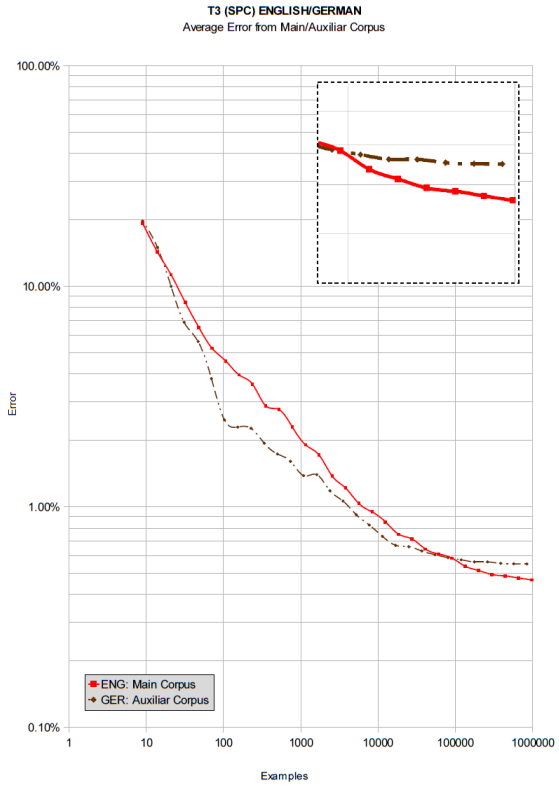

FIG. 87: Comparativa amb la tasca auxiliar de T3(SPC): corpus Anglès vs Alemany.

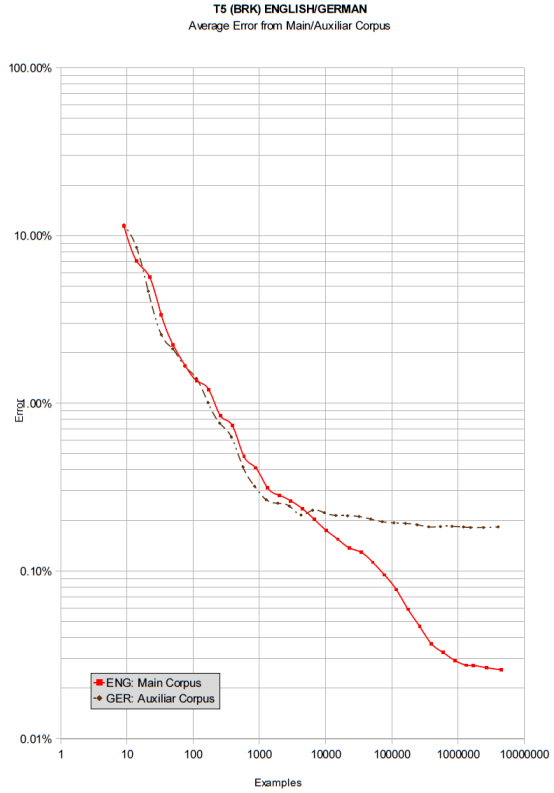

FIG. 88: Comparativa amb la tasca auxiliar de T5(BRK): corpus Anglès vs Alemany.

CORPUS AUXILIAR DE T3 (SPC)

El gràfic de la **[Fig. 87]** mostra l'evolució de l'error de la tasca T3 entrenada amb el corpus principal en anglès, línia de color vermell, i el d'un entrenament equivalent però amb el corpus auxiliar en alemany, línia de color marró. Tornem a veure dues corbes que evolucionen de forma similar, tot i presentar ritmes diferents. Sembla que l'alemany, inicialment, permet reduir l'error més ràpidament però curiosament assoleix un error final lleugerament més elevat (0,55% respecte 0,46%).

CORPUS AUXILIAR DE T5 (BRK)

Finalment, el gràfic de la **[Fig. 88]** mostra l'evolució de l'error de la tasca T5 entrenada amb el corpus principal en anglès, línia de color vermell, i el d'un entrenament equivalent però amb el corpus auxiliar en alemany, línia de color marró. En aquest cas les dues corbes evolucionen paral·lelament fins als 5.000 exemples aproximadament, a partir d'aquest punt el corpus auxiliar es satura amb un error al voltant de 0,2%, i el corpus principal continua millorant fins a un error deu vegades inferior, 0,03%. Resultats que indiquen una major dificultat o un corpus més sorollós, és a dir, amb més errors d'anotació, en el cas de la separació d'oracions en els textos en alemany.

12.4 PARÀMETRE *WEIGHT*

A l'hora d'entrenar un classificador amb dos corpus diferents és possible controlar el pes relatiu de les dades de cada un. Concretament el que podem controlar és el pes individual de cada exemple i, per tant, podem assignar un pes a tots els exemples del corpus principal, diguem-li A, i un altre pes a tots els exemples del corpus auxiliar, diguem-li B. Si fem la mitjana dels pesos dels corpus tenint en compte la quantitat d'exemples de cada un podem obtenir el pes relatiu de cada corpus segons:

$$RatioCorpus_A = \frac{W_A \cdot |examples_A|}{(|W_A \cdot |examples_A|) + (W_B \cdot |examples_B|)}$$

On W_A és el pes donat als exemples del corpus principal, W_B el pes donat als exemples del corpus auxiliar, i $|exemples_X|$ la quantitat d'exemples que formen un corpus determinat.

Com el classificador utilitzat és un *Naïve Bayes* i el seu model intern són una sèrie de matrius de comptadors d'ocurrències, el més senzill és augmentar el pes de les dades noves, de manera que cada nova ocurrència compti com W vegades. Així per exemple, si als exemples del corpus auxiliar li donem un pes W_B igual a 1 i als del corpus principal li donem un pes W_A de 10, suposant que tenen una quantitat similar d'exemples, estarem combinant els dos corpus donant un pes del 9% (1/11) al pre-entrenament i un 91% (10/11) a l'entrenament pròpiament dit. I això és exactament el que es farà exactament en

4els experiments d'aquest capítol, variar el factor multiplicatiu *W* dels exemples del corpus principal per observar l'efecte de diferents pesos relatius.

Abans de mostrar els resultats obtinguts, cal tenir en compte quin ha estat el raonament que s'ha seguit per deduir la influència teòrica d'aquest valor.

- A l'inici de l'entrenament el valor *W* no hauria de tenir cap influència, ja que en aquest moment el model únicament està format per les dades del pre-entrenament, amb tots els exemples amb el mateix pes unitari.
- Al final de l'entrenament el valor *W* hauria de ser determinant per conèixer la proporció, el *RatioCorpus*, de la seva participació en el model obtingut. Però si es defineix un valor prou elevat (100, 1000, ...), a efectes pràctics, el model obtingut serà pràcticament el mateix que tindríem sense pre-entrenament.

Així doncs, sembla que la millor opció seria proporcionar un valor suficientment alt que permeti obtenir els beneficis buscats amb aquesta tècnica: a) reduir l'error en la fase inicial, és a dir, durant el període transitori en el qual no s'han observat prou exemples, i b) eliminar qualsevol efecte del pre-entrenament en la fase final, de manera que el model obtingut coincidiria amb l'induït únicament amb dades del corpus principal. Per contra, la utilització de *W* amb valors baixos provocaria que el pre-entrenament amb dades auxiliars interferís amb el model final i s'obtinguessin errors lleugerament superiors.

A la següent secció s'analitzen els resultats obtinguts a les quatre tasques en cas d'utilitzar pre-entrenament i assignar pesos creixents (1, 10, 100, 1.000 i 10.000) als corpus principals.

12.5 EVOLUCIÓ DELS MODELS

En els següents apartats d'aquesta secció es mostren les corbes d'avaluació obtingudes per a cada una de les tasques de referència, i en cada una d'elles es superposen les corbes corresponents a diferents valors de W. A la pàgina esquerra de cada figura s'analitzen les principals conclusions que es poden extreure de l'observació d'aquestes gràfiques:

[Espai intencionadament en blanc per alinear el text amb les figures.]

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 89]** mostra l'evolució de l'error de la tasca T1 entrenada directament amb el corpus principal en català, línia de color vermell, i utilitzant pre-entrenament amb el corpus espanyol per a diferents valors per *W*, línies de color marró.

A primer cop d'ull s'observa com totes les corbes amb pre-entrenament presenten errors més baixos durant l'etapa inicial, i errors similars en l'etapa final, tal com es preveia. Però si s'analitza amb atenció l'efecte del paràmetre *W* s'observa sorprenentment que el seu efecte és invers a l'esperat. És a dir, a mesura que augmenta el seu valor també ho fa l'error final, amb la paradoxa que el millor resultat s'obté donant el mateix pes (*W*=1) a tots els exemples. I encara més sorprenent, l'error final no només és igual a l'obtingut sense pre-entrenament, sinó que és inferior, una anomalia que quedarà explicada a la següent secció.

Una altre resultat no tant evident però igual de sorprenent és que a l'etapa inicial, per pesos molt elevats (*W*=10.000), l'error augmenta a mesura que avança l'entrenament, fins al voltant del primer centenar d'exemples observats. La explicació rau en les conseqüències de multiplicar per 10.000 les ocurrències observades, ja que els 100 primers exemples tenen un pes total equivalent al milió d'exemples del pre-entrenament però amb una representativitat molt inferior. El resultat és equivalent al d'un sobre-entrenament (*overfitting*) artificial, on la distribució del primer centenar d'exemples no és prou representativa de la distribució del corpus d'avaluació ja que molts casos no hi estan presents.

[Espai intencionadament en blanc per alinear el text amb les figures.]

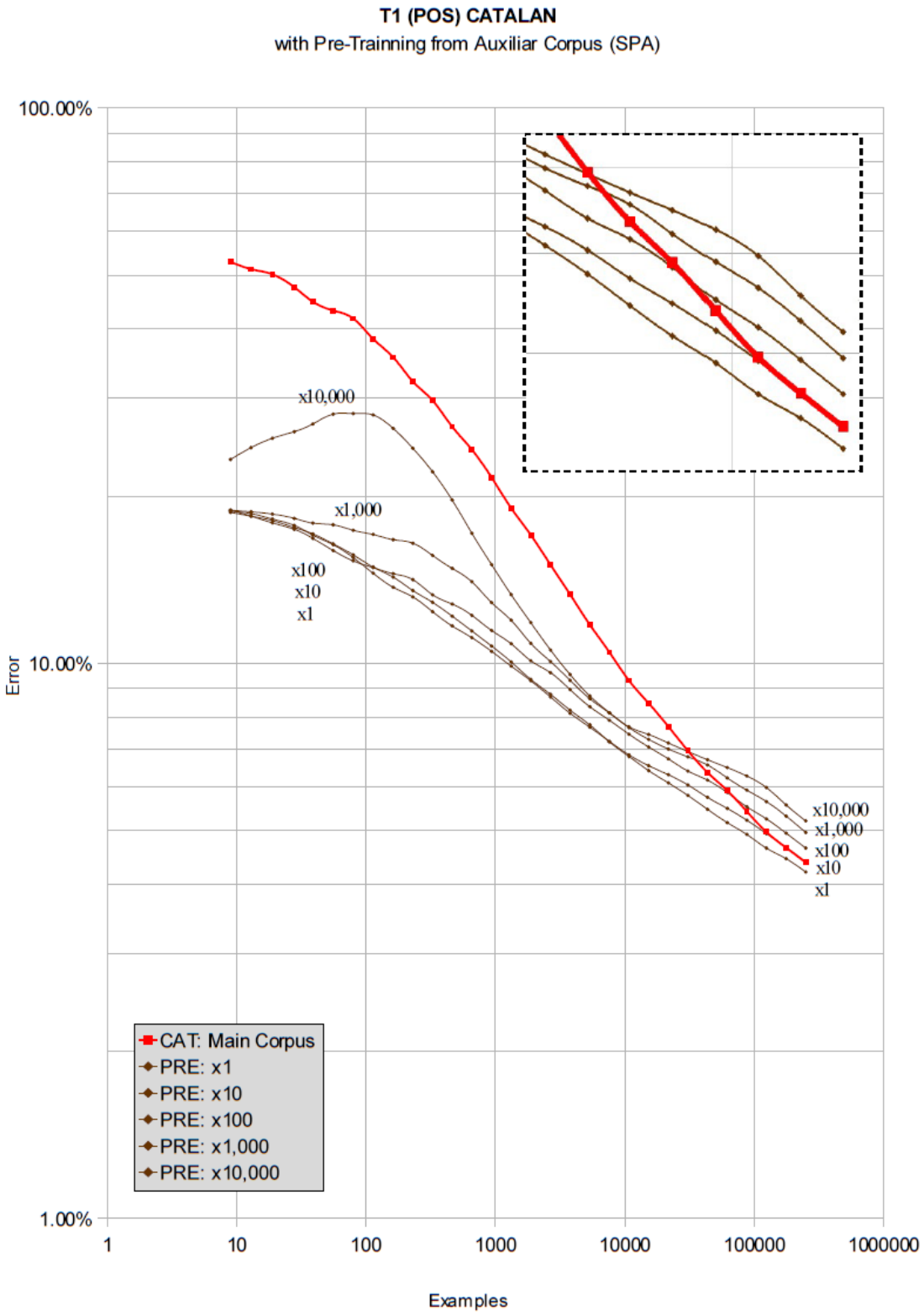

FIG. 89: Evolució dels errors de T1(POS) amb utilització de pre-entrenament.

EVOLUCIÓ DE T2 (ENT)

El gràfic de la **[Fig. 90]** mostra l'evolució de l'error de la tasca T2 amb el corpus principal en català, línia de color vermell, i utilitzant pre-entrenament amb el corpus espanyol per a diferents valors per *W*, línies de color marró.

En aquest cas els resultats són similars, tot i que no tan exagerats. A primer cop d'ull s'observa com les corbes pre-entrenades redueixen el seu error inicial, oferint un petit benefici, mentre que al final de l'entrenament acaben convergint amb l'error de la corba sense pre-entrenament. També s'observa com el paràmetre *W* tampoc es comporta de la manera prevista, i com els millors resultats finals s'obtenen donant el mateix pes als exemples de tots dos corpus.

[Espai intencionadament en blanc per alinear el text amb les figures.]

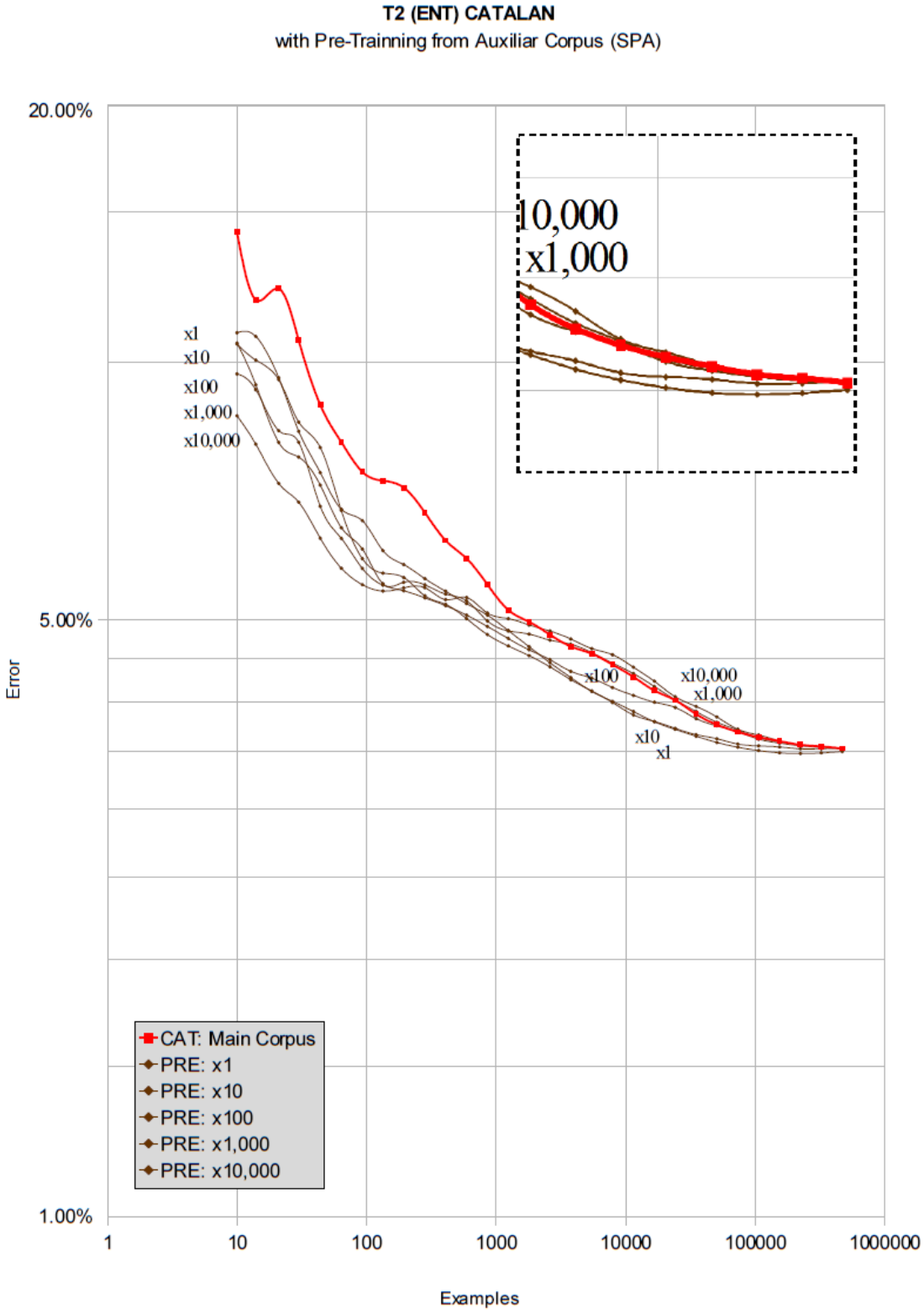

FIG. 90: Evolució dels errors de T2(ENT) amb utilització de pre-entrenament.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 91]** mostra l'evolució de l'error de la tasca T3 entrenada directament amb el corpus principal en anglès, línia de color vermell, i utilitzant pre-entrenament amb el corpus alemany per a diferents valors per *W*, línies de color marró.

Els resultats d'aquests experiments són semblants: el pre-entrenament permet reduir l'error en l'etapa inicial i no afecta a l'error obtingut al finalitzar l'entrenament. A més, en aquest cas, les diferències respecte el paràmetre *W* sí encaixen amb el comportament previst: a l'etapa inicial el valor del paràmetre no és gaire rellevant, i a l'etapa final no ho és per valors suficientment alts (`x100`, `x1.000`, `x10.000`, ...).

És a dir, si el pre-entrenament té un pes similar a l'entrenament principal, interfereix en els resultats finals provocant uns errors superiors al del model no pre-entrenat. I si el pes relatiu del pre-entrenament es minimitza, obtenim una millora en l'etapa inicial sense perjudicar l'etapa final.

[Espai intencionadament en blanc per alinear el text amb les figures.]

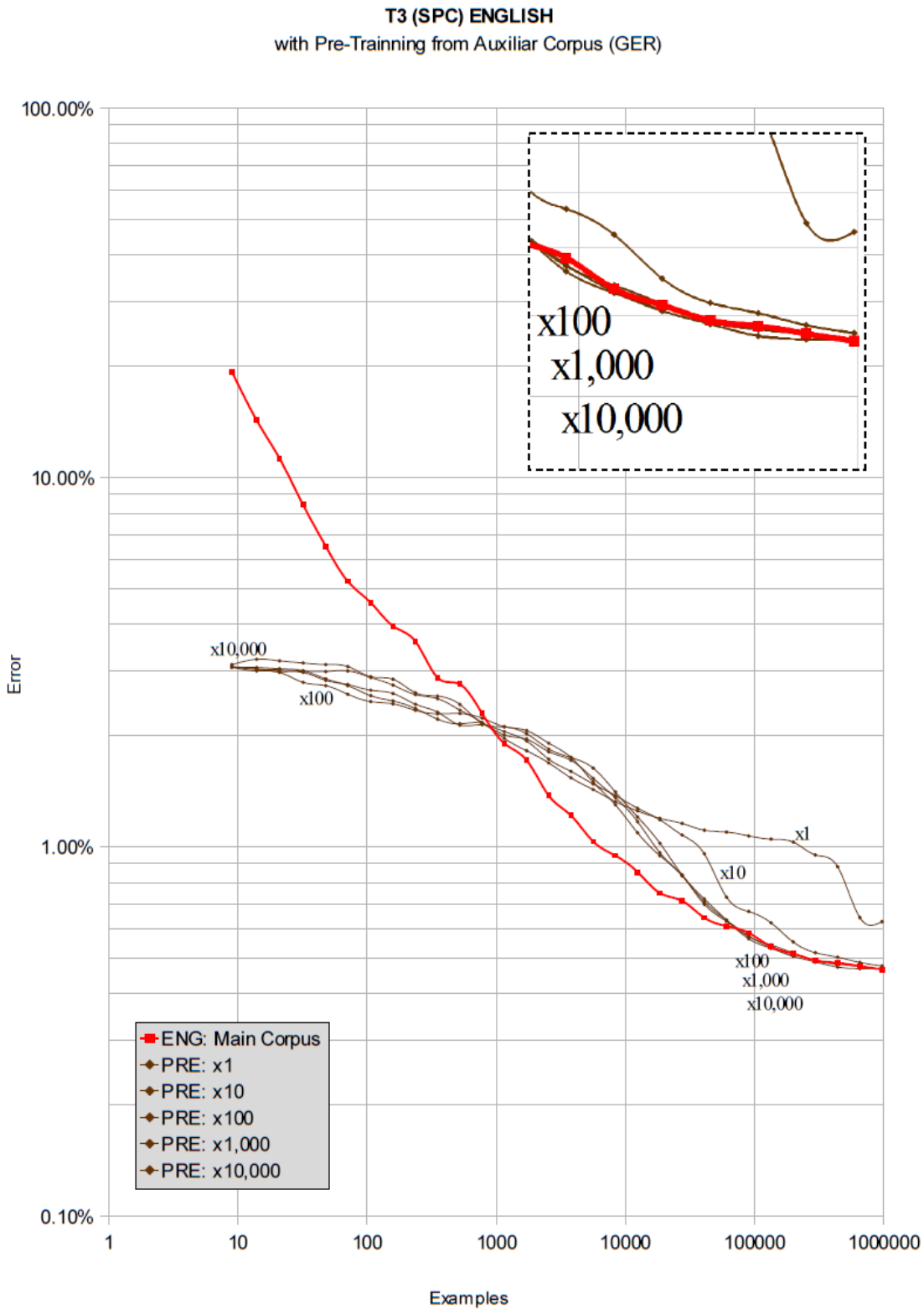

FIG. 91: Evolució dels errors de T3(SPC) amb utilització de pre-entrenament.

EVOLUCIÓ DE T5 (BRK)

El gràfic de la **[Fig. 92]** mostra l'evolució de l'error de la tasca T4 entrenada directament amb el corpus principal, línia de color vermell, i utilitzant pre-entrenament amb el corpus alemany per a diferents valors per *W*, línies de color marró.

En aquesta tasca els resultats són similars al cas anterior, excepte que permet visualitzar les diferències de manera més extrema. En tots els casos el pre-entrenament millora l'error durant l'etapa inicial i els errors finals convergeixen amb el valor de l'entrenament de referència.

És interessant observar com els pesos més baixos (`x1` i `x10`) donen lloc a errors més grans i pràcticament constants, fins a un punt avançat de l'entrenament on comencen a reduir-se molt ràpidament, aquests punts se situen al voltant dels 400.000 i 40.000 exemples respectivament. Els valors no semblen casuals, en aquests punts el pes del corpus principal comença a ser un ordre de magnitud superior al pes del corpus auxiliar; és a dir, punts a partir dels quals la influència del pre-entrenament pot ser ignorada.

[Espai intencionadament en blanc per alinear el text amb les figures.]

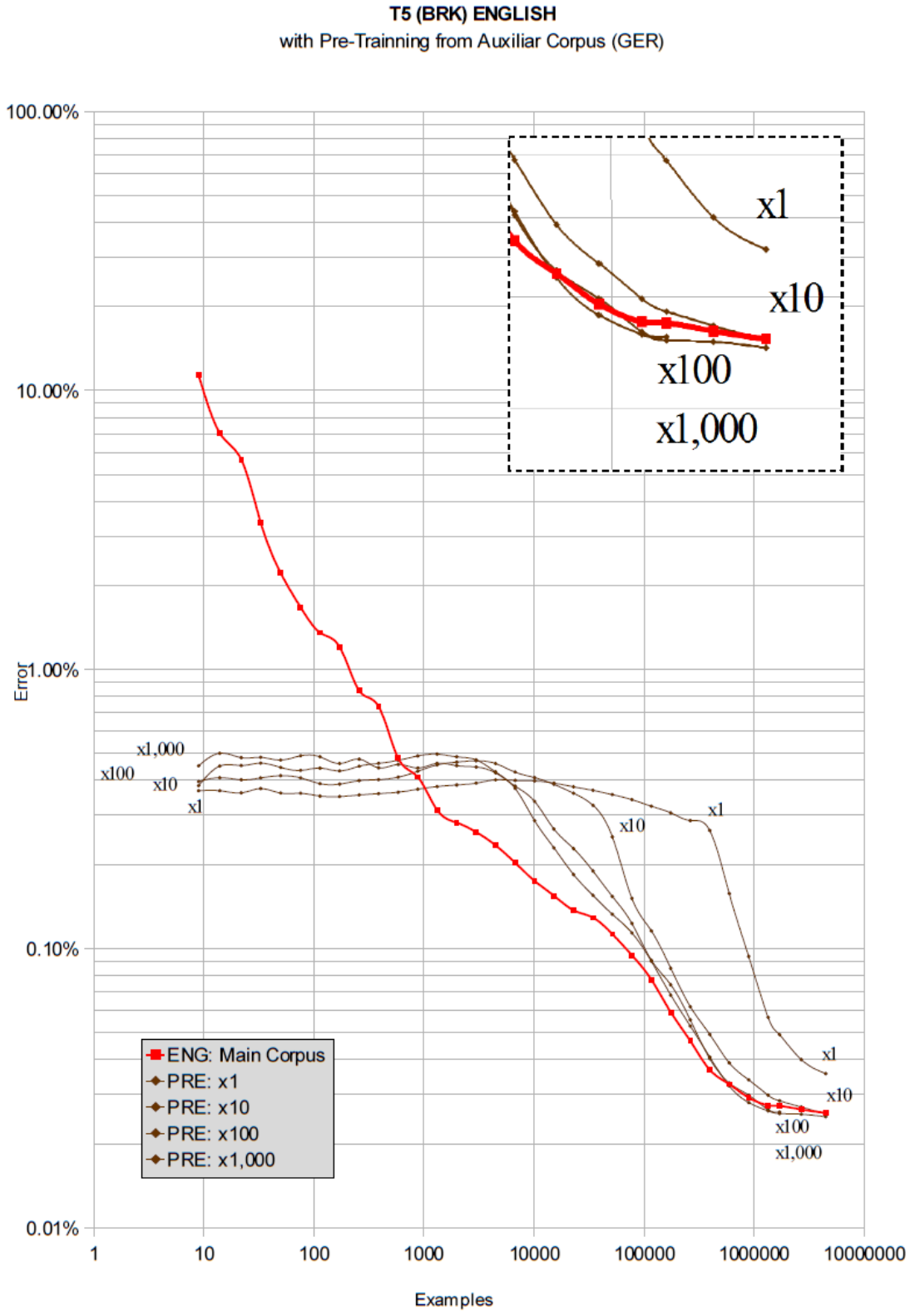

FIG. 92: Evolució dels errors de T5(BRK) amb utilització de pre-entrenament.

12.6 SIMILITUD ENTRE CORPUS

Els resultats dels experiments de T1 i T2, especialment del primer, en relació a l'efecte del paràmetre W ens obliguen a replantejar el model proposat **[12.4 Paràmetre *Weight*]** perquè pugui explicar tots quatre resultats. Recordem que en els dos primers experiments els millors resultats s'obtenen igualant el pes del pre-entrenament i de l'entrenament, i en els dos darrers experiments els millors resultats s'obtenen minimitzant el pes del pre-entrenament.

L'explicació proposada incorpora una variable addicional: la similitud entre el corpus principal i el corpus auxiliar. Per entendre les diferències entre els dos casos analitzem dues situacions límit: un cas on els dos corpus tinguin la màxima similitud i coherència i un cas on els dos corpus no tinguin cap similitud i presentin contradiccions.

12.6.1 MÍNIMA SIMILITUD: CORPUS OPOSATS

Comencem pel cas on tots dos corpus, tot i compartir etiquetari i representació, presenten diferències distribucionals molt importants i, fins i tot, contenen exemples concrets contradictoris.

Aquest marc és el que pressuposava l'explicació inicial, i és on el model induït durant el pre-entrenament amb el corpus auxiliar, que inclou exemples "erronis" segons el criteri del corpus principal, pot interferir i perjudicar al model final. Precisament per això cal assegurar que, a mesura que avança l'entrenament real, el pes del corpus de pre-entrenament es vagi diluint. I per això, en aquest cas, es compleix la necessitat de minimitzar el pes del pre-entrenament i maximitzar el pes del corpus principal mitjançant valors de *W* elevats (`x100`, `x1000`).

12.6.2 MÀXIMA SIMILITUD: SUBCORPUS D'UN CORPUS

El cas oposat, on els dos corpus presenten la màxima similitud i coherència interna, es produeix precisament quan parlem de dos fragments d'un mateix corpus. En aquest cas la similitud és màxima i difícilment contindran exemples oposats.

Curiosament en aquest cas els millors resultats s'obtenen donant el mateix pes a tots dos, de manera que no se sobreponderi la distribució de cap d'ells, i per tant amb un valor baix per *W*, idealment d'1. En aquest cas, incloure un pre-entrenament produiria exactament el mateix model que produiria un entrenament incremental en una sola etapa però amb un corpus més gran format per la suma del corpus principal i l'auxiliar.

Aquesta similitud és el que explicaria el comportament de *W* en les dues primeres tasques, T1 i T2.

12.7 CONCLUSIONS

En aquest capítol s'ha descrit la utilització d'una tècnica d'entrenament basada en el pre-entrenament del model mitjançant un corpus auxiliar. I s'han presentat els resultats d'aplicar aquesta tècnica per a entrenar classificadors per a les quatre tasques de referència.

Els resultats confirmen que és possible obtenir millors models i reduir els errors en l'etapa inicial transitòria mitjançant la utilització d'un corpus auxiliar per pre-entrenar els classificadors. També demostren que si es tria amb cura el pes relatiu de cada corpus, mitjançant el valor *W*, aquest benefici pot assolir-se sense perjudicar el model final, de forma que el resultat final sigui el mateix que si no s'hagués pre-entrenat.

En el cas en que els dos corpus siguin similars i presentin una elevat grau de coherència, és recomanable utilitzar *W* baixos (`x1`, `x10`) que donin un pes similar a tots dos corpus. En el cas que tinguem corpus equivalents però amb diferències significatives, és recomanable optar per valors de *W* més elevats (`x100`, `x1000`) que minimitzin la interferència del pre-entrenament en el model final. Queda pendent definir alguna mètrica o metodologia que permeti, *a priori*, determinar la similitud i coherència entre un corpus principal i un corpus auxiliar determinats.

Si apliquem aquesta explicació a les tasques de referència, podem concloure que en relació a la tasca T1(POS), els dos idiomes de l'*AnCora*, són pràcticament equivalents, probablement per la similitud dels n-grames PoS entre el català i el castellà. Sorprenentment, en aquesta tasca el pre-entrenament amb el corpus castellà ha tingut un efecte equivalent a haver duplicat la mida del corpus català. Fins i tot a nivell quantitatiu, ja que si s'observa amb atenció el gràfic de la [**Fig. 89**] es pot comprovar com l'extrem de la corba inferior (*W*=1) coincideix amb el valor que tindria una hipotètica continuació de la corba vermella de referència.

Per contra, el comportament de *W* en les tasques T3(SPC) i T5(BRK) suggereixen que els corpus B1 i B3, en essència els dos idiomes del *DeNews*, tot i ser prou semblants presenten algunes diferències significatives, probablement degut a les diferències ortotipogràfiques, com la diferent utilització de les majúscules entre l'anglès i l'alemany .

13 UTILITZACIÓ D'ENTRENAMENT ACTIU

13.1 INTRODUCCIÓ

En aquest capítol es descriu la segona de les tècniques d'entrenament incremental que poden augmentar l'eficiència i proporcionar importants avantatges en relació a l'entrenament incremental estàndard: l'entrenament actiu, una tècnica basada en la selecció d'exemples d'entrenament **[Cohn et al., 1994]** segons els principis de l'aprenentatge actiu **[Thompson et al., 1999]**.

En primer lloc es descriu en què consisteix aquesta tècnica, quina metodologia s'ha utilitzat per portar a terme els experiments corresponents, i la importància i efectes del paràmetre *Confidence*, o llindar de confiança, determinant per portar a terme la selecció d'exemples.

En les dues seccions següents s'avaluen els resultats dels experiments per a diferents valors de dos paràmetres (`Rand` i `First`) que controlen l'aplicació de l'entrenament actiu. Tot seguit es realitza una comparativa exhaustiva de l'eficiència i error final obtingut en els diferents experiments. L'objectiu és determinar els valors que proporcionen millors resultats i analitzar el seu efecte en els entrenaments de les diferents tasques. Una vegada determinats els valors òptims, es comparen les corbes d'avaluació obtingudes de l'entrenament actiu amb les de l'entrenament incremental estàndard, utilitzat com a referència, i es quantifica l'estalvi i la millora obtinguts.

Finalment, a les conclusions, es sintetitzen els resultats mitjançant les taules amb els resultats obtinguts en les quatre tasques de referència. En tots els casos s'obtenen errors finals inferiors als obtinguts amb l'entrenament estàndard i, tot i que les diferents tasques presenten valors d'eficiència molt diversos, aquests errors sempre s'assoleixen utilitzant una fracció del corpus disponible. Per tant, no hi ha dubte que l'entrenament actiu permet obtenir models lleugerament més precisos utilitzant una quantitat molt inferior d'exemples, una eficiència que suposa un cost d'anotació varies vegades inferior.

13.2 ENTRENAMENT ACTIU

A la primera part d'aquest document **[5. Classificació Inter-Activa]** s'han explicat els avantatges que suposa utilitzar algorismes incrementals en l'aprenentatge actiu **[5.3 Aprenentatge Actiu]** i com aquesta sinergia permet desenvolupar entorns molt eficients d'anotació *inter-activa*. Si canviem el punt de vista del classificador cap a l'entrenador, el terme *aprenentatge actiu* esdevé *entrenament actiu*, que podem descriure com una tècnica d'inducció de models en la qual l'entrenador anota i resol aquells exemples que

han estat seleccionats prèviament pel classificador. O amb altres paraules, és una tècnica d'entrenament on s'utilitza activament el model del classificador per a seleccionar aquells exemples més informatius i que en cas de ser anotats poden millorar més el model.

L'objectiu que es busca és la inducció eficient de models mitjançant la utilització d'una fracció dels exemples disponibles, i l'essència de la tècnica consisteix bàsicament en descartar aquells exemples que siguin redundants o poc informatius. Seleccionar els exemples menys freqüents o els més ambigus de classificar permet reduir considerablement el cost d'anotació sense perjudicar la qualitat del model obtingut.

Per quantificar l'estalvi d'exemples i la millora del model que suposaria la utilització d'aquesta tècnica d'entrenament en una anotació *inter-activa*, s'ha substituït la creació d'una interfície gràfica i l'anotació manual dels exemples per un procés automàtic en el que l'ordinador proporciona, a partir d'un corpus anotat, les anotacions que el classificador requereix.

En aquest procés, en comptes d'iterar per les dades d'entrenament i mostrar al classificador cada exemple amb la seva categoria, només se li mostra la categoria si el model el considera dubtós, és a dir, si el grau de certesa amb el qual el classificaria és inferior a un llindar determinat. Aquest procés simula un entrenament *inter-actiu* on, a partir d'un conjunt de dades sense etiquetar, l'expert anotaria manualment aquells exemples seleccionats pel classificador.

13.3 PARÀMETRE *CONFIDENCE*

A l'hora d'aplicar tècniques d'aprenentatge actiu amb algorismes *batch* el més habitual és analitzar totes les dades d'entrenament i ordenar-les de més a menys informatives o segons la seva certesa o grau d'acord en cas de classificadors basats en sistemes de votació. I en una segona fase se seleccionen els N exemples més informatius, s'etiqueten i s'utilitzen per induir un nou model classificador.

En l'entrenament incremental, donada la seva naturalesa, això no és possible perquè els exemples, pertanyents a una seqüència il·limitada, es processen individualment sense tenir accés al conjunt de dades. L'alternativa és redefinir el procés i, en comptes de triar els *N* exemples més informatius, triar els exemples que superin un determinat llindar, veure **[Fig. 93]**.

El valor d'aquest paràmetre `Confidence` (*confidence threshold*) controlarà indirectament la proporció entre els exemples descartats i els exemples seleccionats per entrenar el model. Un llindar de confiança elevat només descartarà la petita fracció d'exemples que serien classificats amb un grau de certesa molt alt, però obligaria a anotar la immensa majoria dels exemples, i per tant no proporcionaria un gran estalvi en l'anotació. Un llindar de confiança baix només obligaria a anotar els casos més dubtosos, produint un gran estalvi, però el model donaria per apresos una gran quantitat d'exemples amb un grau de certesa

relativament baix. Per tant és determinant seleccionar un llindar que maximitzi l'estalvi però que minimitzi el desaprofitament d'exemples informatius.

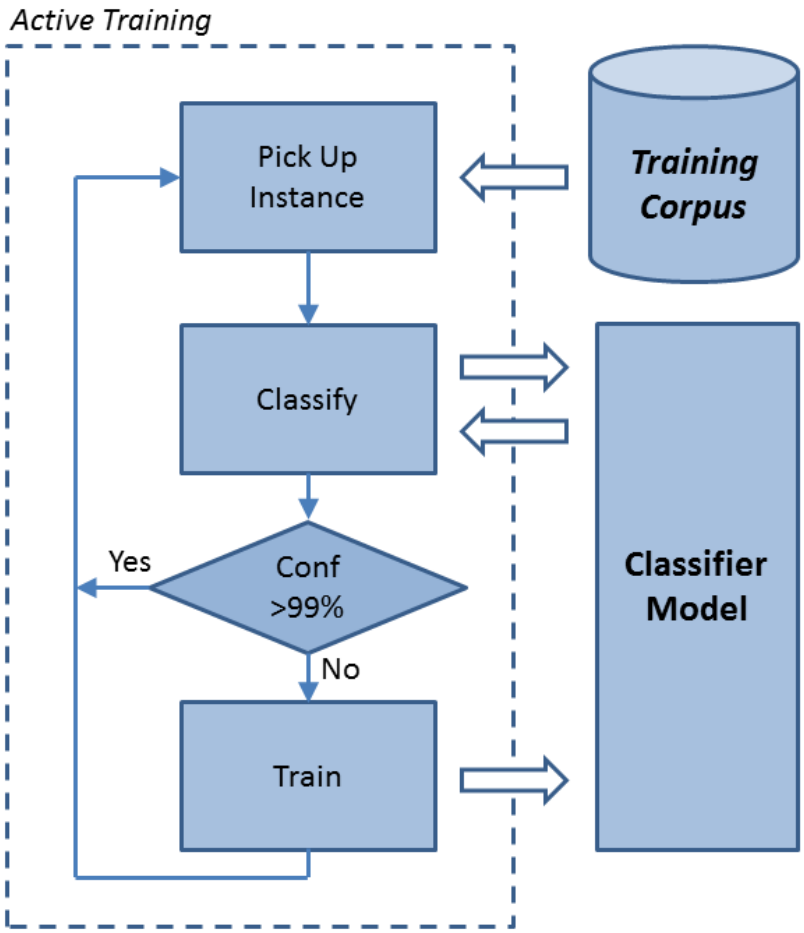

FIG. 93: Diagrama de l'algorisme aplicat durant l'entrenament actiu.

En els primers experiments es van provar graus de certesa corresponents al 60%, 90% i 99%, però de seguida es va comprovar que aquest paràmetre imposava un límit inferior a l'error final obtingut pel model. Per tant la majoria d'experiments s'han fet amb llindars elevats, sent els valors més habituals 99%, 99,9% i 99,99%.

13.4 PARÀMETRE *RAND*

A la secció anterior s'ha explicat que l'entrenament actiu d'un model es basa en la utilització del mateix model per seleccionar els exemples amb què s'entrenarà, però aquesta dependència introdueix un problema de circularitat. Tot i que aquest problema és soluble en els models incrementals, que són funcionals en tot moment, és cert que alguns autors recomanen afavorir l'exploració de tot l'univers d'exemples **[Cawley, 2011]** per optimitzar els resultats i evitar problemes de biaix que es retroalimentin.

Cal tenir en compte que la qualitat del model és determinant a l'hora d'utilitzar-lo per seleccionar els exemples més informatius i descartar els redundants. Si un classificador sobrepondera la seva capacitat per classificar una determinada regió de l'univers d'exemples, assignant una confiança que no es correspongui amb la realitat, descartarà la majoria dels exemples d'aquesta àrea, dificultant que pugui corregir el model.

Per això cal afavorir la representativitat dels exemples utilitzats per entrenar el model. Una manera d'aconseguir-ho és mostrejar aleatòriament una fracció de la seqüència d'entrenament i només aplicar el filtre de l'entrenament actiu a la resta d'exemples. En els experiments realitzats aquesta fracció del mostreig, el tant per cent seleccionat aleatòriament, ve controlat per un paràmetre que s'ha anomenat `Rand`.

En els següents apartats d'aquesta secció es mostren les corbes d'avaluació dels diferents experiments, per cada una de les tasques de referència, on s'han combinat diferents valors del tant per cent `Rand` i del llindar de `Confidence`. A la pàgina a l'esquerra de cada una de les figures es comenten les característiques més rellevants i, si s'escau, s'interpreten les causes.

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 94]** mostra l'evolució de l'error de la tasca T1 en ser entrenada directament amb el corpus català, línia vermella, juntament amb les corbes dels entrenaments actius amb mostreig aleatori, línies taronges.

S'observa clarament com el pendent de les línies taronges és més pronunciat que el de la línia vermella, és a dir, com l'entrenament actiu redueix l'error més ràpidament i, per tant, pot assolir el mateix error amb menys exemples. S'observa també com a mesura que s'eleva el llindar de `Confidence` les línies assoleixen errors més baixos, i com per un mateix llindar les corbes amb un `Rand` d'1%, taronja fosc, necessiten menys exemples que les corbes amb un `Rand` del 5%, taronja clar. El `Confidence` mínim per assolir el mateix error que l'entrenament estàndard és 0,999.

[Espai intencionadament en blanc per alinear el text amb les figures.]

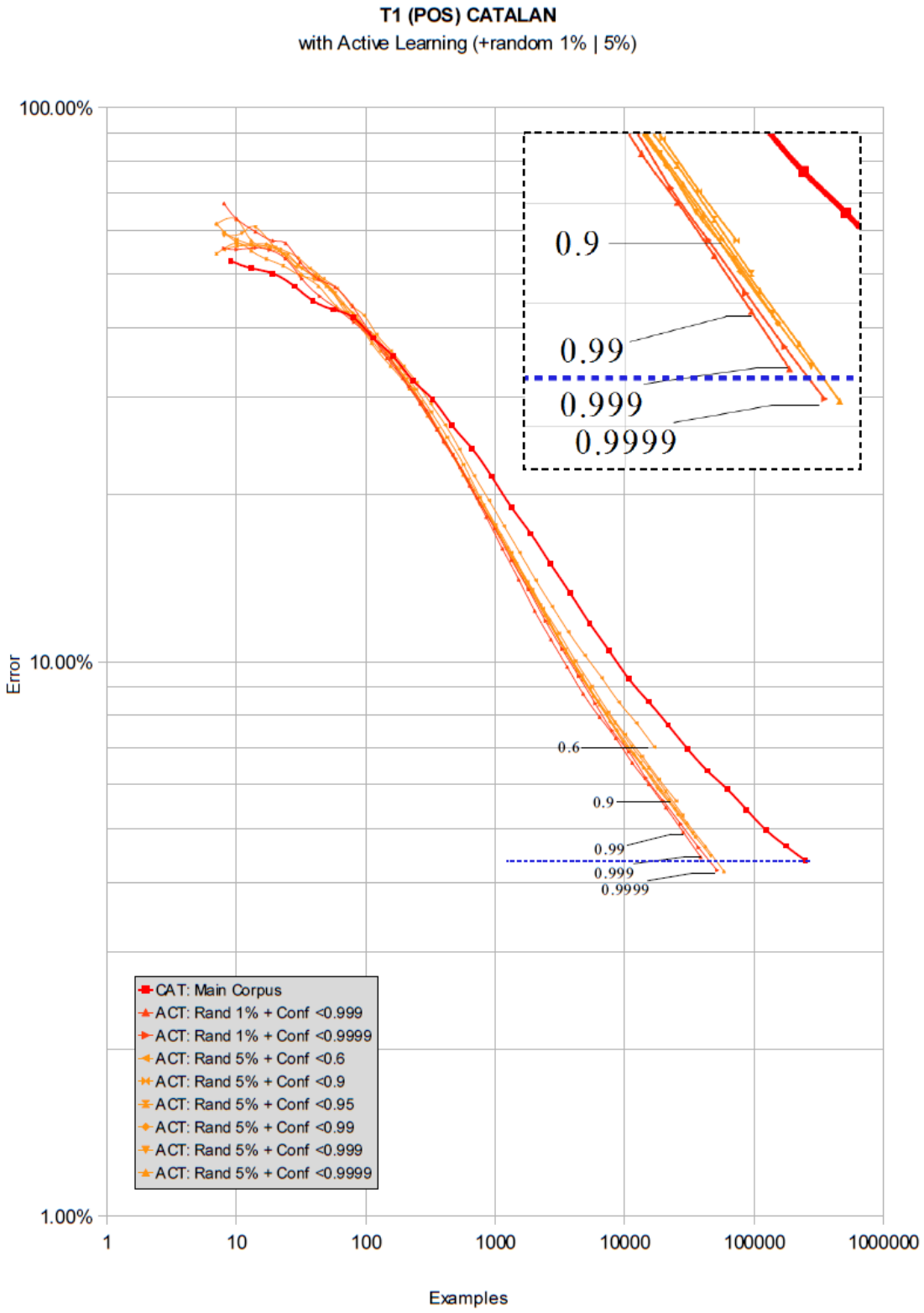

FIG. 94: Evolució de l'error de T1(POS) amb entrenament actiu per *Rand* d'1% i 5%.

EVOLUCIÓ DE T2 (ENT)

El gràfic de la **[Fig. 95]** mostra l'evolució de l'error de la tasca T2 entrenada directament amb el corpus català, línia vermella, juntament amb els dels entrenaments actius amb mostreig aleatori, línies taronges.

S'observa també com en tots els casos l'entrenament actiu assoleix l'error de l'entrenament de referència amb menys quantitat d'exemples. I com, per un mateix llindar de `Confidence`, el mostreig més baix (1%) obté millors resultats. Curiosament el llindar òptim s'obté per un valor de 0,99, i un augment d'aquest llindar perjudica el resultat tot i que continua sent millor que el de l'entrenament de referència. Probablement, en tasques amb certa dificultat, exigir un llindar de certesa més alt de la que el model pot obtenir, suposa que el classificador seleccioni la majoria d'exemples i, per tant, que l'error es redueixi més lentament.

[Espai intencionadament en blanc per alinear el text amb les figures.]

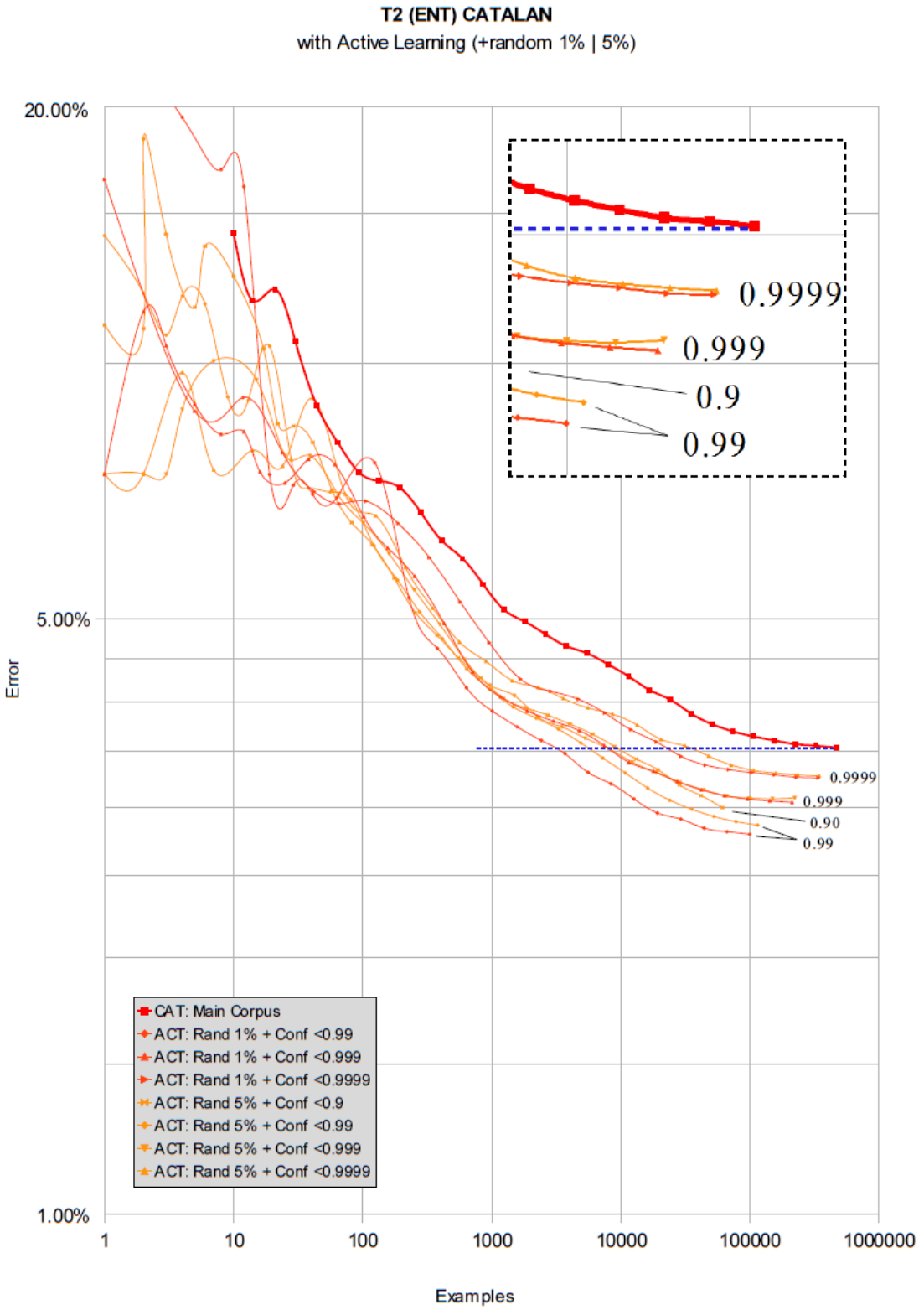

FIG. 95: Evolució de l'error de T2(ENT) amb entrenament actiu per *Rand* d'1% i 5%.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 96]** mostra l'evolució de l'error de la tasca T3 entrenada directament amb el corpus anglès, línia vermella, juntament amb els dels entrenaments actius juntament amb mostreig aleatori, línies taronges.

Com abans, els resultats més eficients s'obtenen amb mostreigs baixos i un llindar del 0,99. Un llindar inferior descarta una quantitat massa gran d'exemples i esgota el corpus abans d'haver observat prou casos. Per contra, probablement degut a la saturació del model, llindars de certesa massa elevats tampoc són eficients ja que utilitza més exemples sense que això suposi una disminució de l'error.

[Espai intencionadament en blanc per alinear el text amb les figures.]

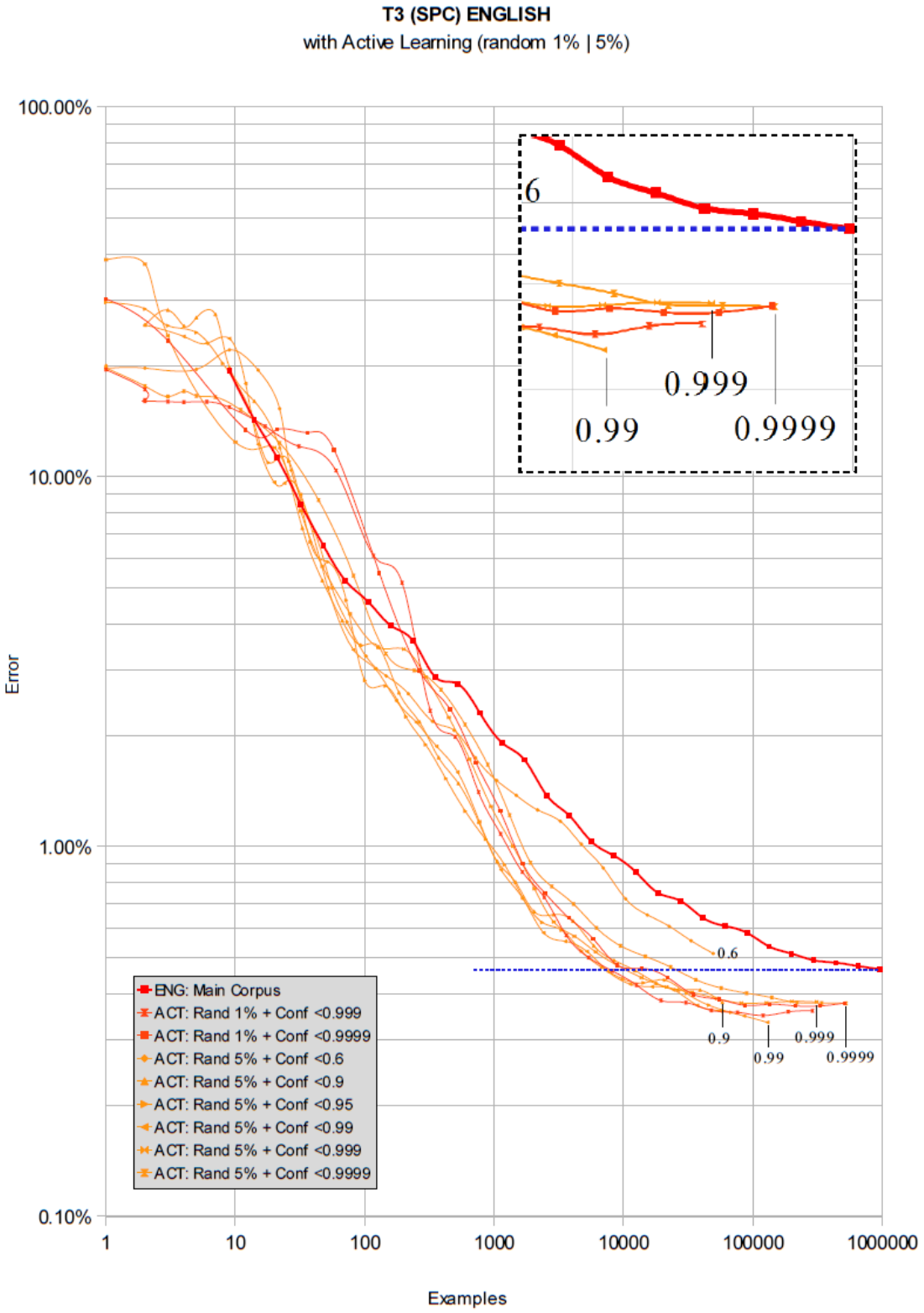

FIG. 96: Evolució de l'error de T3(SPC) amb entrenament actiu per *Rand* d'1% i 5%.

EVOLUCIÓ DE T5 (BRK)

El gràfic de la **[Fig. 97]** mostra l'evolució de l'error de la tasca T5 entrenada directament amb el corpus anglès, línia vermella, juntament amb els dels entrenaments actius amb mostreig aleatori, línies taronges.

En aquesta tasca, on el nombre d'exemples disponibles és molt alt en relació a la dificultat de la tasca, els beneficis de l'entrenament actiu es fan evidents. Amb només uns milers d'exemples el classificador assoleix un error equivalent al de l'entrenament estàndard amb 1 milió d'exemples, cosa que suggereix una elevada redundància en el corpus.

També s'observa com les corbes amb un `Rand` de l'1% acceleren l'entrenament i obtenen corbes de més pendent, ja que el mostreig aleatori representa un terra en l'eficiència obtinguda per l'entrenament actiu. A més, per un mateix mostreig, un llindar de certesa més elevat també redueix l'error final. Probablement el fet d'obtenir els millors resultats amb una `Confidence` del 0,9999 està relacionat amb la facilitat de la tasca. Una tasca on el classificador assoleix errors de l'ordre de 0,02% permet definir llindars complementaris del mateix ordre de magnitud, en aquest cas, del 99,99%.

[Espai intencionadament en blanc per alinear el text amb les figures.]

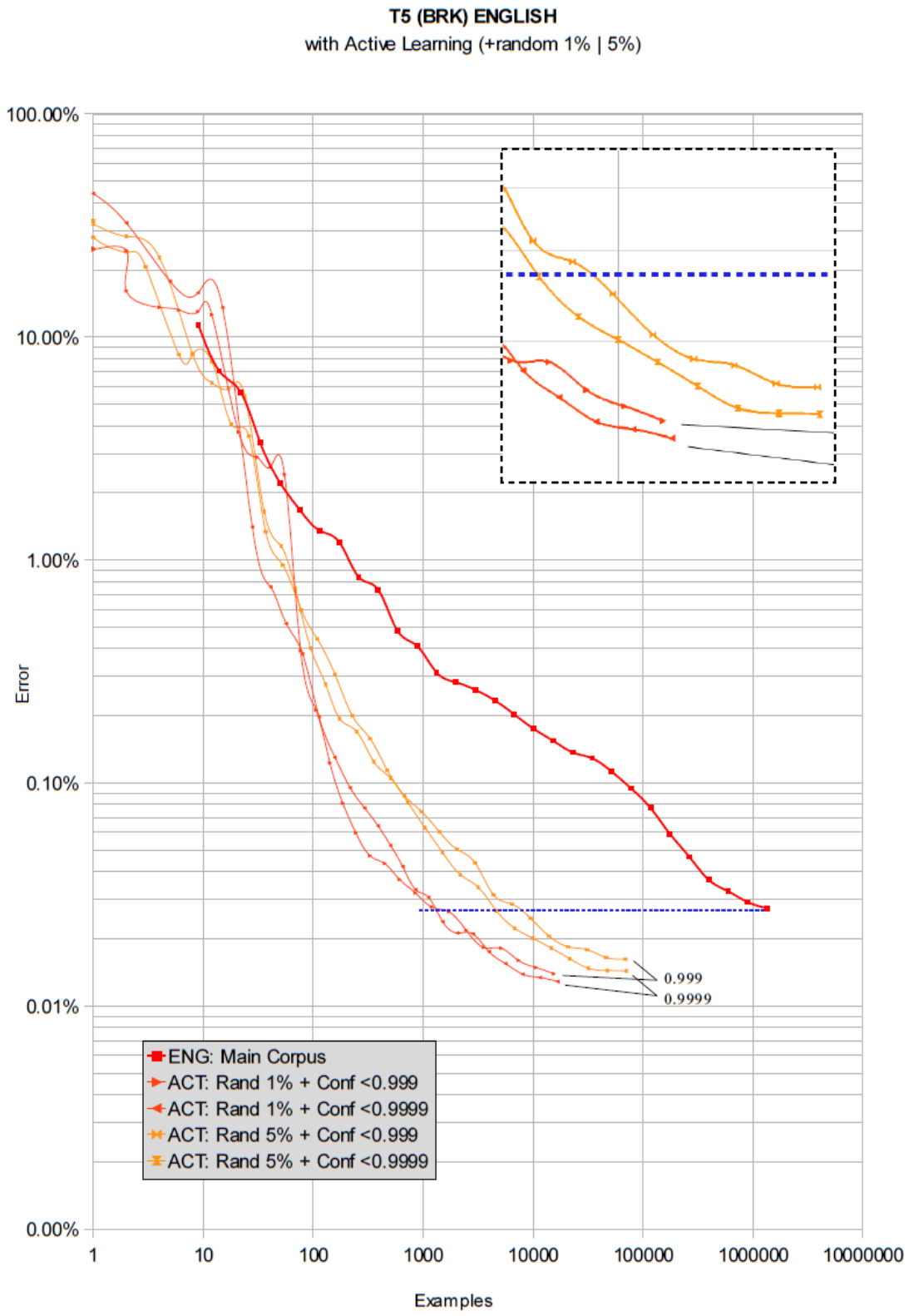

FIG. 97: Evolució de l'error de T5(BRK) amb entrenament actiu per *Rand* d'1% i 5%.

13.5 PARÀMETRE *FIRST*

Una altra forma d'aconseguir que el model disposi de prou informació per discriminar els exemples informatius dels redundants consisteix a endarrerir l'inici de l'entrenament actiu una determinada quantitat d'exemples.

És a dir, s'inicia un entrenament incremental estàndard en el qual es mostren els primers N exemples amb les seves les etiquetes, a partir d'aquest punt, en que el classificador ha induït un model preliminar, s'inicia la selecció d'exemples descartant aquells que considera etiquetables amb prou certesa. En els experiments realitzats la quantitat d'exemples inicials està controlat per un paràmetre que s'ha anomenat *First*.

En els següents apartats d'aquesta secció es mostren les corbes d'avaluació de diferents experiments, per a cada una de les tasques de referència, on s'han combinat diferents valors del nombre d'exemples inicials `First` i del llindar de `Confidence`.

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 98]** mostra l'evolució de l'error de la tasca T1 entrenada directament amb el corpus català, línia vermella, juntament amb els dels entrenaments actius, línies verdes.

Es pot observar clarament com les línies segueixen el mateix camí que l'entrenament estàndard fins al moment on s'inicia l'entrenament actiu (a 100 i 1.000 exemples), moment on se separen i les línies verdes acceleren l'entrenament. També és interesant veure com, independentment de quan s'inicia la selecció d'exemples, tots dos grups de línies acaben convergint. De manera que l'únic paràmetre que influeix en el resultat final és el llindar de certesa utilitzat.

L'error de l'entrenament de referència queda superat per qualsevol de les corbes amb el llindar de `Confidence` suficientment alt, en aquest cas per a 0,999 i 0,9999.

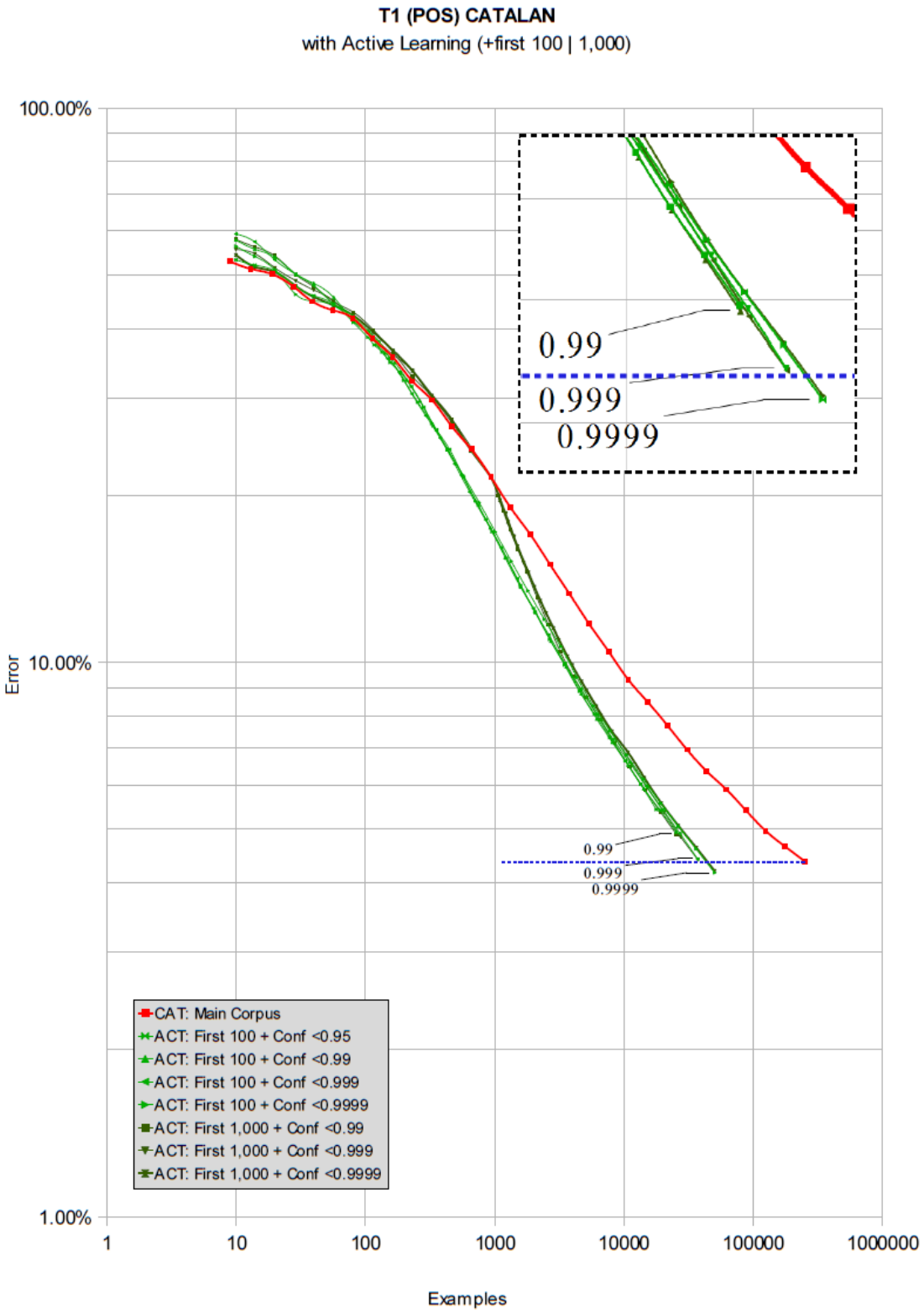

FIG. 98: Evolució de l'error de T1(POS) amb entrenament actiu per *First* de 100 i 1.000.

EVOLUCIÓ DE T2 (ENT)

El gràfic de la **[Fig. 99]** mostra l'evolució de l'error de la tasca T2 entrenada directament amb el corpus català, línia vermella, juntament amb els dels entrenaments actius, línies verdes.

Els resultats són molt semblants als de la tasca anterior: el valor de `First` determina el punt on s'accelera l'entrenament i comencen a separar-se les corbes, però no afecta l'error assolit al final de l'entrenament, ja que les corbes acaben solapant-se.

De la mateixa manera, totes les corbes superen l'error de referència i l'error final ve determinat pel llindar de `Confidence`. De la mateixa manera que succeïa amb el mostreig aleatori, existeix un llindar òptim (0,99) que en cas de superar-se empitjora els resultats.

[Espai intencionadament en blanc per alinear el text amb les figures.]

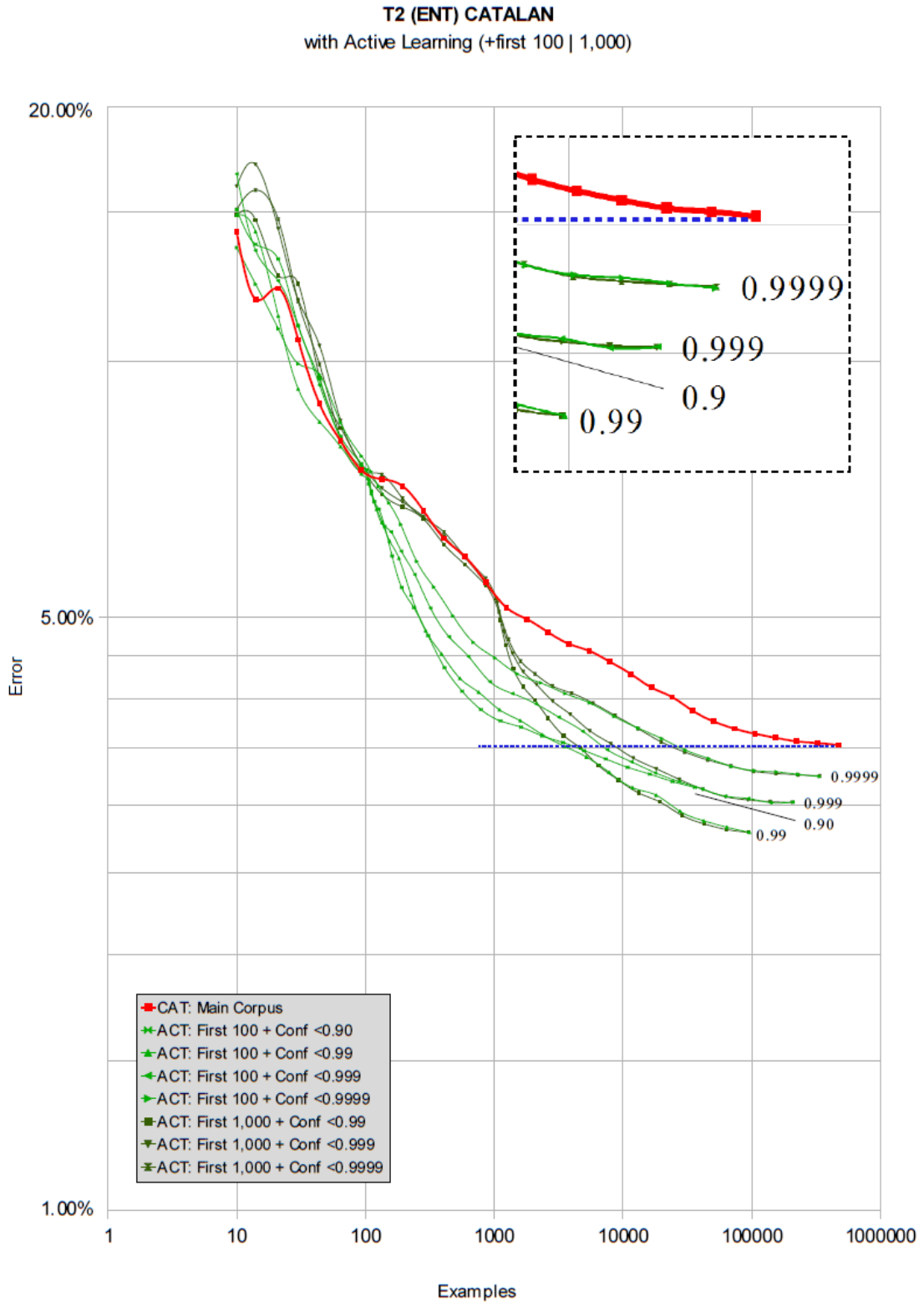

FIG. 99: Evolució de l'error de T2(ENT) amb entrenament actiu per *First* de 100 i 1.000.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 100]** mostra l'evolució de l'error de la tasca T3 entrenada directament amb el corpus anglès, línia vermella, juntament amb els dels entrenaments actius, línies verdes.

El comportament que s'observa és l'esperat: l'inici de l'entrenament actiu marca l'acceleració de l'entrenament, la majoria de línies verdes arriben fins a errors similars i, de la mateixa manera que veiem amb el mostreig aleatori, a partir d'un determinat valor del llindar de `Confidence` no s'observen millores a causa de la saturació del model per haver arribat al terra de la tasca.

[Espai intencionadament en blanc per alinear el text amb les figures.]

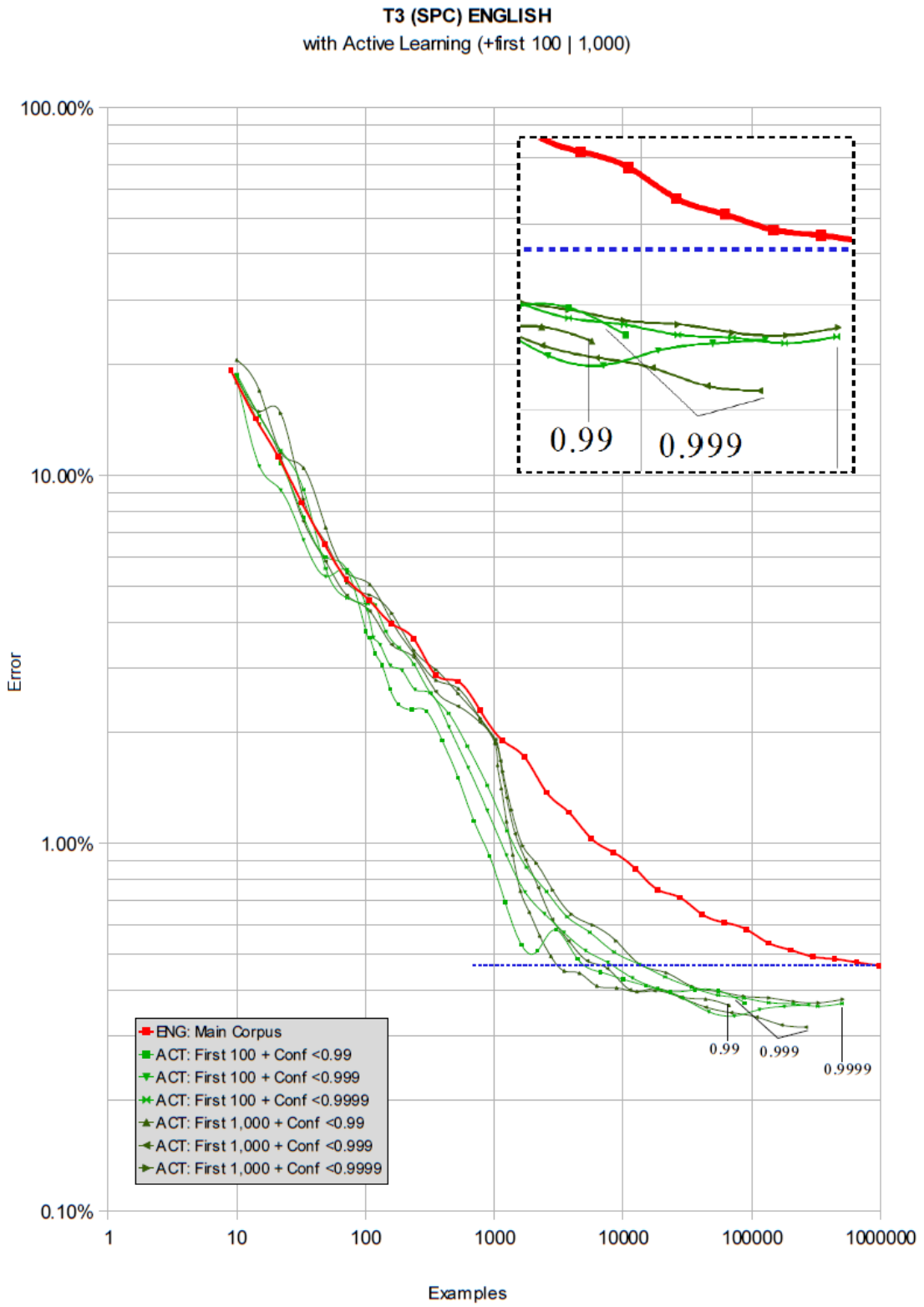

FIG. 100: Evolució de l'error de T3(SPC) amb entrenament actiu per *First* de 100 i 1.000.

EVOLUCIÓ DE T5 (BRK)

El gràfic de la **[Fig. 101]** mostra l'evolució de l'error de la tasca T5 entrenada directament amb el corpus anglès, línia vermella, juntament amb els dels entrenaments actius, línies verdes.

Tot i que els resultats qualitatius són els mateixos, en aquest cas els resultats quantitatius són molt més marcats. Probablement a causa de l'elevada redundància del corpus sintètic, la corba d'avaluació cau pràcticament en vertical una vegada s'inicia l'aprenentatge actiu. Tant les corbes amb `First` igual a 100 com igual a 1.000, acaben superant l'entrenament de referència amb el primer miler d'exemples i reduint l'error a menys de la meitat poc després.

[Espai intencionadament en blanc per alinear el text amb les figures.]

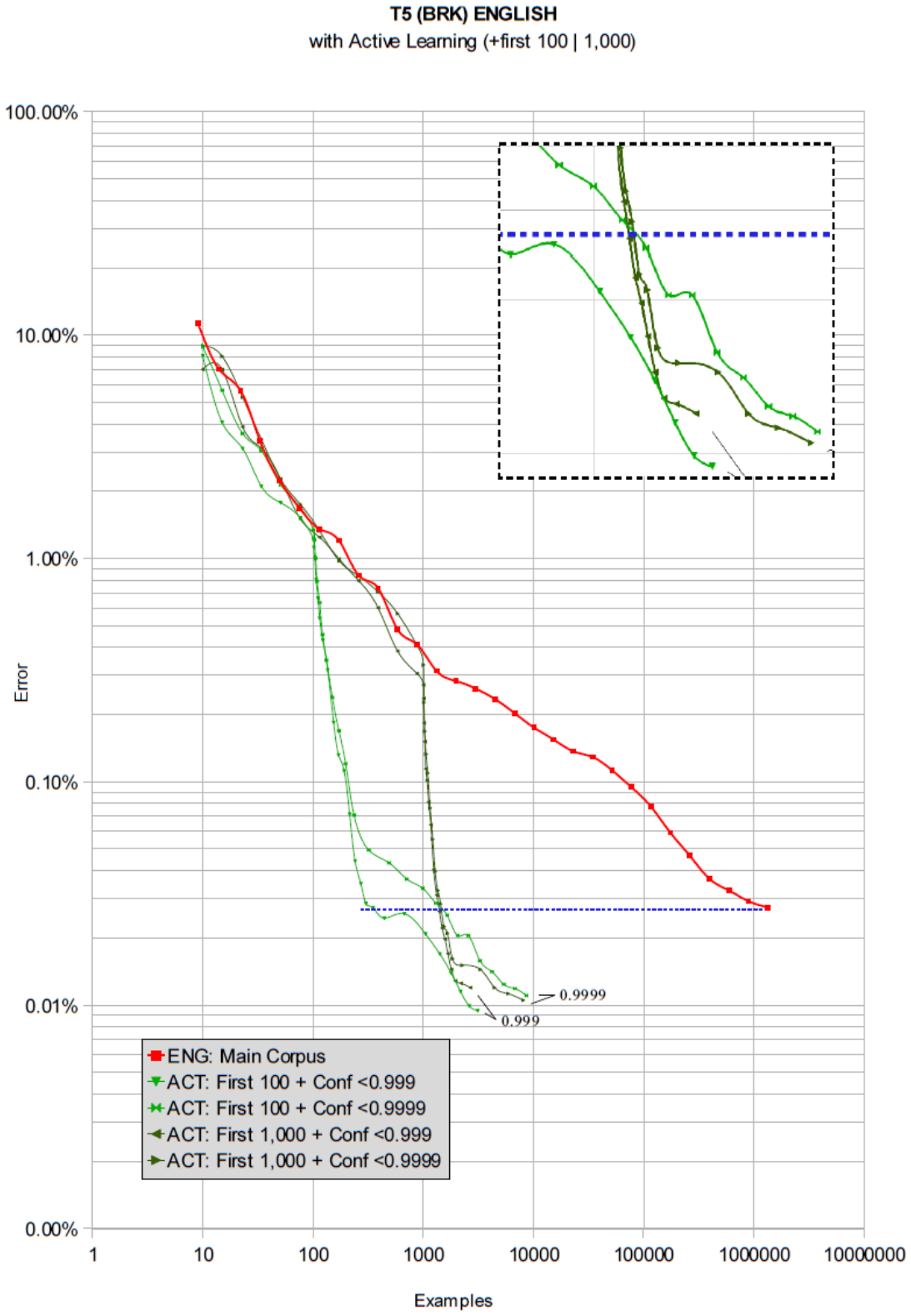

FIG. 101: Evolució de l'error de T5(BRK) amb entrenament actiu per *First* de 100 i 1.000.

13.6 COMPARATIVES

En les seccions precedents s'han mostrat les corbes d'avaluació de més de 50 experiments, i tot i que aquestes corbes són imprescindibles per obtenir una idea general sobre l'evolució de l'entrenament, no en faciliten la comparació quantitativa ni la presa de decisions.

Així doncs, en aquesta secció s'han volgut sintetitzar els resultats centrant-se en els valors finals: error i nombre d'exemples. Però tenint en compte que els objectius del treball són determinar si és possible obtenir classificadors més eficients que assoleixin un error comparable, ens hem centrat en aquestes dues dades: l'error assolit i el tant per cent d'exemples utilitzats o ASR (*Annotated Sample Ratio*).

L'ASR és l'índex utilitzat en aquest treball per mesurar l'eficiència d'un entrenament actiu, representa el tant per cent d'exemples seleccionats pel classificador, o la proporció entre els exemples etiquetats utilitzats per entrenar el model respecte el total d'exemples disponibles. La mesura és equivalent al que a **[Sculley, 2007]** anomenen *Sampling Rate*.

Les gràfiques combinen aquests valors en un eix de coordenades per comparar-los amb l'entrenament de referència (quadrat vermell) de manera que la seva interpretació sigui més senzilla: a) els punts a l'esquerra de la referència són models més eficients que han utilitzat menys exemples, i b) els punts per sota de la referència són models millors que han assolit errors inferiors.

EVOLUCIÓ DE T1 (POS)

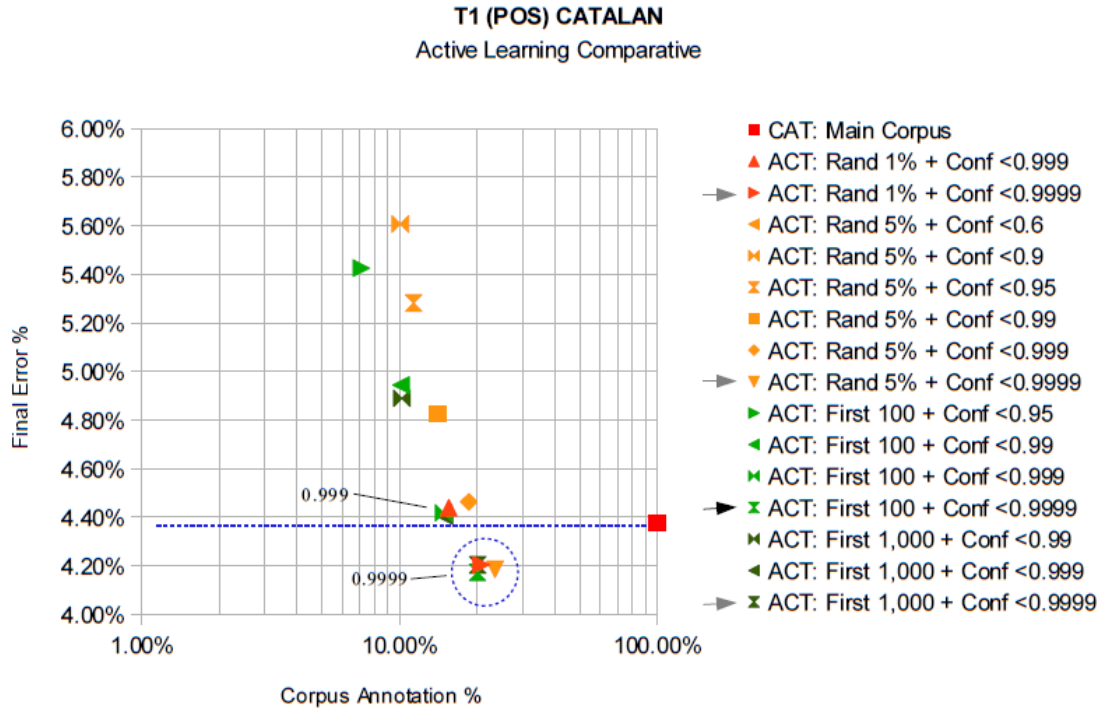

FIG. 102: Comparativa dels errors finals de T1(POS) amb entrenament actiu.

Aquest gràfic **[Fig. 102]** mostra els resultats al final dels entrenaments de totes les variants experimentades per la tasca T1. El cercle discontinu i les fletxes de la llegenda assenyalen els millors resultats.

Es pot observar com independentment de la utilització d'un mostreig aleatori o l'endarreriment de l'entrenament actiu, l'error final està determinat pel llindar de certesa utilitzat. Amb un valor de 0,999 l'error obtingut és similar al de referència, però utilitzant un 15% dels exemples, i amb un valor de 0,9999, cercle discontinu, l'error obtingut és significativament inferior però utilitzant un 20% dels exemples.

EVOLUCIÓ DE T2 (ENT)

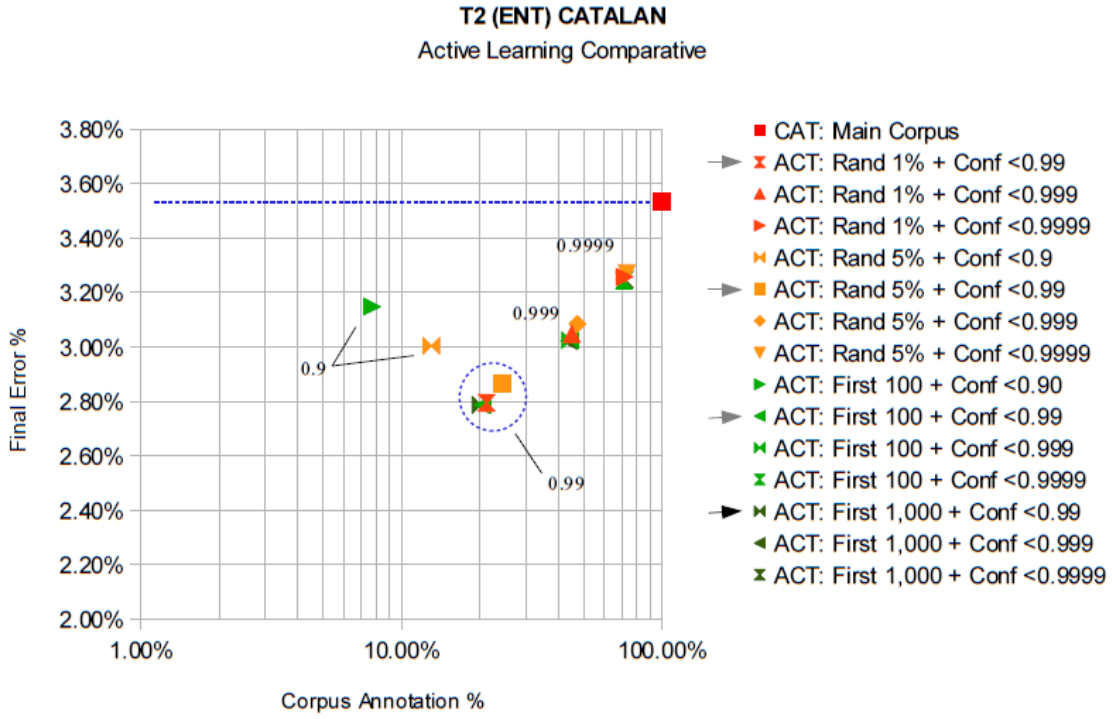

FIG. 103: Comparativa dels errors finals de T2(ENT) amb entrenament actiu.

Els resultats de la tasca T2 es mostren a la **[Fig. 103]**, on es continua observant com la principal variable que afecta els resultats és el llindar de certesa. Es confirma l'anàlisi feta a les proves de `Rand` i `First` per a aquesta tasca, en les quals existeix un valor de `Confidence` òptim (0,99) a partir del qual l'error torna a augmentar per demanar una certesa massa elevada en relació a la dificultat o soroll de la tasca. En qualsevol cas, tots els resultats obtenen un error inferior al de l'entrenament de referència i fàcilment s'assoleixen eficiències d'entre el 10% i el 20%.

EVOLUCIÓ DE T3 (SPC)

Al gràfic següent **[Fig. 104]**, amb les dades dels experiments de T3, es pot observar com la saturació del model fa que hi hagi diferents variants d'entrenament amb errors similars, tots ells millors que l'error del model de referència, però amb importants diferències

d'eficiència. Tot i que, per exemple, un dels models assoleix un error un 20% inferior al de referència utilitzant únicament un 6% dels exemples, l'error mínim és un 30% inferior al de referència però utilitzant un 27% dels exemples.

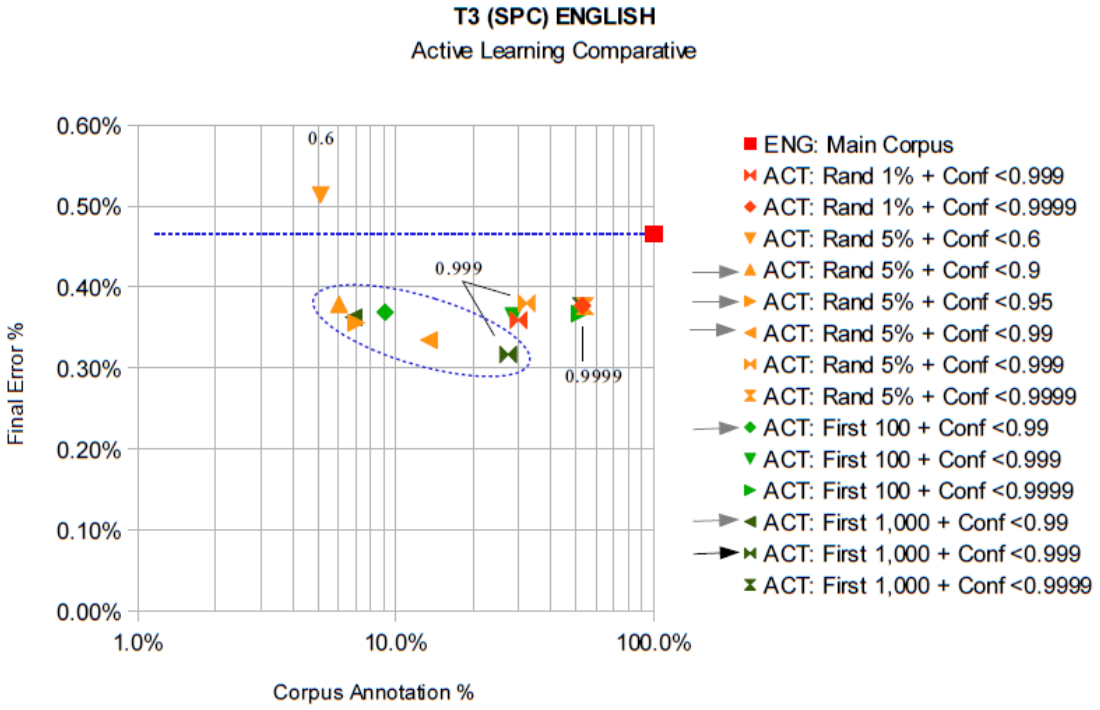

FIG. 104: Comparativa dels errors finals de T3(SPC) amb entrenament actiu.

EVOLUCIÓ DE T5 (BRK)

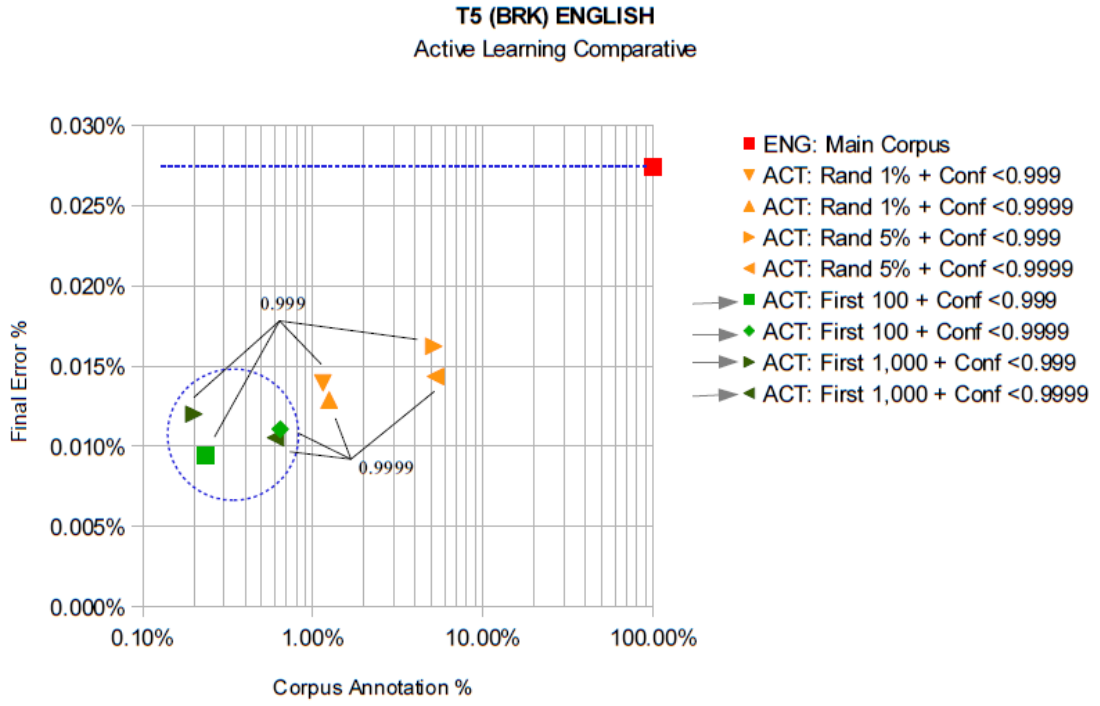

FIG. 105: Comparativa dels errors finals de T5(BRK) amb entrenament actiu.

Finalment, a el gràfic de la tasca T5 **[Fig. 105]**, una tasca on les dades d'entrenament presenten una gran redundància, s'observa una clara preferència per l'entrenament actiu endarrerit sobre el mostreig aleatori que, com ja s'ha dit, representa un límit inferior en l'eficiència, com demostren els triangles grocs just a la dreta de l'1% i 5%. Els millors resultats els obtenen els entrenament endarrerits (`First` de 100 i 1.000) amb errors de menys de la meitat del de referència i eficiències de fins al 0,2%, això representa la utilització de només 1 de cada 500 exemples disponibles.

13.7 EVOLUCIÓ DELS MODELS

En aquesta secció es mostren les corbes d'avaluació obtingudes per les quatre tasques de referència, i en cada una d'elles se superposen les corbes obtingudes pels entrenaments actius amb els paràmetres seleccionats. En els apartats següents es mostren els resultats finals i s'analitzen algunes conclusions que poden extreure's de l'observació d'aquestes gràfiques:

[Espai intencionadament en blanc per alinear el text amb les figures.]

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 106]** mostra l'evolució de l'error de la tasca T1 entrenada directament amb el corpus principal en català, línia de color vermell, i algunes variants d'entrenament actiu, línies verdes i taronges.

La principal característica de les corbes d'avaluació de l'entrenament actiu és que comencen amb un error equivalent al de l'entrenament estàndard però l'error disminueix més ràpidament utilitzant menys exemples. Recordem que el pendent d'aquesta corba pot interpretar-se com la velocitat de reducció de l'error.

Independentment de la variant d'entrenament actiu utilitzada (`Rand` o `First`) els trams finals de les línies recorren camins semblants, obtenint-se resultats lleugerament millors endarrerint l'entrenament actiu 100 exemples. Tot i això, tots ells superen l'error de referència utilitzant al voltant del 20% dels exemples **[Taula 52]**, és a dir, requerint l'anotació de 5 vergades menys exemples.

Best Training Parameters	Error			ASR
	Min	Max	Mean	
T1-STD CAT (reference)	*4.20%*	*4.52%*	***4.38%***	***100.0%***
T1-ACT Rand 1% + Conf<0.9999	4.07%	4.33%	**4.20%**	**20.72%**
T1-ACT Rand 5% + Conf<0.9999	3.99%	4.50%	**4.19%**	**23.46%**
T1-ACT First 100 + Conf<0.9999	4.06%	4.32%	**4.17%**	**20.06%**
T1-ACT First 1.000 + Conf<0.9999	3.89%	4.42%	**4.20%**	**20.08%**

TAULA 52: Resultats de les proves amb entrenament actiu per la T1 (POS).

[Espai intencionadament en blanc per alinear el text amb les figures.]

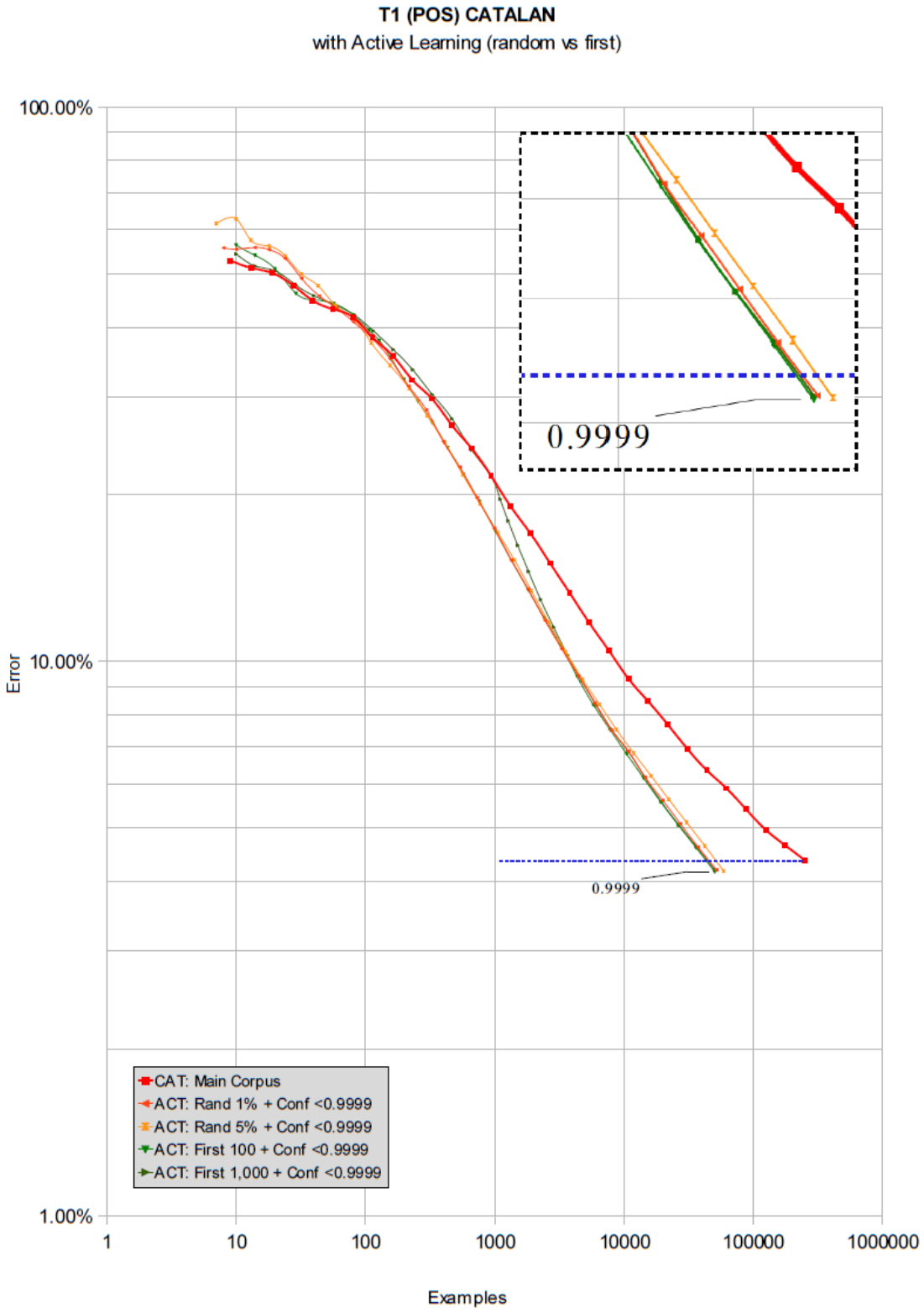

FIG. 106: Evolució de l'error de T1(POS) pels millors casos amb entrenament actiu.

EVOLUCIÓ DE T2 (ENT)

El gràfic de la **[Fig. 107]** mostra l'evolució de l'error de la tasca T2 entrenada directament amb el corpus principal en català, línia de color vermell, i algunes variants d'entrenament actiu, línies verdes i taronges.

En aquest cas les corbes són inicialment més inestables, degut al desequilibri del corpus, però a partir dels centenars d'exemples s'estabilitzen i a partir d'uns 3.000 exemples totes les corbes superen el model de referència entrenat amb 100 vegades més exemples.

Per a aquesta tasca els millors resultats **[Taula 53]** s'obtenen amb un `Rand` de l'1% o amb qualsevol de les dues variants `First`. En aquests casos tots tres arriben al mateix punt final: un error del 2,79% (un 80% de l'error de referència) i una eficiència del 20%-21%, és a dir, que redueix el temps i cost de l'anotació al voltant de 5 vegades.

Best Training Parameters	Error			ASR
	Min	Max	**Mean**	
T2-STD CAT (reference)	*3.36%*	*3.65%*	***3.53%***	***100%***
T2-ACT Rand 1% + Conf<0.99	2.69%	2.88%	**2.79%**	**21.18%**
T2-ACT Rand 5% + Conf<0.99	2.72%	3.00%	**2.86%**	**24.41%**
T2-ACT First 100 + Conf<0.99	2.71%	2.85%	**2.79%**	**20.48%**
T2-ACT First 1.000 + Conf<0.99	2.65%	2.93%	**2.79%**	**20.10%**

TAULA 53: Resultats de les proves amb entrenament actiu per la T2 (ENT).

[Espai intencionadament en blanc per alinear el text amb les figures.]

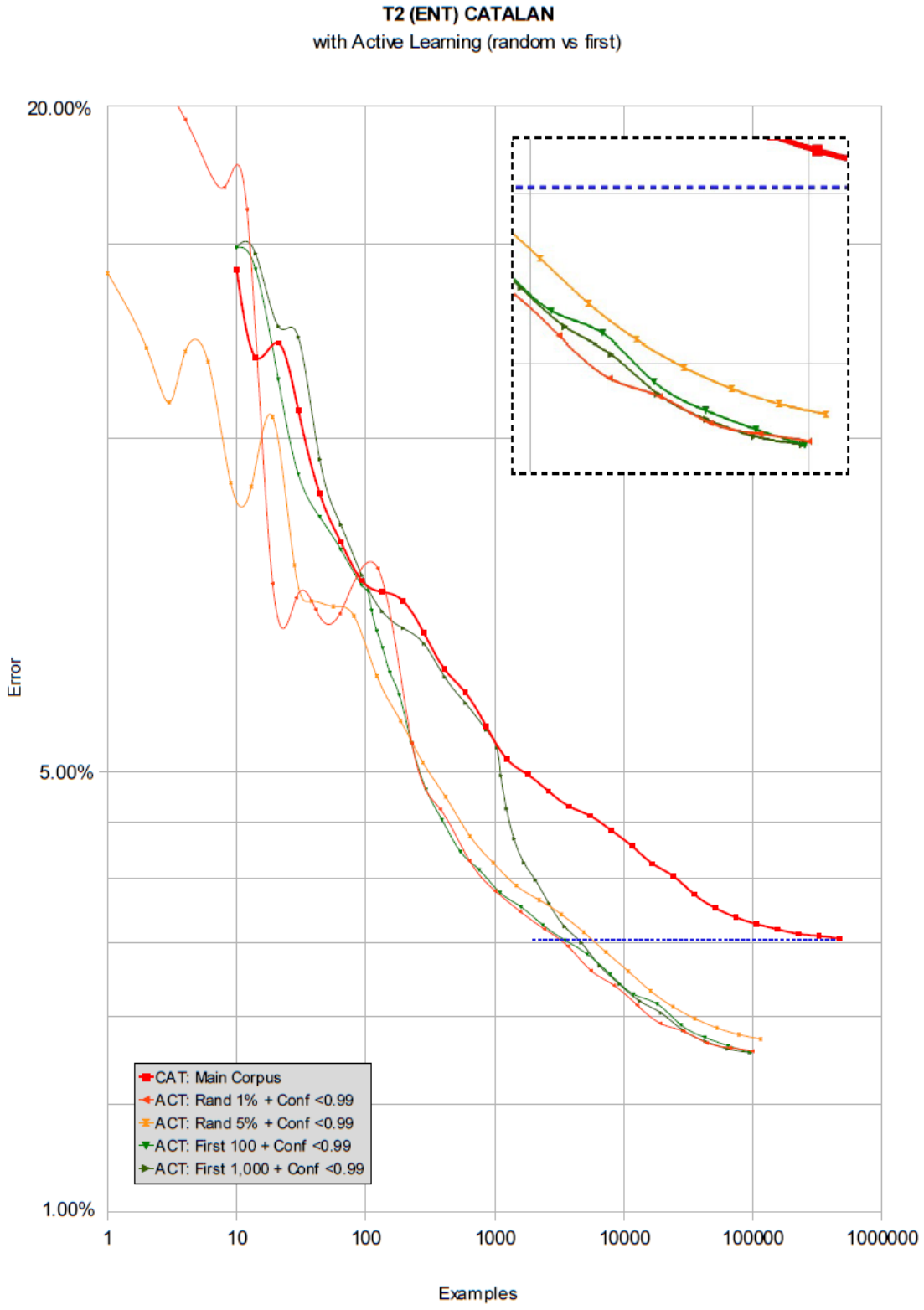

FIG. 107: Evolució de l'error de T2(ENT) pels millors casos amb entrenament actiu.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 108]** mostra l'evolució de l'error de la tasca T3 entrenada directament amb el corpus principal en anglès, línia de color vermell, i algunes variants d'entrenament actiu, línies verdes i taronges.

Els resultats observats segueixen en la mateixa línia, a mesura que l'entrenament avança les corbes es van separant de la de referència i redueixen l'error a més velocitat. Tot i que presenten petites variacions, el conjunt segueix el mateix patró, i assoleix l'error de referència amb 100 vegades menys exemples.

Si es continua l'entrenament, tot i que s'observen símptomes de saturació, es redueix l'error de referència al voltant d'un 25% assolint un ASR d'entre un 6% i un 10% dels exemples **[Taula 54]**, és a dir entre 10 i 15 vegades menys exemples.

Best Training Parameters	Error			ASR
	Min	Max	Mean	
T3-STD ENG (reference)	*0.439%*	*0.489%*	***0.466%***	***100%***
T3-ACT Rand 5% + Conf<0.9	0.352%	0.409%	**0.379%**	**6.0%**
T3-ACT Rand 5% + Conf<0.95	0.316%	0.394%	**0.356%**	**7.0%**
T3-ACT Rand 5% + Conf<0.99	0.308%	0.363%	**0.335%**	**13.4%**
T3-ACT First 100 + Conf<0.99	0.342%	0.405%	**0.369%**	**9.1%**
T3-ACT First 1.000 + Conf<0.99	0.307%	0.404%	**0.363%**	**6.9%**
T3-ACT First 1.000 + Conf<0.999	0.266%	0.354%	**0.317%**	**27.3%**

TAULA 54: Resultats de les proves amb entrenament actiu per la T3 (SPC).

[Espai intencionadament en blanc per alinear el text amb les figures.]

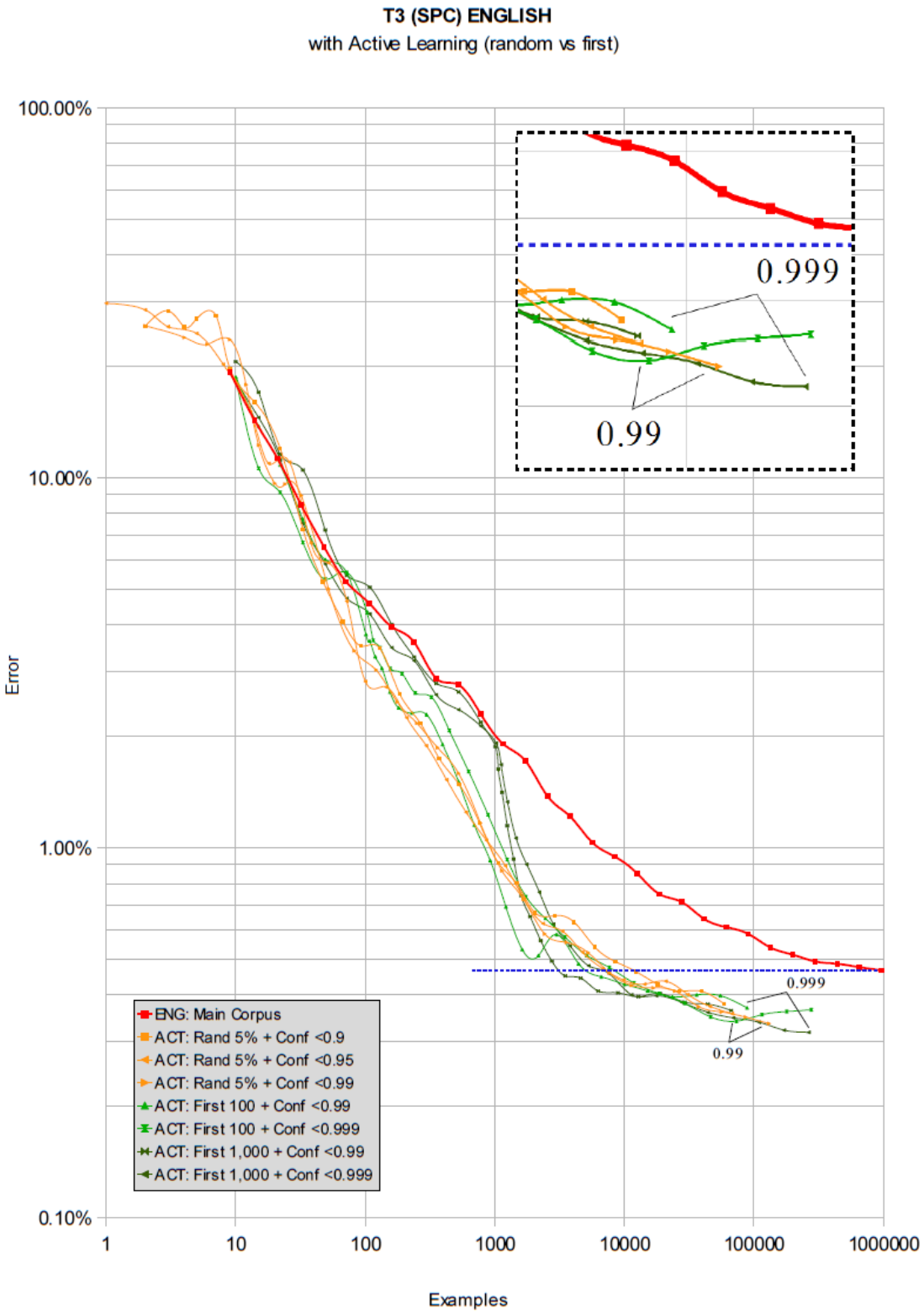

FIG. 108: Evolució de l'error de T3(SPC) pels millors casos amb entrenament actiu.

EVOLUCIÓ DE T5 (BRK)

El gràfic de la **[Fig. 109]** mostra l'evolució de l'error de la tasca T5 entrenada directament amb el corpus principal en anglès, línia de color vermell, i algunes variants d'entrenament actiu, línies verdes i taronges.

Com s'ha comentat anteriorment, els resultats d'aquesta tasca no són representatius d'una tasca típica a causa de l'elevada redundància que presenta el corpus sintètic, però pot servir de referència per a casos reals on les dades presentin aquest nivell de redundància.

Es pot observar com l'entrenament actiu amb mostreig aleatori, línia taronja, ja s'allunya de la línia de referència molt més ràpidament que en les tasques anteriors, però és amb l'entrenament actiu endarrerit on el canvi de pendent és extrem. La corba de l'entrenament amb un `First` de 100, la caiguda és tan brusca que assoleix l'error de referència amb uns centenars d'exemples seleccionats, que suposen 5.000 vegades menys exemples. Però fins i tot si ens centrem en l'error final, els dos entrenaments `First` redueixen l'error final 2 i 3 vegades havent utilitzat entre 400 i 500 vegades menys exemples. Un exemple clar de la capacitat de l'entrenament actiu per separar el gra de la palla en corpus altament redundats.

Best Training Parameters	Error			ASR
	Min	Max	Mean	
T5-STD ENG (reference)	*0.021%*	*0.036%*	***0.027%***	***100.0%***
T5-ACT Rand 1% + Conf<0.9999	0.008%	0.019%	**0.013%**	**1.25%**
T5-ACT First 100 + Conf<0.999	0.004%	0.013%	**0.009%**	**0.23%**
T5-ACT First 1.000 + Conf<0.999	0.007%	0.016%	**0.012%**	**0.20%**

TAULA 55: Resultats de les proves amb entrenament actiu per la T5 (BRK).

[Espai intencionadament en blanc per alinear el text amb les figures.]

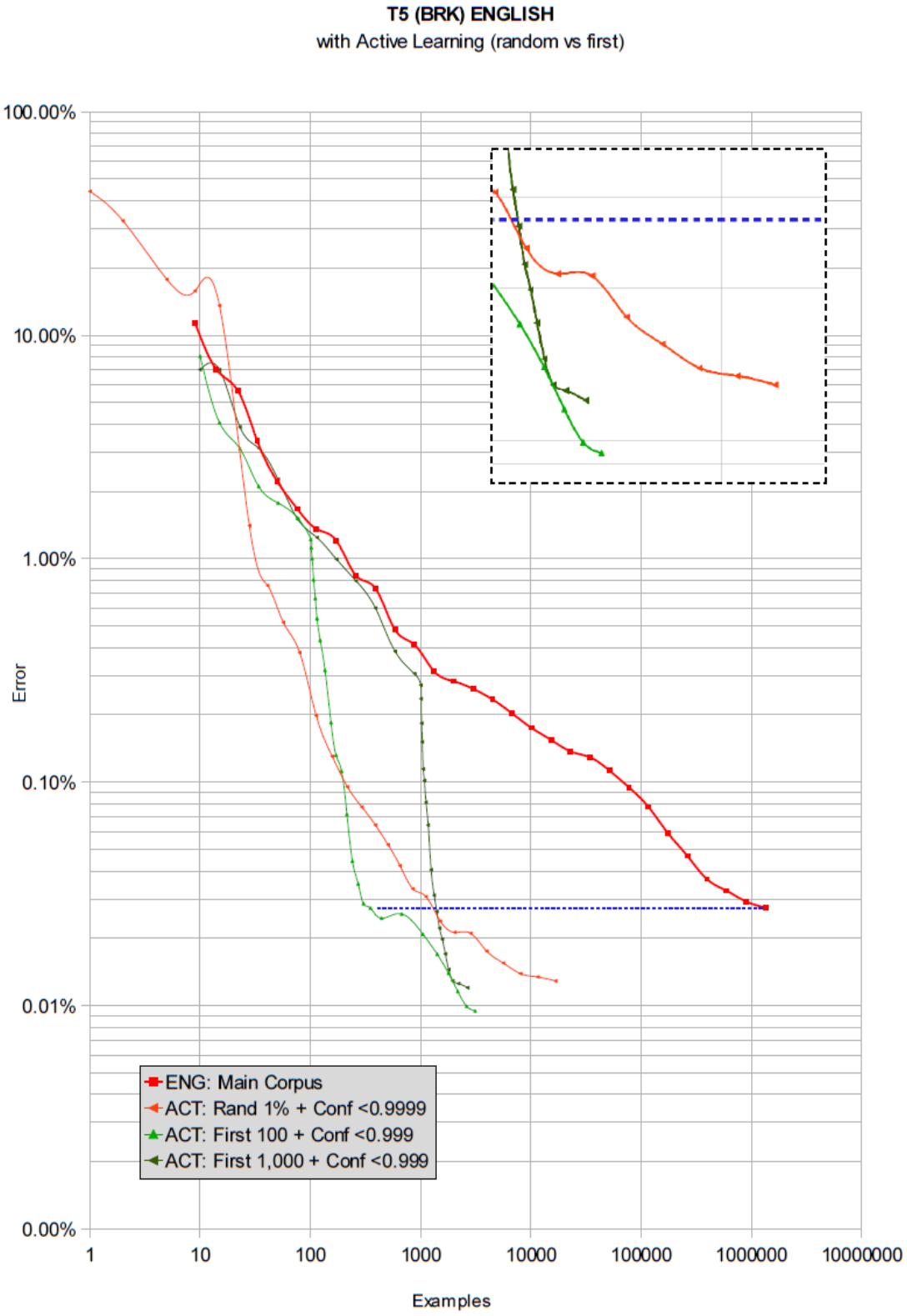

FIG. 109: Evolució de l'error de T5(BRK) pels millors casos amb entrenament actiu.

13.8 CONCLUSIONS

En aquest capítol s'ha descrit la utilització de l'entrenament actiu com una tècnica per a accelerar l'entrenament i induir models d'una manera més eficient. També s'han presentat els resultats d'aplicar aquesta tècnica a les quatre tasques de referència i se n'ha fet una comparativa intensiva.

Els resultats indiquen sense cap dubte que la utilització de l'entrenament actiu per induir models incrementals permet millorar els resultats de l'entrenament incremental estàndard, especialment en relació a la quantitat d'exemples utilitzats.

Una manera de quantificar aquesta millora és mesurar la quantitat d'exemples necessaris per assolir el mateix error que l'obtingut al finalitzar l'entrenament estàndard. Aquest valor es pot representar com un tant per cent del corpus utilitzat, per conèixer la seva eficiència, o com el factor reductor de temps i cost, si fem el seu invers. A la taula següent **[Taula 56]** es pot veure com la quantitat d'exemples necessaris per igualar l'error de l'entrenament estàndard pot reduir-se entre cinc i alguns centenars de vegades:

Tasks	Corpus Used	Saving Factor
T1(POS)	18%	×5.5
T2(ENT)	0.9%	×100
T3(SPC)	0.3%	×300
T5(BRK)	0.03%	×3.800

TAULA 56: Corpus utilitzat per l'entrenament actiu per a igualar l'error obtingut per l'entrenament estàndard.

En un altre escenari és pot considerar que l'objectiu és minimitzar l'error, aprofitant de la forma més eficient possible el corpus disponible, deixant que l'algorisme seleccioni tants exemples com consideri oportú. En aquest cas, veure la taula següent, sembla que és possible reduir l'error final entre un 5% i un 50%, utilitzant però una fracció dels exemples, amb una factor reductor situat entre cinc i unes poques centenes.

Tasks	Reference Error	Final Error	Improvement	Corpus Used	Saving Factor
T1(POS)	4.38%	4.17%	**-4.8%**	**22%**	×5
T2(ENT)	3.53%	2.79%	**-21%**	**20%**	×5
T3(SPC)	0.466%	0.356%	**-24%**	**7%**	×14
T5(BRK)	0.027%	0.009%	**-67%**	**0.21%**	×470

TAULA 57: Reducció de l'error i corpus utilitzat al finalitzar l'entrenament actiu.

De totes maneres, tot i que els beneficis d'aquesta tècnica són clars, els resultats mostren variacions molt grans segons el cas. Les característiques de cada una de les tasques (dificultat, mida del corpus, mida de l'univers de representació, ...) suggereixen que el factor determinant és el *grau de redundància* del corpus. A més, tot i que en la literatura sobre l'aprenentatge actiu *batch* és habitual posar el focus en la selecció d'exemples informatius **[Huang et al, 2010]**, en l'aprenentatge actiu incremental potser és més correcte plantejar-

ho com una eliminació d'exemples redundants, que tot i ser equivalent ofereix un punt de vista amb altres matisos.

En relació als paràmetres d'entrenament, els resultats dels experiments suggereixen que la variant utilitzada per alimentar el model inicial no proporciona grans diferències a mesura que avança l'entrenament. Ja sigui amb el mostreig aleatori o amb l'endarreriment de l'entrenament actiu, l'error final no queda influenciat pels valors dels paràmetres `First` i `Rand`. Tot i que per a corpus molt redundants, cal tenir en compte que el valor de `Rand` suposa un límit inferior a l'ASR (*Annotated Sampling Ratio*) i per tant pot penalitzar l'eficiència.

Per contra, tots els experiments i totes les tasques apunten clarament a l'elevada sensibilitat de l'error final respecte el paràmetre del llindar de `Confidence`. La majoria de valors de certesa al voltant del 99%, que representa un nivell de "dubte" o error de l'1%, donen bons resultats i supera l'entrenament estàndard. En determinades tasques especialment solubles i amb corpus amb una baixa taxa d'error, es poden assolir errors de classificació molt inferiors augmentant el llindar de `Confidence` a valors superiors a 99,9% o 99,99%.

14 UTILITZACIÓ D'ENTRENAMENT COMBINAT

14.1 INTRODUCCIÓ

En els capítols anteriors s'ha mostrat com el pre-entrenament **[12. Utilització de Pre-Entrenament]** permet reduir l'error inicial del classificador sense afectar l'error final, millora que suposa un desplaçament cap a baix de la part esquerra de les corbes d'entrenament. També s'ha vist com l'entrenament actiu **[13. Utilització d'Entrenament Actiu]** permet accelerar l'entrenament i reduir l'error final, requerint l'anotació de menys exemples, el que gràficament suposa un augment del pendent de la corba d'entrenament.

El dubte que queda és determinar si és possible combinar aquestes dues tècniques i si els beneficis obtinguts individualment seran additius. És a dir, si serà possible reduir l'error en l'etapa inicial i accelerar l'entrenament obtenint un error més baix al llarg de tot l'entrenament. En cas afirmatiu, la corba d'entrenament obtinguda hauria d'allunyar-se de la corba de referència tant a l'inici com al final **[Fig. 110]**, i com més gran sigui la distància millors seran els beneficis: reducció de l'error i reducció del corpus utilitzat.

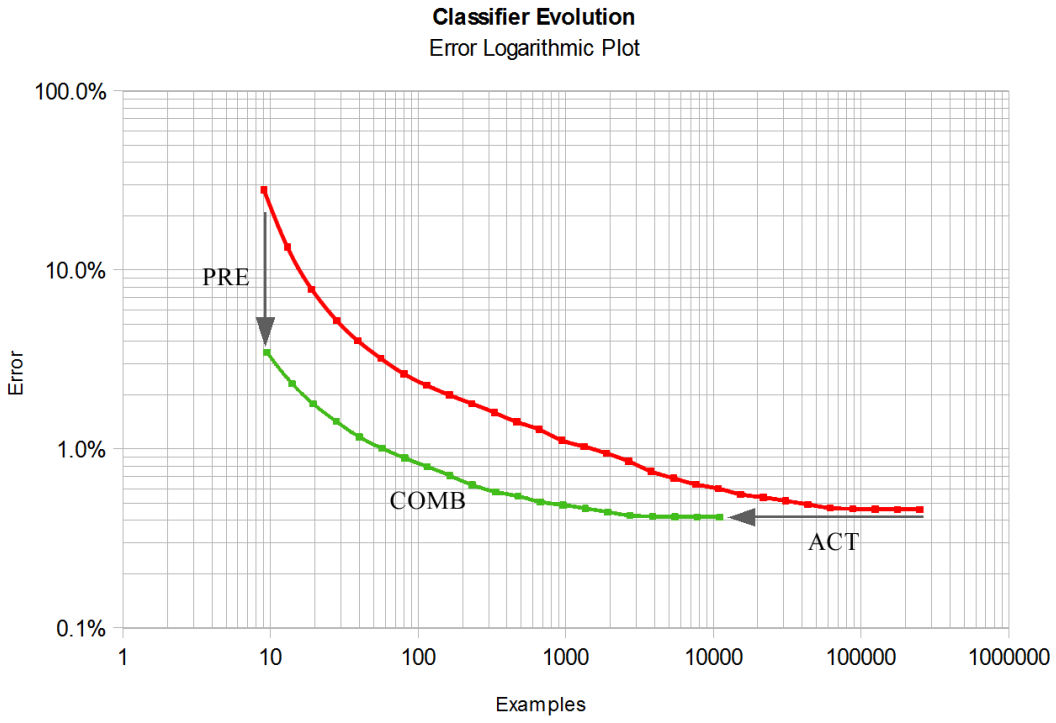

FIG. 110: Efecte combinat teòric del pre-entrenament i de l'entrenament actiu.

A més, el pre-entrenament ofereix un avantatge en ser combinat amb l'entrenament actiu: l'existència inicial d'un model que pot discriminar els exemples informatius. Recordem que en l'entrenament actiu pur és necessari alimentar el model amb un mostreig aleatori constant (`Rand`) o amb un endarreriment de l'entrenament actiu (`First`). Però el fet que el model hagi estat pre-entrenat amb un corpus auxiliar permet realitzar un entrenament actiu amb aquests dos paràmetres amb uns valors de zero.

En aquest capítol s'analitza la combinació d'un pre-entrenament amb un entrenament actiu amb l'objectiu de comprovar si és possible obtenir simultàniament totes dues millores: la reducció de l'error inicial i, al mateix temps, l'acceleració de l'entrenament.

En la propera secció s'avaluen diferents combinacions entre els entrenaments actius seleccionats al capítol anterior i els pre-entrenaments seleccionats prèviament. Tot seguit, es realitza una comparativa exhaustiva entre els diferents resultats per determinar els casos que maximitzen l'eficiència i igualen o milloren l'error final. L'objectiu és determinar les combinacions que proporcionen millors resultats. Una vegada seleccionats els millors casos es comparen les corbes d'avaluació obtingudes a l'entrenament combinat amb les de l'entrenament incremental estàndard, utilitzat com a referència, i es quantifica l'estalvi i la millora obtinguts.

Finalment, a les conclusions, es sintetitzen els resultats mitjançant les taules amb els resultats obtinguts en les quatre tasques de referència. En tots els casos s'obtenen errors finals inferiors als obtinguts amb l'entrenament estàndard i, tot i que les diferents tasques presenten valors d'eficiència molt diversos, aquests errors sempre s'assoleixen utilitzant una fracció del corpus disponible. Per tant, no hi ha dubte que l'entrenament combinat permet obtenir models lleugerament més precisos utilitzant una quantitat molt inferior d'exemples, amb l'avantatge de reduir l'error inicial escorçant l'etapa de transició. De totes maneres, com es veurà en les properes seccions, els resultats no superen els valors absoluts obtinguts per l'entrenament actiu.

[Espai intencionadament en blanc per alinear el text amb les figures.]

[Espai intencionadament en blanc per alinear el text amb les figures.]

14.2 COMBINACIÓ DE TÈCNIQUES

En els següents apartats d'aquesta secció es mostren, per a cada una de les tasques de referència, les corbes d'avaluació dels diferents experiments on s'han combinat les tècniques anteriors i se'n comparen els resultats amb els entrenaments estàndard de referència. A continuació es comenten els resultats obtinguts:

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 111]** mostra l'evolució de l'error de la tasca T1 entrenada directament amb el corpus principal, línia de color vermell, i els resultats de diferents entrenaments combinats.

S'observa com tots els experiments combinats segueixen una trajectòria similar, allunyada de la de l'entrenament estàndard, i que coincideix amb l'efecte combinat teòric predit a l'inici del capítol.

L'únic resultat destacable s'observa clarament a l'alçada dels 5.000 exemples, on es pot observar com el conjunt de corbes sembla agrupar-se en dos feixos diferenciats i que coincideixen amb els experiments pre-entrenats amb el mateix pes. És a dir, tot i que els dos feixos continuen apropant-se fins solapar-se, sembla que en l'entrenament combinat d'aquesta tasca reduir el pes del corpus de pre-entrenament permet obtenir un error lleugerament inferior.

[Espai intencionadament en blanc per alinear el text amb les figures.]

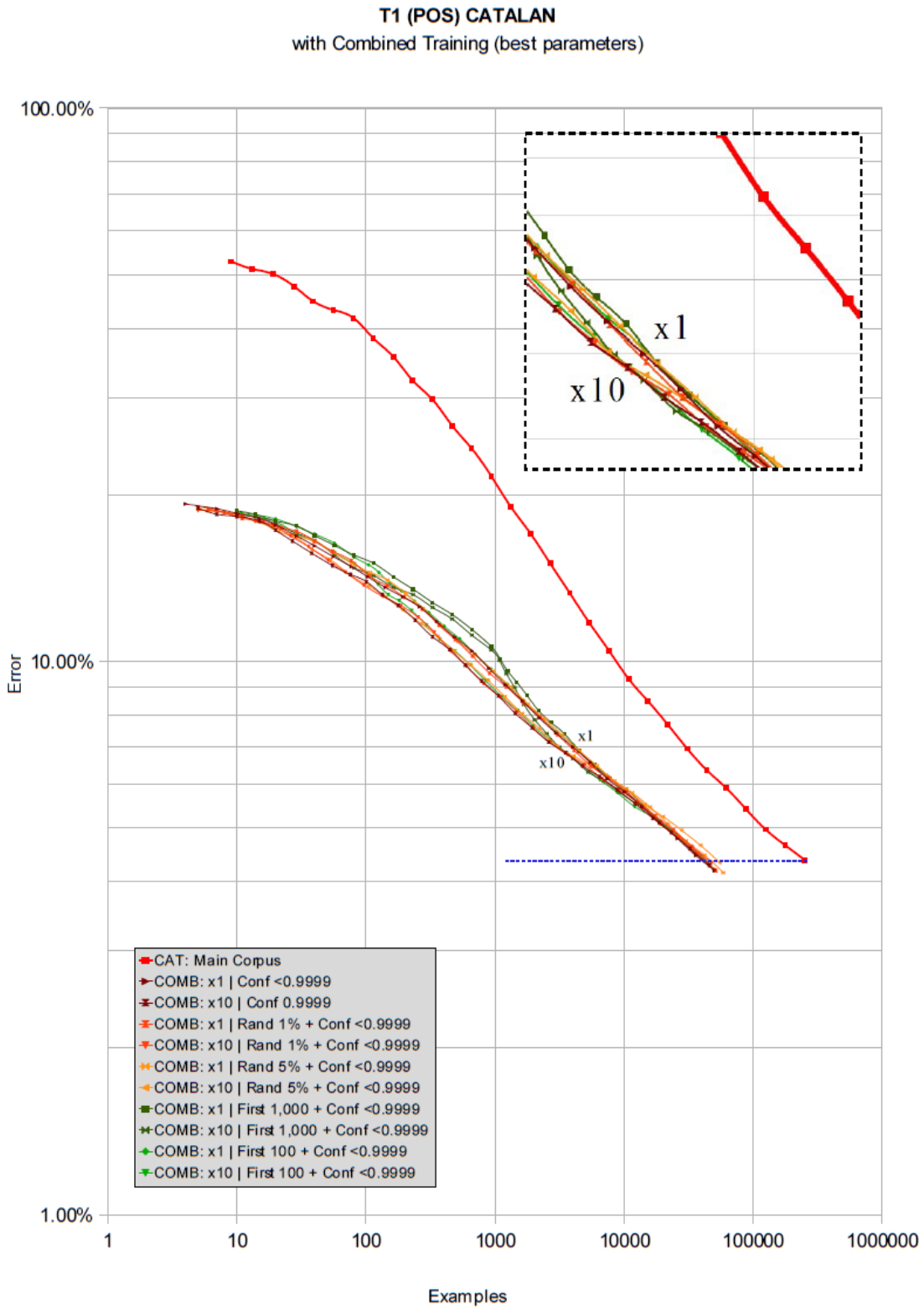

FIG. 111: Evolució de l'error de T1(POS) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T2 (ENT)

El gràfic de la **[Fig. 112]** mostra l'evolució de l'error de la tasca T2 entrenada directament amb el corpus principal, línia de color vermell, i els resultats de diferents entrenaments combinats.

Com ja se sabia, el pre-entrenament a la tasca T2 no redueix especialment l'error inicial, però en qualsevol cas les corbes mostren l'efecte combinat del pre-entrenament i l'entrenament actiu, de manera que l'error és menor al llarg de tot l'entrenament.

De la mateixa manera que a la tasca anterior s'observa com les corbes s'agrupen en dos feixos segons el pes corresponent al pre-entrenament, però en aquest cas la diferència és molt més gran. Tot i que amb la utilització del pre-entrenament els millors resultats s'obtenien donant el mateix pes als exemples de tots dos corpus (`x1`), els dos feixos de la **[Fig. 112]** mostren que en aplicar un entrenament combinat és preferible augmentar el pes dels exemples del corpus principal (`x10`) i reduir el pes del pre-entrenament.

[Espai intencionadament en blanc per alinear el text amb les figures.]

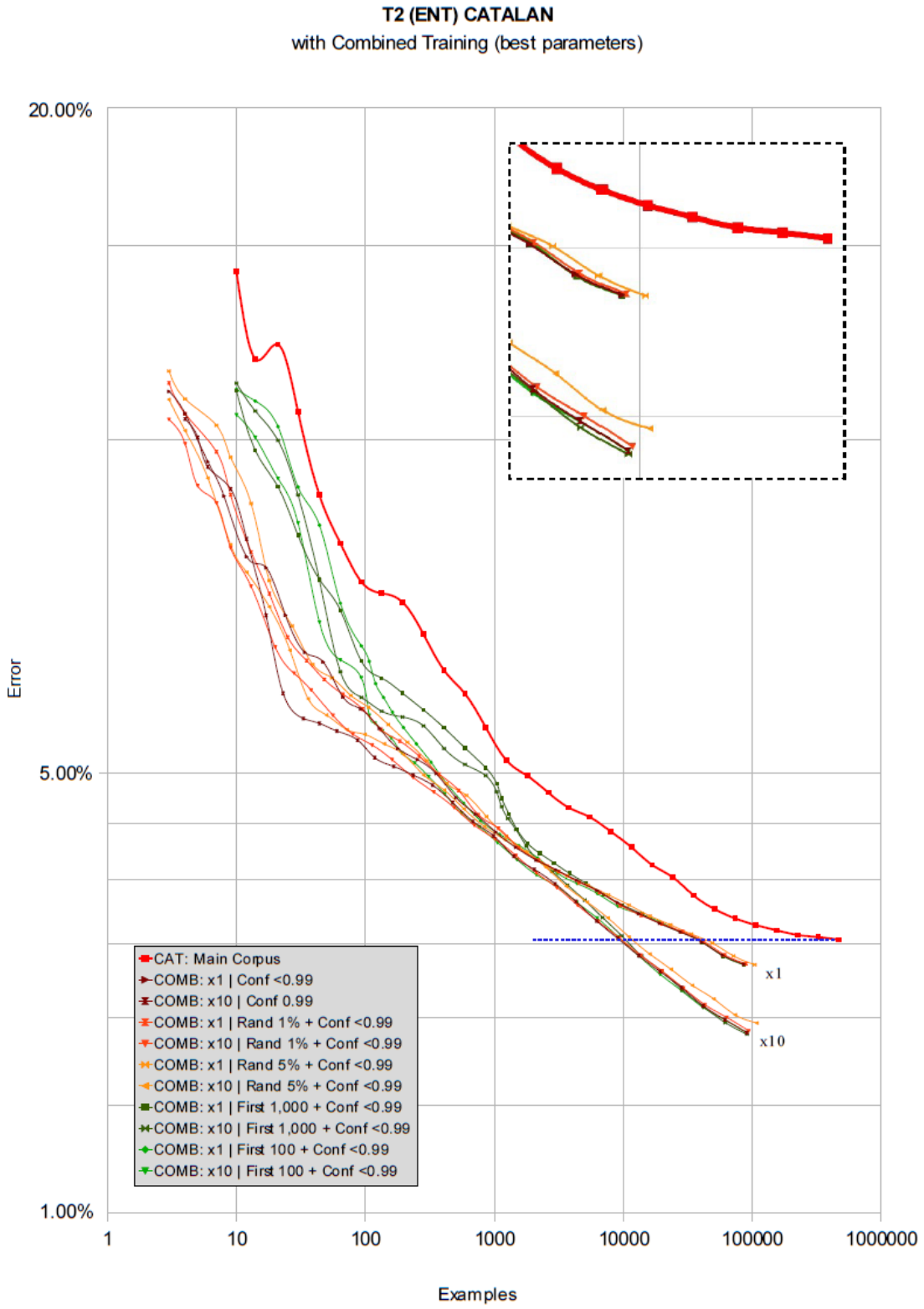

FIG. 112: Evolució de l'error de T2(ENT) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 113]** mostra l'evolució de l'error de la tasca T3 entrenada directament amb el corpus principal, línia de color vermell, i els resultats de diferents entrenaments combinats.

Es pot veure com encara que no presenta uns feixos de corbes tan compactes com a la tasca T1 els resultats van en la mateixa línia: les corbes d'error dels entrenaments combinats redueixen l'error en el tram inicial i també en el tram final. I tot i que el gràfic no permet discriminar-ho, els resultats de les corbes amb un pes `x100` tenen errors lleugerament inferiors a les que tenien un pes de `x10`.

[Espai intencionadament en blanc per alinear el text amb les figures.]

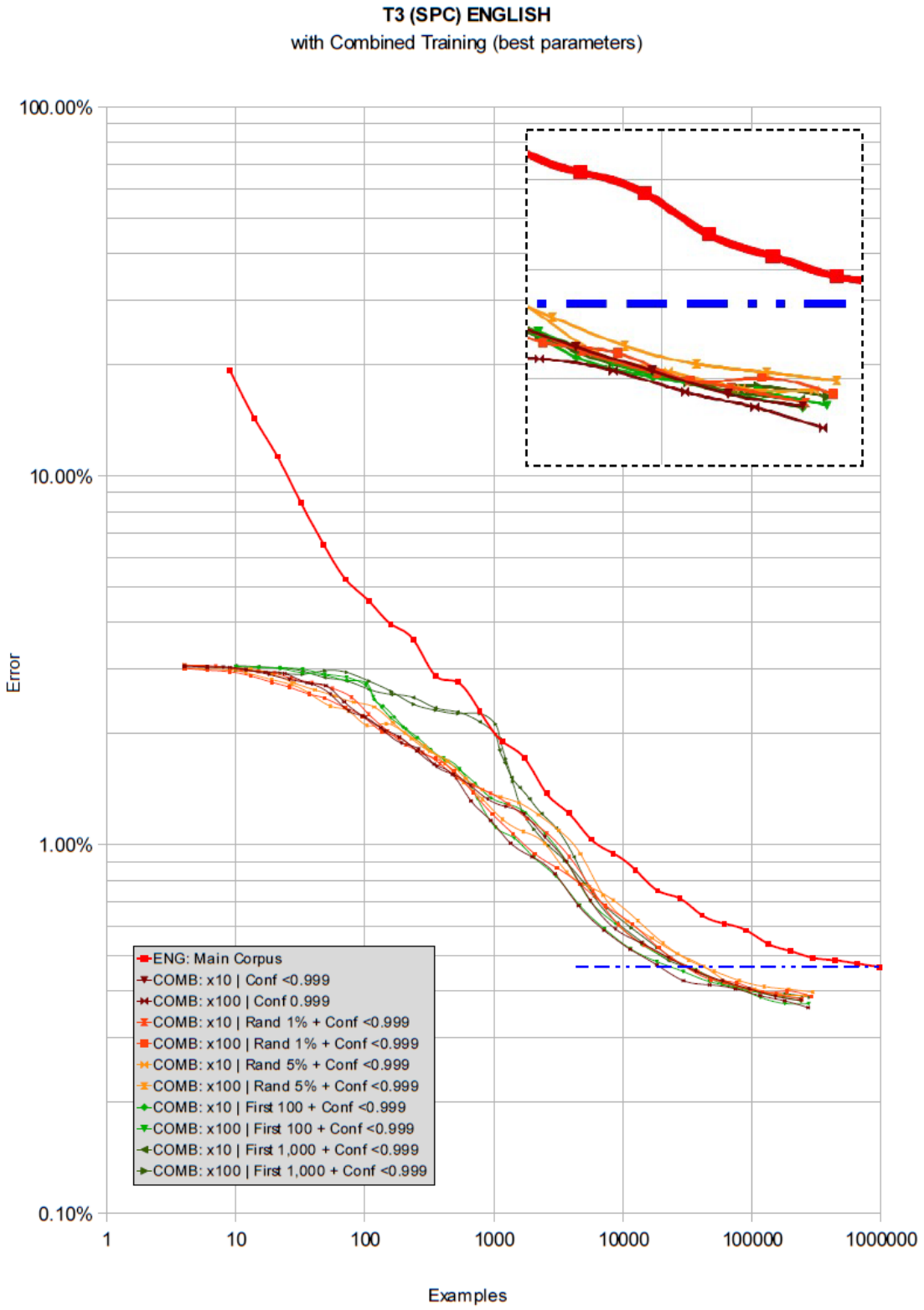

FIG. 113: Evolució de l'error de T3(SPC) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T5 (BRK)

Finalment, el gràfic de la **[Fig. 114]** mostra l'evolució de l'error de la tasca T5 entrenada directament amb el corpus principal, línia de color vermell, i els resultats de diferents entrenaments combinats.

Les corbes mostren sense cap mena de dubte que l'entrenament combinat permet reduir l'error a l'inici de l'entrenament i, simultàniament, accelerar l'entrenament amb la utilització de molts menys exemples. És interessant observar com per a cada parell de corbes del mateix color, és a dir, de la mateixa variant d'entrenament actiu, de manera sistemàtica la corba de més a l'esquerra és la `x100` i la de la dreta `x10`. Això confirma que en un entrenament combinat cal infraponderar els exemples de pre-entrenament.

[Espai intencionadament en blanc per alinear el text amb les figures.]

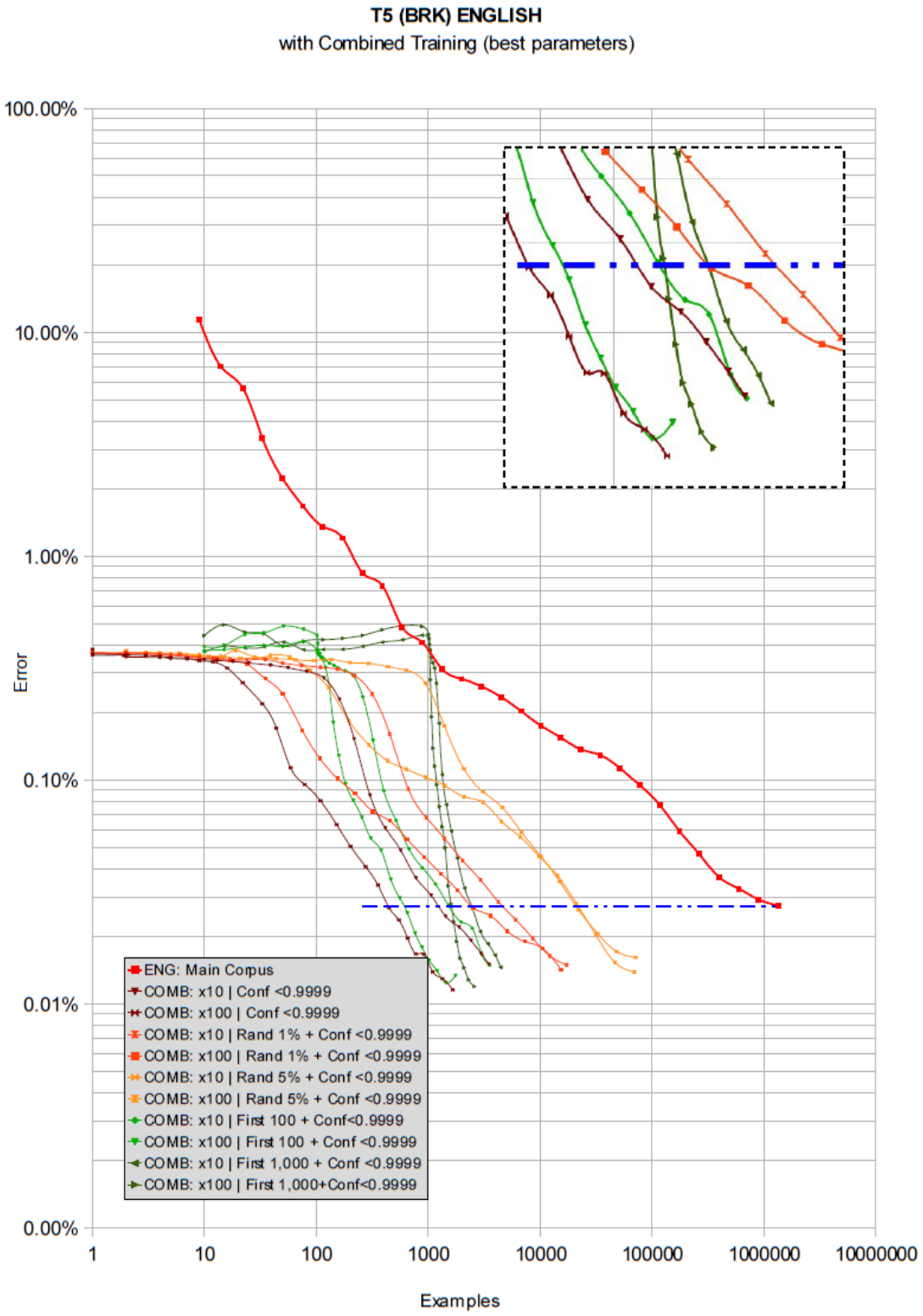

FIG. 114: Evolució de l'error de T5(BRK) pels millors casos amb entrenament combinat.

14.3 COMPARATIVES

De la mateixa manera que s'ha fet al capítol anterior, en aquesta secció s'han volgut sintetitzar els resultats dels més de 40 experiments d'entrenament combinat, centrant-se en les dades finals: error i tant per cent del corpus utilitzat.

Recordem que les següents gràfiques combinen aquests dos valors en un eix de coordenades per comparar-los amb l'entrenament de referència (quadrat vermell), perquè la seva interpretació sigui directa: a) els punts a l'esquerra de la referència són models més eficients, i b) els punts per sota de la referència són models més precisos.

EVOLUCIÓ DE T1 (POS)

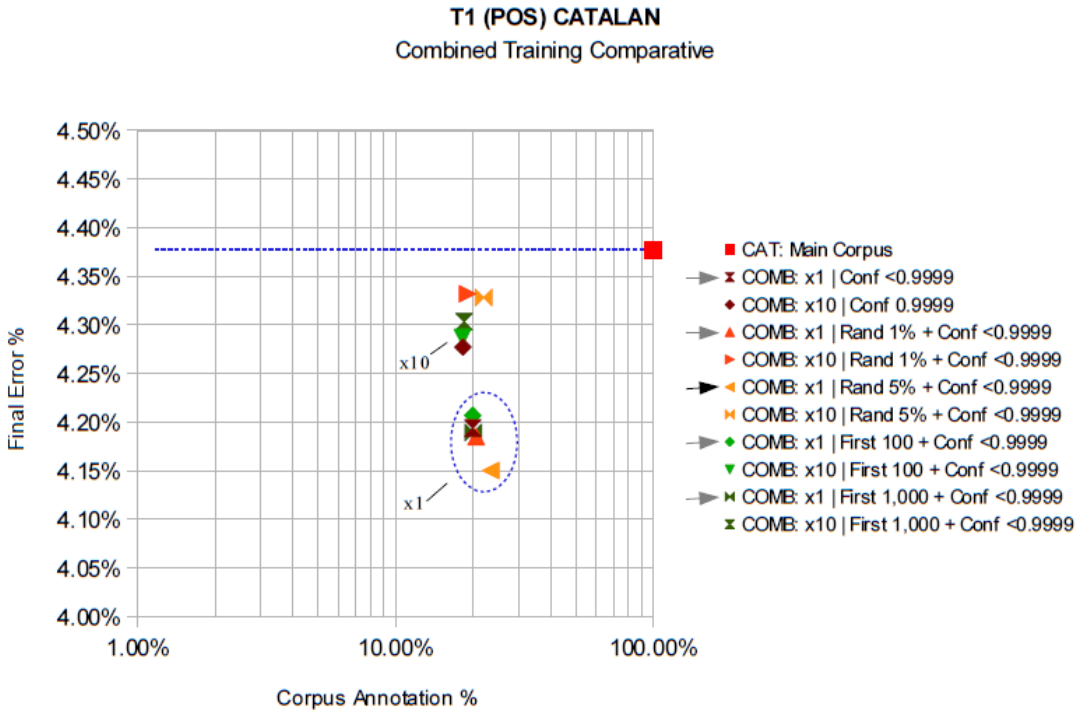

FIG. 115: Comparativa dels errors finals de T1(POS) amb entrenament combinat.

Aquest gràfic **[Fig. 115]** mostra els resultats finals de totes les variants d'entrenament combinat per la tasca T1. El cercle discontinu i les fletxes de la llegenda assenyalen els millors resultats.

Es pot observar com els resultats s'agrupen segons els dos feixos comentats anteriorment, però contràriament al que aparentment indicaven les corbes d'error, els millors resultats en aquesta tasca s'obtenen donant el mateix pes als exemples del corpus de pre-entrenament que als del corpus principal. Per tant queda clar que, en relació a la tasca T1, les dades d'entrenament dels corpus català i castellà són pràcticament equivalents.

EVOLUCIÓ DE T2 (ENT)

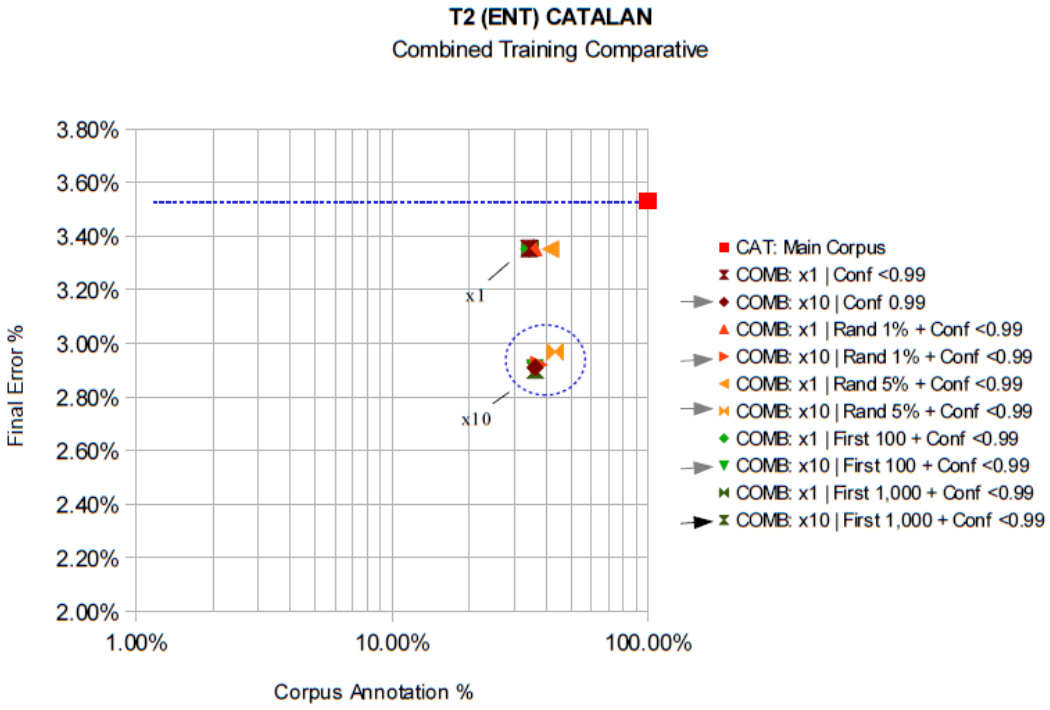

FIG. 116: Comparativa dels errors finals de T2(ENT) amb entrenament combinat.

En aquest cas **[Fig. 116]** es mostren els resultats per la tasca T2 on, malgrat utilitzar el mateix corpus, el resultat és el contrari tal com mostraven les corbes d'error: els millors resultats, cercle discontinu, s'obtenen reduint el pes dels exemples del pre-entrenament (`x10`).

En qualsevol cas, tots els resultats obtenen un error inferior al de l'entrenament de referència malgrat que les eficiències no són massa elevades, d'entre el 30% i 40%.

EVOLUCIÓ DE T3 (SPC)

Al gràfic **[Fig. 117]**, amb les dades dels experiments de la tasca T3, es pot observar com tots els experiments obtenen resultats molt similars i tots els punts s'agrupen al voltant d'un error del 0,38%.

El motiu és la saturació del model que malgrat les diferents tècniques i la utilització de diferents quantitats d'exemples, entre el 22% i el 30% del corpus, l'error assolit és el mínim de la tasca.

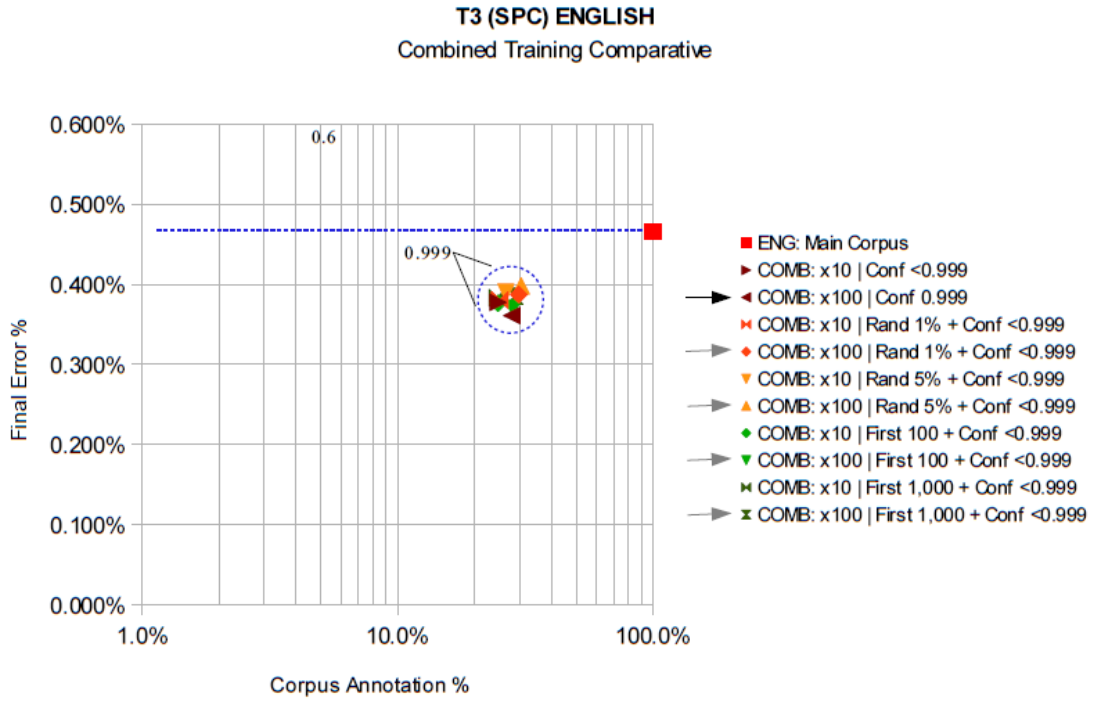

FIG. 117: Comparativa dels errors finals de T3(SPC) amb entrenament combinat.

EVOLUCIÓ DE T5 (BRK)

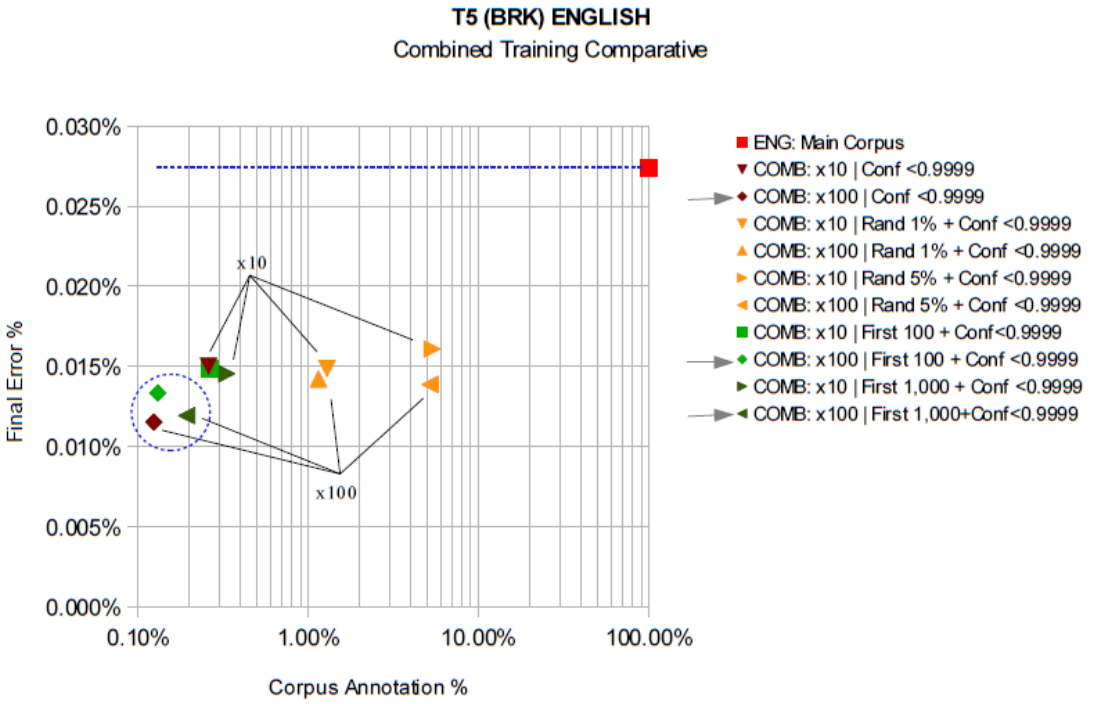

FIG. 118: Comparativa dels errors finals de T5(BRK) amb entrenament combinat.

Finalment, al gràfic de la tasca T5 **[Fig. 118]**, la tasca basada en el corpus sintètic molt redundant, es confirma la preferència per l'entrenament actiu endarrerit respecte el mostreig aleatori que, com ja s'ha dit, imposa un límit inferior en l'eficiència, com mostren els triangles grocs just a la dreta de l'1% i 5%. Els millors resultats els obtenen els entrenament endarrerits (`First` de 100 i 1.000) i amb sobreponderació del corpus principal (`x100`).

14.4 EVOLUCIÓ DELS MODELS

En els següents apartats d'aquesta secció es mostra una comparativa resum de les corbes d'avaluació que han obtingut millors resultats en les diferents tècniques utilitzades i per les quatre tasques de referència.

Concretament, es superposen els entrenaments de referència, els resultats amb pre-entrenament, amb entrenament actiu i amb entrenament combinat. Juntament amb cada figura es mostren els resultats obtinguts i s'analitzen les conclusions que poden observar-se en cada una de les gràfiques:

[Espai intencionadament en blanc per alinear el text amb les figures.]

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 119]** permet comparar les principals corbes d'entrenament i observar l'evolució de l'entrenament combinat de la tasca T1 en relació a l'entrenament estàndard amb línia vermella i els entrenaments actiu i amb pre-entrenament amb línies discontinues.

L'additivitat de les dues tècniques és pràcticament perfecte en aquest gràfic, reduint l'error inicial d'un 55% a menys d'un 20%, i superant l'error final amb un 20% dels exemples.

De totes maneres, segons es pot comprovar a la taula següent, els resultats finals són molt similars, per no dir idèntics, als millors de l'entrenament actiu; i l'únic experiment que el supera (`Rand=5%`) per dues centèsimes és a costa d'utilitzar un 10% més d'exemples.

Best Training Parameters	**Error**			**ASR**
	Min	Max	**Mean**	
T1-STD CAT (reference)	*4.20%*	*4.52%*	***4.38%***	***100.0%***
T1-ACT First 100 + Conf<0.9999	*4.06%*	*4.32%*	***4.17%***	***20.06%***
T1-COMB x1 + Conf<0.9999	4.09%	4.32%	**4.19%**	**19.93%**
T1-COMB x1 + Rand 1% + Conf<0.9999	3.98%	4.33%	**4.18%**	**20.53%**
T1-COMB x1 + Rand 5% + Conf<0.9999	3.98%	4.39%	**4.15%**	**23.41%**
T1-COMB x1 + First 100 + Conf<0.9999	4.03%	4.36%	**4.21%**	**19.91%**
T1-COMB x1 + First 1,000 + Conf<0.9999	4.07%	4.36%	**4.19%**	**20.04%**

TAULA 58: Resultats de les proves amb entrenament combinat per la T1 (POS).

[Espai intencionadament en blanc per alinear el text amb les figures.]

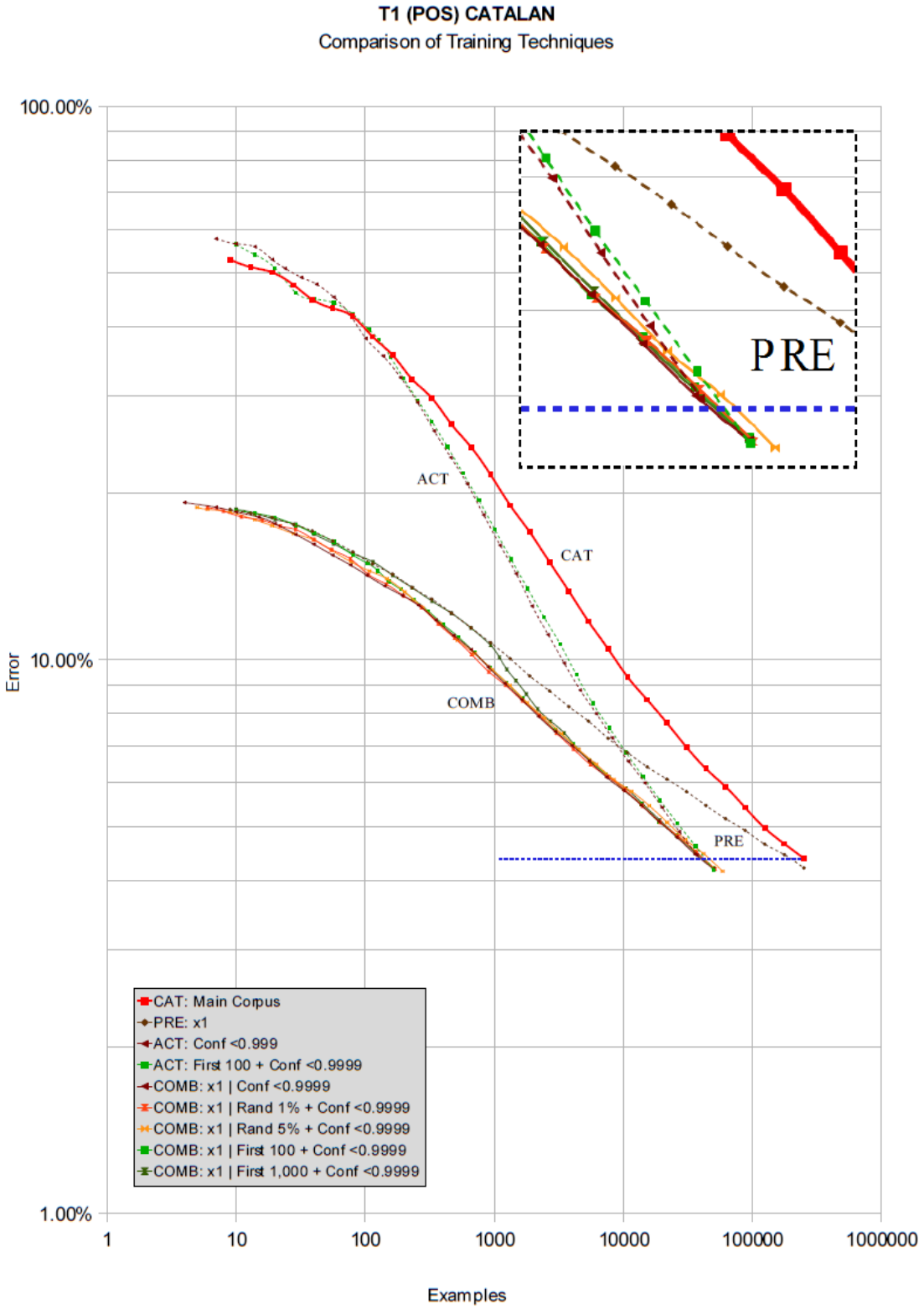

FIG. 119: Evolució de l'error de T1(POS) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T2 (ENT)

A la **[Fig. 120]** es comparen l'evolució dels entrenaments combinats de la tasca T2 amb l'entrenament estàndard amb línia vermella i els entrenaments actiu i amb pre-entrenament amb línies discontinues.

Es pot veure com l'entrenament combinat permet assolir els beneficis dels dos entrenaments simples, obtenint un error inferior a l'entrenament estàndard al llarg de tot l'entrenament. A diferència del que passava a la tasca anterior, l'entrenament combinat no supera els millors resultats de l'entrenament actiu. Com es pot comprovar a la taula següent, aconsegueix un error més alt, 2,91% en comptes de 2,79%, i a més utilitzant un 80% més d'exemples.

Best Training Parameters	**Error**			**ASR**
	Min	Max	**Mean**	
T2-STD CAT (reference)	*3.36%*	*3.65%*	***3.53%***	***100.0%***
T2-ACT First 1,000 + Conf<0.99	*2.65%*	*2.93%*	***2.79%***	***20.10%***
T2-COMB x10 + Conf<0.99	2.76%	3.06%	**2.91%**	**36.20%**
T2-COMB x10 + Rand 1% + Conf<0.99	2.84%	3.05%	**2.92%**	**37.43%**
T2-COMB x10 + First 100 + Conf<0.99	2.85%	3.02%	**2.91%**	**36.08%**
T2-COMB x10 + First 1,000 + Conf<0.99	2.83%	3.03%	**2.90%**	**36.22%**

TAULA 59: Resultats de les proves amb entrenament combinat per la T2 (ENT).

[Espai intencionadament en blanc per alinear el text amb les figures.]

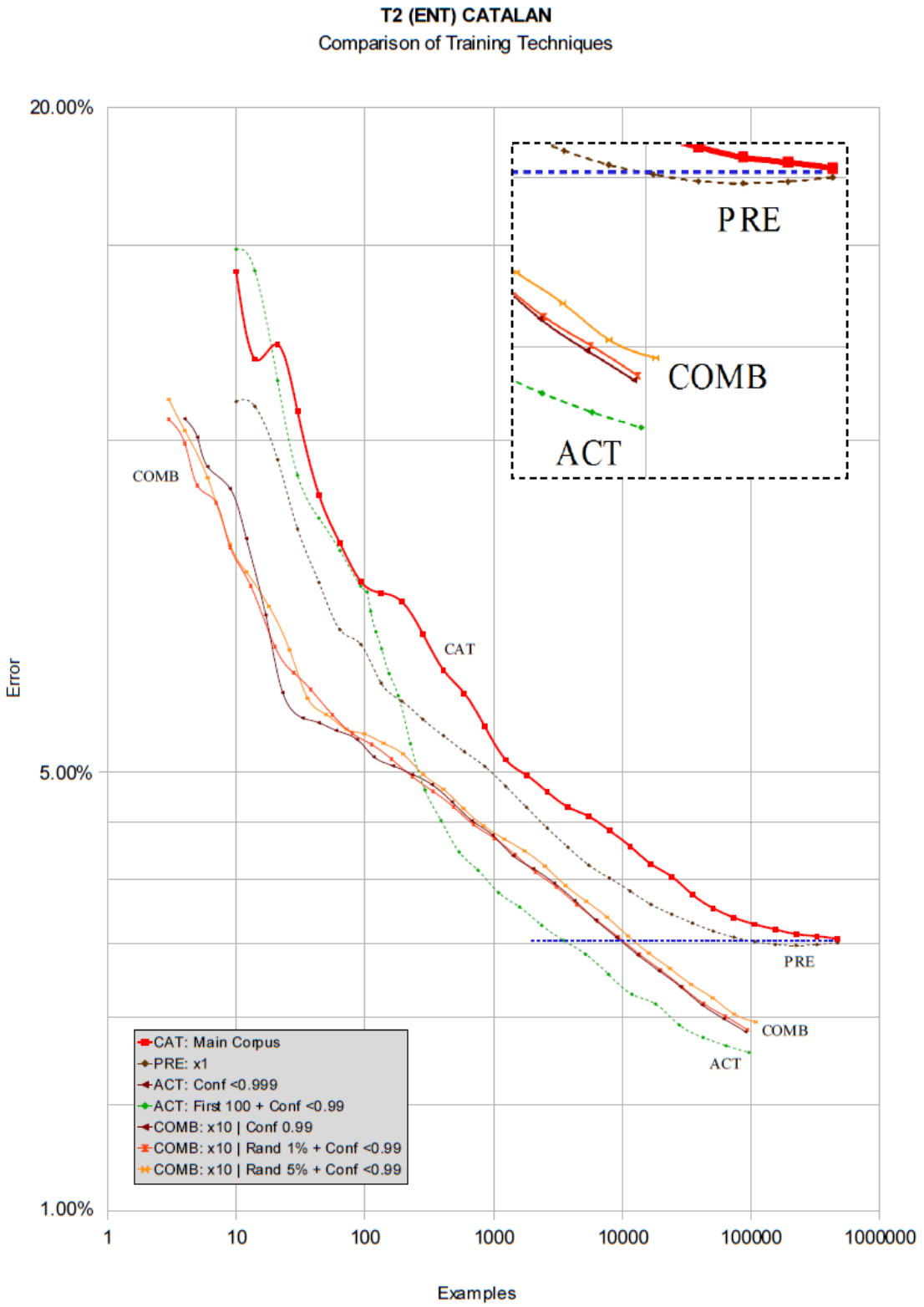

FIG. 120: Evolució de l'error de T2(ENT) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T3 (SPC)

El gràfic de la **[Fig. 121]** compara l'evolució de les diferents tècniques d'entrenament utilitzades a la tasca T3. Els resultats continuen demostrant la possibilitat de combinar les dues tècniques i els seus beneficis. L'entrenament combinat permet superar l'entrenament estàndard al llarg de tot l'entrenament, cosa que no s'aconseguia amb només el pre-entrenament, però tampoc pot superar els millors resultats de l'entrenament actiu, ja que utilitzant més exemples assoleix un error mínim de 0,361% respecte el 0,317%.

Best Training Parameters	Error			ASR
	Min	Max	Mean	
T3-STD ENG (reference)	*0.439%*	*0.489%*	***0.466%***	***100.0%***
T3-ACT First 1,000 + Conf<0.999	*0.266%*	*0.354%*	***0.317%***	***27.3%***
T3-COMB x100 + Conf<0.999	0.316%	0.426%	**0.361%**	**28.0%**
T3-COMB x100 + Rand 1% + Conf<0.999	0.319%	0.422%	**0.387%**	**29.8%**
T3-COMB x100 + Rand 5% + Conf<0.999	0.365%	0.433%	**0.398%**	**30.4%**
T3-COMB x100 + First 100 + Conf<0.999	0.312%	0.404%	**0.370%**	**28.3%**
T3-COMB x100 + First 1,000 + Conf<0.999	0.350%	0.421%	**0.385%**	**28.8%**

TAULA 60: Resultats de les proves amb entrenament combinat per la T3 (SPC).

[Espai intencionadament en blanc per alinear el text amb les figures.]

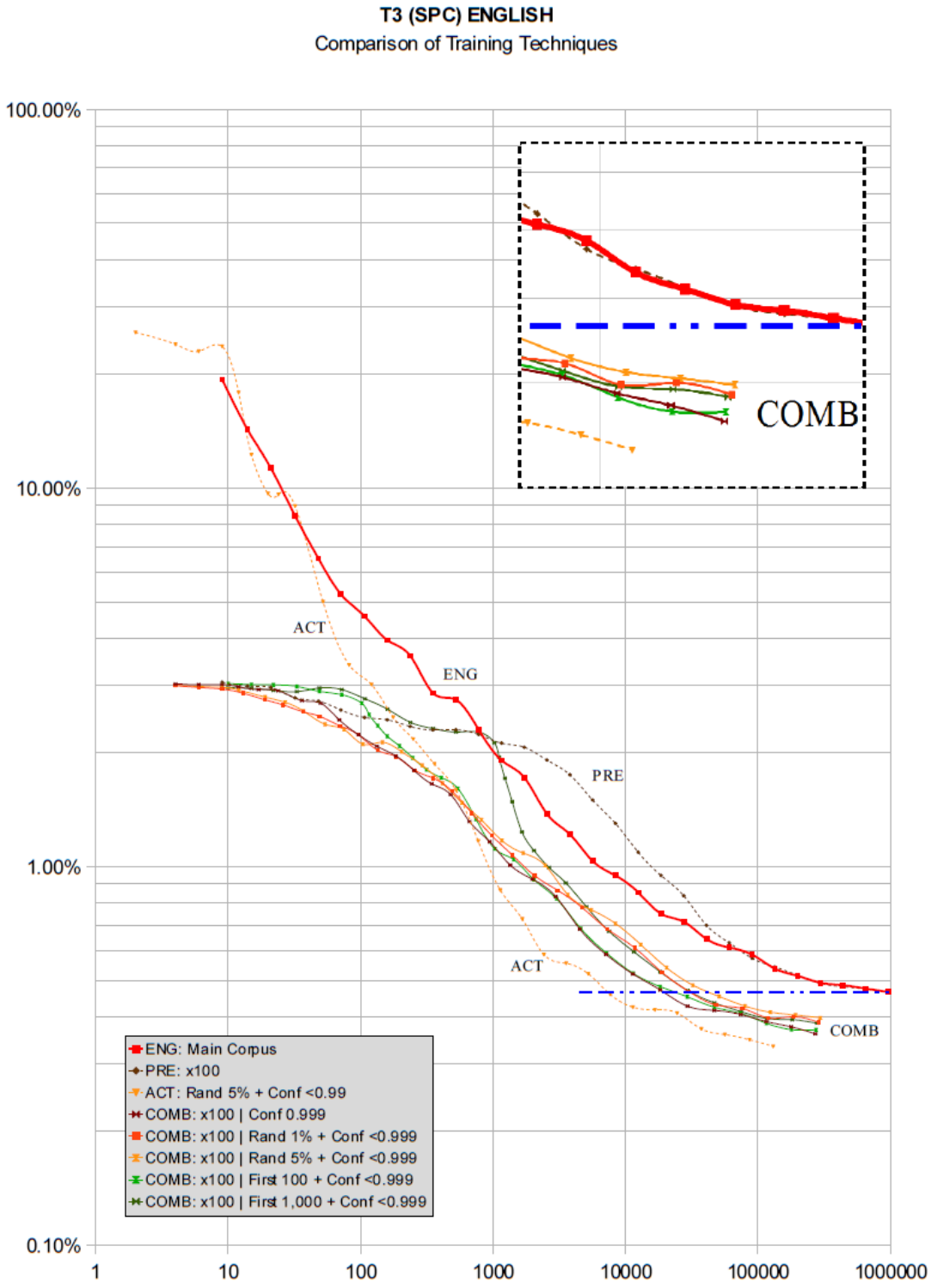

FIG. 121: Evolució de l'error de T3(SPC) pels millors casos amb entrenament combinat.

EVOLUCIÓ DE T5 (BRK)

Finalment, al gràfic comparatiu per a la tasca T5, veure **[Fig. 122]**, els resultats són els mateixos però amb valors més extrems. L'entrenament combinat comença amb un error molt baix, al voltant de 0,4%, que es redueix extraordinàriament fins al 0,012% abans d'arribar als 1.000 exemples.

I tot i que aquest error no supera el millor resultat de l'entrenament actiu, amb un error final de 0,009%, n'obté un error pràcticament equivalent utilitzant la meitat d'exemples. Les dades completes poden consultar-se a la taula següent:

Best Training Parameters	**Error**			**ASR**
	Min	Max	**Mean**	
T5-STD ENG (reference)	*0.021%*	*0.036%*	***0.027%***	***100.0%***
T5-ACT First 100 + Conf<0.999	*0.004%*	*0.013%*	***0.009%***	***0.23%***
T5-COMB x100 + Conf<0.9999	0.009%	0.017%	**0.012%**	**0.12%**
T5-COMB x100 + First 100 + Conf<0.9999	0.008%	0.019%	**0.013%**	**0.13%**
T5-COMB x100 + First 1,000 + Conf<0.9999	0.006%	0.017%	**0.012%**	**0.19%**

TAULA 61: Resultats de les proves amb entrenament combinat per la T5 (BRK).

[Espai intencionadament en blanc per alinear el text amb les figures.]

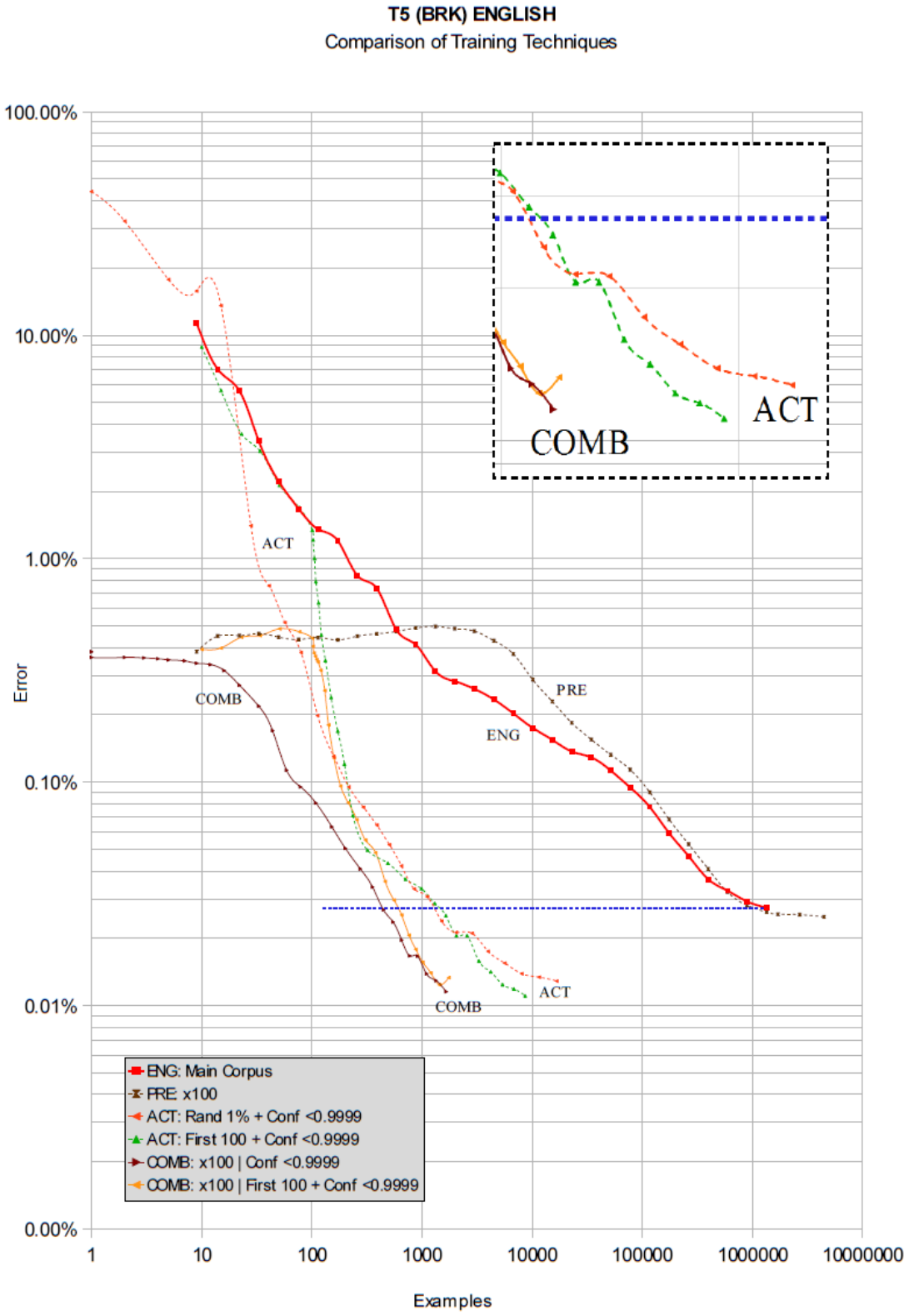

FIG. 122: Evolució de l'error de T5(BRK) pels millors casos amb entrenament combinat.

14.5 CONCLUSIONS

Els resultats mostren d'una manera clara que la utilització de l'entrenament combinat permet induir models que no només assoleixen errors finals inferiors als de l'entrenament estàndard, sinó que des del primer moment i al llarg de tot l'entrenament; a més, ho fan de manera més eficient utilitzant molts menys exemples.

Si prenem com a referència la quantitat d'exemples necessaris per assolir el mateix error que l'entrenament estàndard, i ho representem com a tant per cent del corpus utilitzat i com el factor reductor dels exemples necessaris, obtenim la **[Taula 62]**. A la taula es pot veure com la quantitat d'exemples necessaris pot reduir-se a valors molt semblants als de l'entrenament actiu pur, entre sis i centenars de vegades segons la tasca:

Tasks	Corpus Used	Saving Factor
T1(POS)	16.6%	x 6
T2(ENT)	2.06%	x 48
T3(SPC)	1.96%	x 51
T5(BRK)	0.045%	x 2.200

TAULA 62: Corpus utilitzat per l'entrenament combinat per a igualar l'error obtingut per l'entrenament estàndard.

Si l'objectiu és minimitzar l'error i aprofitar al màxim el corpus disponible, obtenim reduccions de l'error d'entre un 5% i un 50%, utilitzant una quarta part dels exemples, com mostra la taula següent:

Tasks	Reference Error	Final Error	Improvement	Corpus Used	Saving Factor
T1(POS)	4.38%	4.15%	**-5.2%**	**23%**	x 4
T2(ENT)	3.53%	2.91%	**-17%**	**36%**	x 3
T3(SPC)	0.466%	0.361%	**-22%**	**28%**	x 4
T5(BRK)	0.027%	0.012%	**-55%**	**0.12%**	x 830

TAULA 63: Reducció de l'error i corpus utilitzat al finalitzar l'entrenament actiu.

Tot i que l'entrenament combinat obté resultats molt millors que l'entrenament estàndard, en general no supera els resultats obtinguts per l'entrenament actiu, com es pot veure a les dues taules següents **[Taula 64; Taula 65]**. A l'hora d'igualar l'error de l'entrenament estàndard, l'entrenament combinat necessita més exemples que l'entrenament actiu. I a l'hora de minimitzar l'error aconsegueix resultats molt semblants, errors lleugerament més alts utilitzant més quantitat d'exemples.

L'única tasca on l'entrenament combinat supera àmpliament l'eficiència de l'entrenament actiu és la T5, la del corpus sintètic, però donada l'elevada redundància que presenta i la seva naturalesa artificial no permet extreure cap conclusió sòlida.

Tasks	Corpus Used		Saving Factor	
	ACT	COMB	ACT	COMB
T1(POS)	18%	**16.6%**	×5.5	**x 6**
T2(ENT)	**0.9%**	2.06%	**×100**	x 48
T3(SPC)	**0.3%**	1.96%	**×300**	x 51
T5(BRK)	**0.03%**	0.045%	**×3.800**	x 2.200

TAULA 64: Comparativa entre l'entrenament actiu i el combinat per assolir l'error de l'entrenament estàndard.

Tasks	Improvement		Corpus Used		Saving Factor	
	ACT	COMB	ACT	COMB	ACT	COMB
T1(POS)	-4.8%	**-5.2%**	**22%**	23%	**×5**	x 4
T2(ENT)	**-21%**	-17%	**20%**	36%	**×5**	x 3
T3(SPC)	**-24%**	-22%	**7%**	28%	**×14**	x 4
T5(BRK)	**-67%**	-55%	0.21%	**0.12%**	×470	**x 830**

TAULA 65: Comparativa entre l'entrenament actiu i el combinat al finalitzar els entrenaments.

Finalment, en relació a l'anomalia observada a l'entrenament combinat de la tasca T2 (en la qual, a diferència d'en el pre-entrenament pur, és preferible sobreponderar (`x10`) el corpus principal) existeixen dues hipòtesis que suggereixen una explicació.

En primer lloc cal recordar que la variable `Weight` no controla el pes relatiu dels corpus auxiliar i principal, sinó el pes relatiu dels seus exemples individuals. Per això, tenint en compte que l'entrenament actiu redueix considerablement el nombre d'exemples del corpus principal, el fet de donar el mateix pes a tots els exemples (`x1`), cosa que en el pre-entrenament suposava donar el mateix pes a tots dos corpus, en l'entrenament combinat suposa donar un pes molt superior al corpus auxiliar. Per això, en cas de voler donar el mateix pes a tots dos corpus, cal augmentar el pes del corpus principal amb un factor aproximat a l'invers a la fracció d'exemples seleccionats ($1/ASR$).

Per un altre costat no és cert que les dues tècniques (pre-entrenament i entrenament actiu) siguin totalment independents i puguin combinar-se sense interferir-se. Cal tenir en compte que l'entrenament actiu depèn de la qualitat del propi model classificador per seleccionar els exemples amb què serà entrenat. Això significa que el pre-entrenament amb un corpus auxiliar afecta el criteri de selecció, i fa que el subconjunt d'exemples seleccionats del corpus principal no sigui exactament el mateix que en un entrenament actiu pur.

15 EFICIÈNCIA: ANNOTATION SAMPLING RATIO (ASR)

15.1 INTRODUCCIÓ

En els capítols anteriors s'han presentat diferents tècniques d'entrenament incremental i s'han mostrat els seus resultats des d'un punt de vista estrictament utilitari: l'error assolit i l'eficiència de l'entrenament. En aquest capítol, i els dos següents, s'analitzen altres variables que indirectament permeten entendre l'evolució interna dels classificadors.

Aquest capítol comença introduint la variable *ASR* (*Annotation Sampling Ratio*) com un valor dinàmic que evoluciona al llarg de l'entrenament. En les seccions següents es presenta l'evolució d'aquesta variable durant els entrenaments actius de les tasques de referència i s'analitzen els resultats per extreure algunes conclusions; tot seguit es contrasta l'evolució de l'*ASR* durant els entrenaments combinats amb resultats molt similars.

Finalment, es conclou que l'entrenament actiu permet reduir progressivament l'*ASR* obtenint classificadors capaços de processar grans corpus de manera progressivament més eficients. També s'analitza l'efecte que tenen sobre l'*ASR* el paràmetres `First` i `Confidence`, i s'alerta sobre l'elevada sensibilitat que té l'*ASR* respecte aquest darrer paràmetre. Per resoldre aquest inconvenient es suggereix una línia de recerca al voltant de la creació de tècniques d'entrenament incremental amb llindars de confiança variables que es reajustin al llarg de l'entrenament.

15.2 L'ASR DINÀMIC

Com han mostrat els resultats dels experiments realitzats als capítols anteriors, el principal benefici de l'anotació inter-activa és l'eficiència a l'hora de minimitzar la quantitat d'exemples anotats necessaris per entrenar el classificador.

Aquesta eficiència s'ha quantificat mitjançant la proporció d'exemples seleccionats per ser anotats, una mesura que ha estat anomenada amb les sigles de l'acrònim *Annotation Sampling Ratio* (*ASR*). Aquesta mesura és un valor percentual que indica quina fracció dels exemples mostrats al classificador han estat seleccionats com a dubtosos, o informativament més rics, i coincideix amb el que altres autors **[Sculley, 2007]** anomenen *Sampling Rate*.

Fins ara aquesta mesura s'ha presentat com un valor puntual, calculat al finalitzar l'entrenament i que coincideix amb el tant per cent d'exemples seleccionats de la totalitat del corpus. Però en realitat és un valor dinàmic que canvia al llarg de l'entrenament, i que presenta una evolució que pot oferir indicis sobre el que està succeint a l'interior del classificador.

A les seccions següents d'aquest capítol es mostren les corbes d'evolució de l'*ASR* dels diferents entrenaments actius de cada una de les quatre tasques de referència, vegeu **[13. Utilització d'Entrenament Actiu]**, i s'interpreten les possibles causes d'aquestes evolucions.

Les corbes estan representades en un diagrama amb eixos en escala logarítmica, on l'eix horitzontal representa el nombre d'exemples mostrats al classificador, i l'eix vertical representa l'ASR, el tant per cent d'exemples requerits per ser anotats fins aquell moment.

15.3 EVOLUCIÓ DE L'ASR A LA TASCA T1(POS)

Al gràfic de la **[Fig. 123]** es mostra l'evolució de l'*ASR* dels entrenaments actius de la tasca T1, es poden consultar les corbes d'avaluació corresponents a la **[Fig. 94]** i **[Fig. 98]**.

Es pot observar com totes les corbes presenten una trajectòria decreixent. A mesura que avança l'entrenament el sistema descarta més exemples i selecciona una fracció menor, de manera que l'*ASR* va disminuint al llarg del procés.

El nivell final que assoleixen depèn principalment del valor del llindar de `Confidence`. Tot i això, es pot observar com el trajecte per on arriben depèn molt de la variant d'entrenament actiu utilitzada, de manera que es formen tres feixos de corbes ben diferenciats.

Les línies corresponents als entrenaments actius amb mostratge aleatori, de colors taronges, disminueixen des del primer moment amb un pendent no inferior al valor del paràmetre `Rand`. Es pot observar com totes les corbes amb un `Rand` del 5%, taronja clar, se situen per sobre de les corbes amb un `Rand` de l'1% amb el mateix.

Per contra, les línies dels entrenaments actius endarrerits, de colors verds, mostren un valor d'*ASR* constant i igual al 100% fins el moment on s'inicia l'entrenament actiu i el valor cau ràpidament: en un cas en superar els 100 exemples i en l'altre en superar els 1.000 exemples. També s'observa com en el segon cas el pendent amb el qual disminueix és més elevat, cosa que demostra que un classificador entrenat amb més exemples, té més capacitat de discriminació a l'hora d'aplicar l'entrenament actiu.

[Espai intencionadament en blanc per alinear el text amb les figures.]

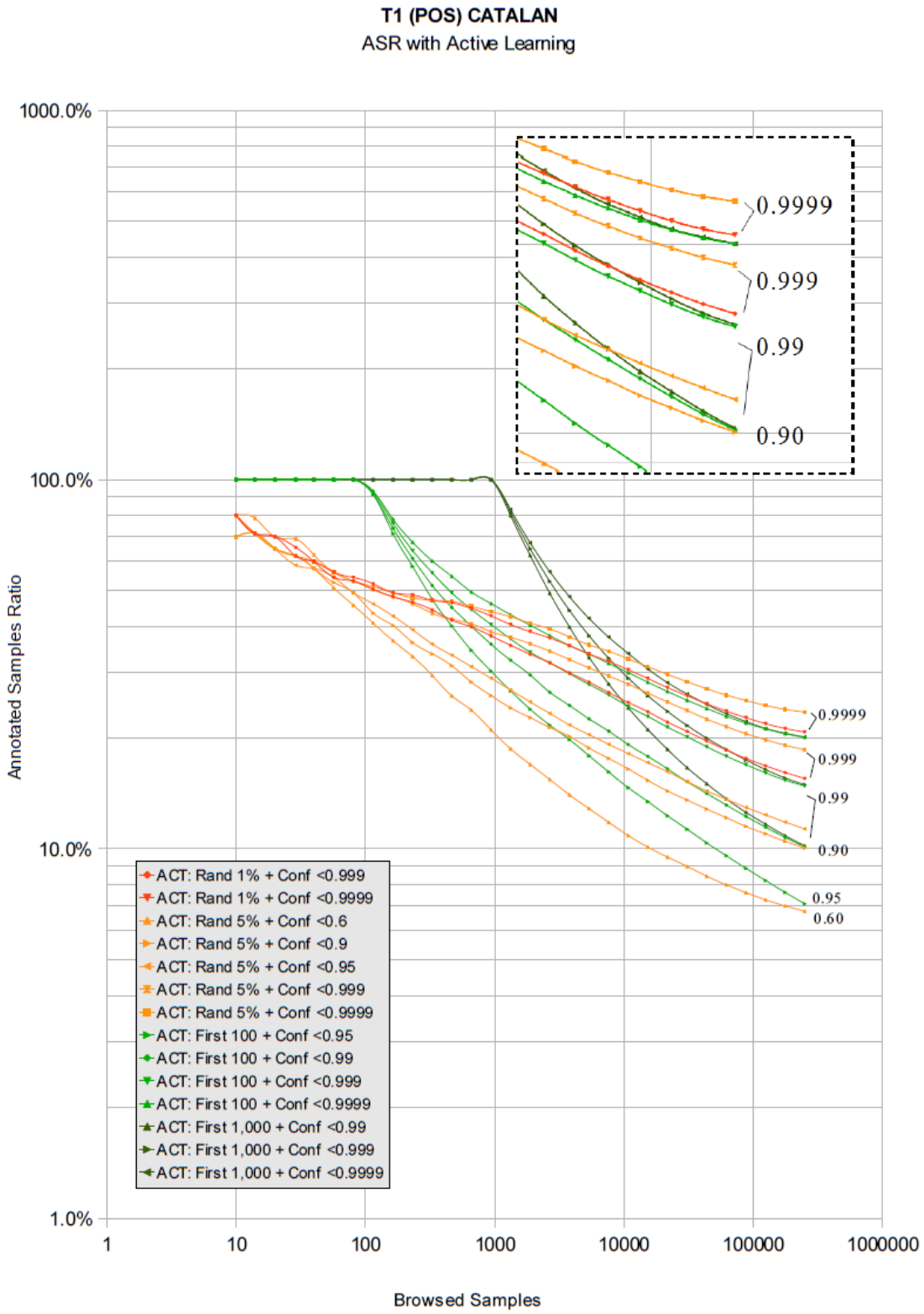

FIG. 123: Evolució de l'eficiència (ASR) de T1(POS) durant l'entrenament actiu.

15.4 EVOLUCIÓ DE L'ASR A LA TASCA T2(ENT)

Al gràfic de la **[Fig. 124]** es mostra l'evolució de l'*ASR* dels entrenaments actius de la tasca T2, es poden consultar les corbes d'avaluació corresponents a la **[Fig. 95]** i **[Fig. 99]**.

La primera cosa que s'observa és que les corbes presenten unes variacions molt més altes, especialment les corbes corresponents als entrenaments amb mostreig aleatori, de colors taronges. Per exemple, les corbes amb un Rand de l'1%, taronja fosc, baixen molt ràpidament fins a un *ASR* de l'1% i després reboten fins a recuperar una trajectòria típica. Aquestes oscil·lacions són pròpies de l'etapa transitòria, en que el model que no ha vist prou exemples sobrepondera els seus graus de certesa i triga a "reaccionar" i detectar la seva incapacitat per classificar correctament. Aquest problema queda molt reduït quan augmenta la freqüència del mostratge al 5%, corbes taronja clar.

Per contra, les corbes de l'entrenament endarrerit, línies verdes, segueixen uns camins molt més estables. Després d'entrenar els classificadors amb una primera mostra d'exemples, inicien l'entrenament actiu i l'*ASR* comença a disminuir suaument. Però a diferència del que passa a la tasca anterior, la trajectòria que inicialment es decreixent, arriba a un mínim i comença a augmentar. El punt d'inflexió és produeix més aviat com més alt sigui el llindar de `Confidence`: amb un valor de 0,99 el mínim se situa amb uns 90.000 exemples observats, amb un valor de 0,999 als 20.000 exemples, i amb un valor de 0,9999 l'*ASR* mínim es produeix amb uns 8.000 exemples observats. Però aquest avançament és aparent, a mesura que s'augmenta el llindar del `Confidence` es descarten menys exemples i per tant augmenta l'*ASR*, com mostren les corbes, i per tant és més alt el nombre d'exemples amb què realment ha estat entrenat. Si l'eix horitzontal fos el nombre d'exemples amb què ha estat entrenat no hi haurien gaires diferències del punt on l'*ASR* comença a pujar.

Independentment d'aquesta diferència, el fet interesant és que no és cert que l'*ASR* sigui un valor permanentment decreixent. Arriba un punt en què el model no millora, el sistema no és capaç de continuar reduint l'error i, mantenint fix el llindar, augmenta la quantitat d'exemples amb un grau de certesa inferior a aquest valor. Si s'observa l'evolució de l'error d'aquests experiments **[Fig. 99]** es pot veure com l'etapa on l'*ASR* és creixent, coincideix amb l'etapa on l'error s'estanca. Podria ser que l'*ASR* fos un indicador de la saturació del model.

[Espai intencionadament en blanc per alinear el text amb les figures.]

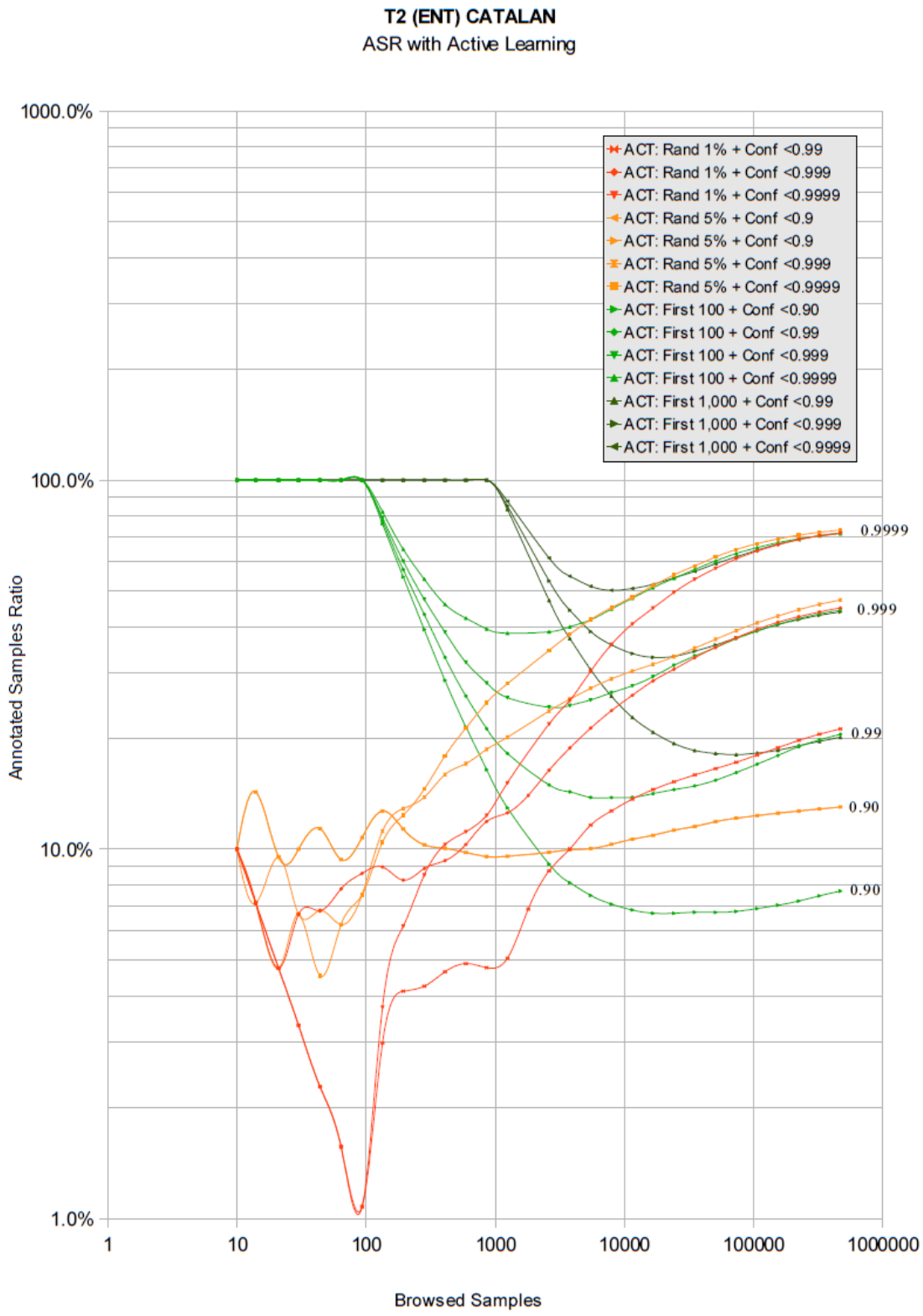

FIG. 124: Evolució de l'eficiència (ASR) de T2(ENT) durant l'entrenament actiu.

15.5 EVOLUCIÓ DE L'ASR A LA TASCA T3(SPC)

Al gràfic de la **[Fig. 125]** es mostra l'evolució de l'*ASR* dels entrenaments actius de la tasca T3, es poden consultar les corbes d'avaluació corresponents a la **[Fig. 96]** i **[Fig. 100]**.

El comportament de les línies continua seguint el mateix patró, tot i que les diferents trajectòries inicials s'acaben agrupant en feixos segons el nivell de `Confidence`, i dintre de cada feix l'*ASR* es redueix seguint les variants de més a menys eficients: mostreig aleatori del 5%, de l'1%, endarreriment fins els 100 primers exemples i, finalment, fins els 1.000 primers exemples.

Aquesta tasca també presenta mínims a les corbes d'*ASR* i, per tant, un *ASR* creixent al final de l'entrenament, que coincideix amb una etapa de saturació en la corba d'avaluació de l'error.

[Espai intencionadament en blanc per alinear el text amb les figures.]

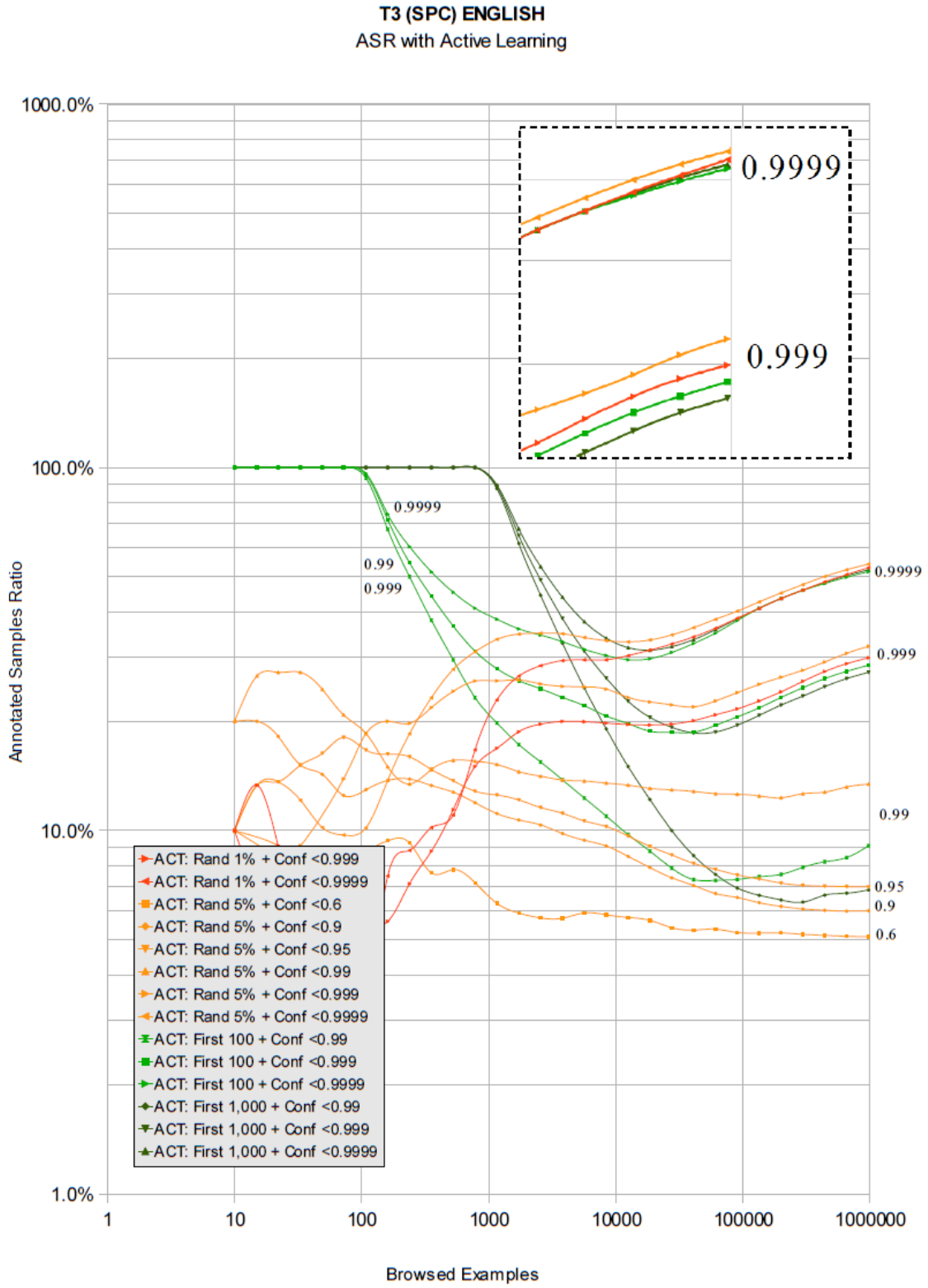

FIG. 125: Evolució de l'eficiència (ASR) de T3(SPC) durant l'entrenament actiu.

15.6 EVOLUCIÓ DE L'ASR A LA TASCA T5(BRK)

Al gràfic de la **[Fig. 126]** es mostra l'evolució de l'*ASR* dels entrenaments actius de la tasca T5, es poden consultar les corbes d'avaluació corresponents a la **[Fig. 97]** i **[Fig. 101]**.

Es pot observar com l'*ASR* de l'entrenament actiu amb mostreig aleatori, línies taronges, disminueix asimptòticament fins a un valor que coincideix amb el del paràmetre `Rand` (1% i 5%). És a dir, a mesura que recorre el corpus cada vegada troba menys exemples informatius, fins que finalment només és entrenat pels exemples mostrejats aleatòriament, amb una proporció de l'1% i 5% respectivament.

Però és amb el mostreig endarrerit, línies verdes, on el comportament és més extrem, ja que en no existir aquest mostreig aleatori aquest límit inferior de l'*ASR* és zero. Una vegada s'inicia l'entrenament actiu l'*ASR* disminueix amb pendent constant al llarg de tot l'entrenament. Aquest comportament sí coincideix amb el comportament ideal que s'havia predit al començament del capítol, i demostra l'artificialitat del corpus sintètic que a causa de la seva elevada redundància permet al classificador aprendre la pràctica totalitat dels casos utilitzant una petita fracció del corpus.

[Espai intencionadament en blanc per alinear el text amb les figures.]

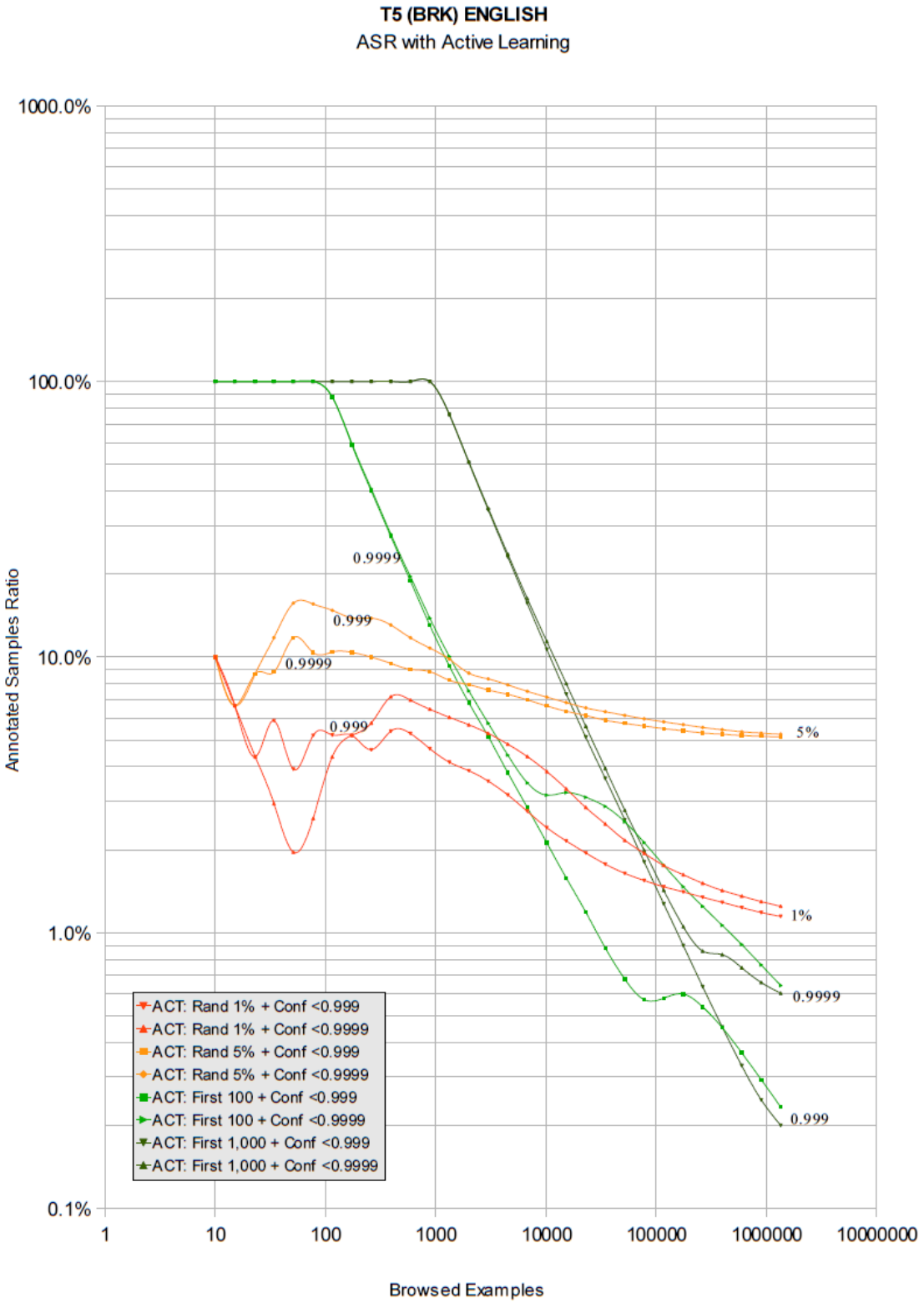

FIG. 126: Evolució de l'eficiència (ASR) de T5(BRK) durant l'entrenament actiu.

15.7 L'ASR ALS ENTRENAMENTS COMBINATS

Posteriorment s'ha analitzat l'evolució dels ASR duran els entrenaments combinats, els resultats es mostren a les quatre figures de la pàgina següent corresponents a cada una de les tasques de referència. Tot i que en general els valors són més alts que en l'entrenament actiu pur, el comportament global és molt similar.

Si observem la **[Fig. 127]** veurem l'evolució de l'*ASR* a l'entrenament combinat de la tasca T1, on la principal diferència és una menor dispersió en els valors assolits. L'explicació és que aquests experiments van realitzar-se únicament amb un valor de `Confidence` de 99,99%, però tant el valor final com les trajectòries de les tres variants són pràcticament idèntiques. Per tant, almenys en aquesta tasca, el pre-entrenament no ha afectat l'ASR.

L'evolució de l'*ASR* a l'entrenament combinat de la tasca T2 es mostra a la **[Fig. 128]**, on es pot veure que segueix una trajectòria molt similar al feix corresponent de l'entrenament actiu per a un valor de `Confidence` de 99%, tot i que ho fa amb valors d'*ASR* més elevats. En aquest cas, el pre-entrenament sí que ha perjudicat l'*ASR*.

Si observem la **[Fig. 129]** veurem l'evolució de l'*ASR* a l'entrenament combinat per la tasca T3, es pot veure com la trajectòria coincideix amb la de l'entrenament actiu pur amb un `Confidence` de 0,999, amb un valor mínim al voltant del 20% i un valor final al voltant del 30%.

I finalment, a la **[Fig. 130]** es mostra l'evolució de l'ASR de l'entrenament combinat, les trajectòries seguides són molt similars a les obtingudes a l'entrenament actiu amb un `Confidence` de 0,9999, però assolint un *ASR* lleugerament inferior, de fins a un 0,13% on abans assolia un 0,20%.

[Espai intencionadament en blanc per alinear el text amb les figures.]

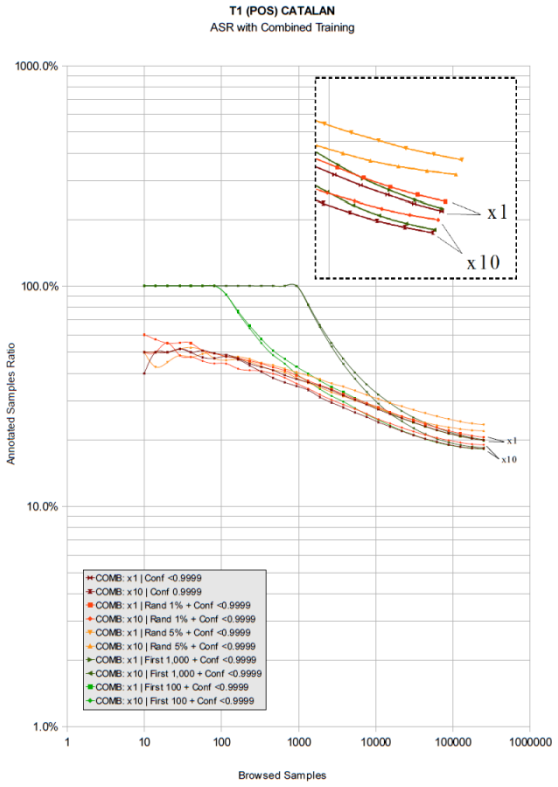

FIG. 127: Evolució de l'eficiència de T1(POS) durant l'entrenament combinat.

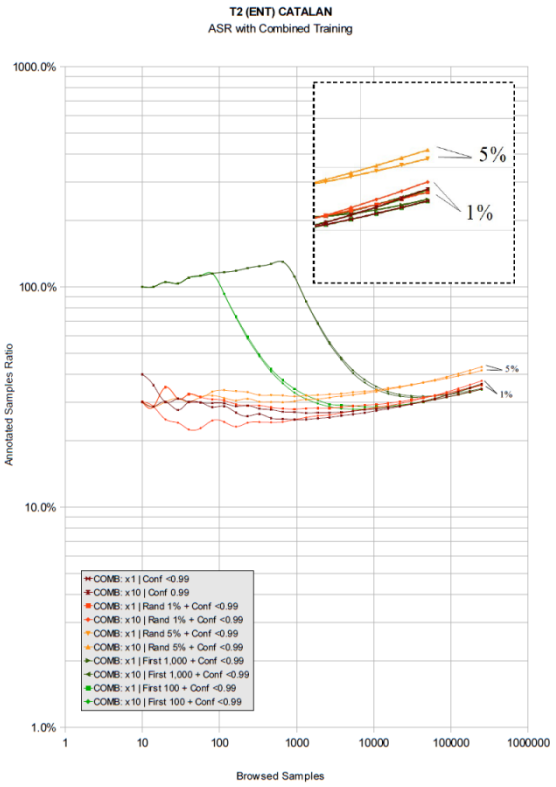

FIG. 128: Evolució de l'eficiència de T2(ENT) durant l'entrenament combinat.

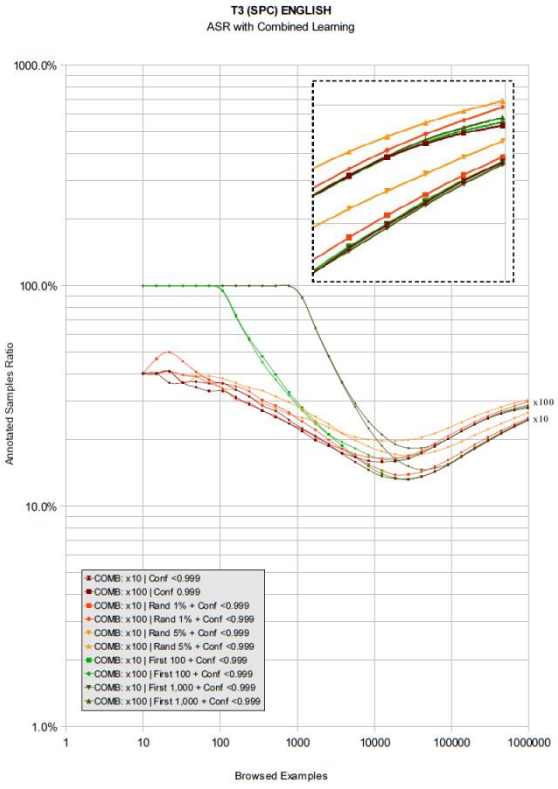

FIG. 129: Evolució de l'eficiència de T3(SPC) durant l'entrenament combinat.

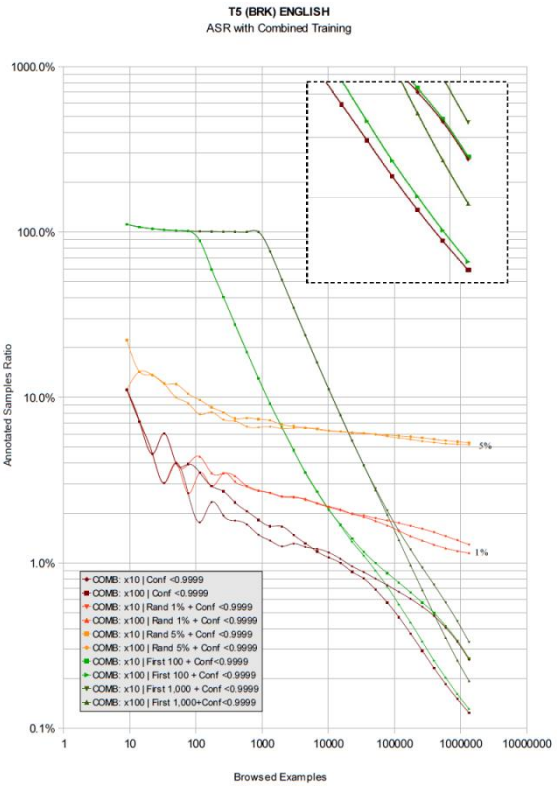

FIG. 130: Evolució de l'eficiència de T5(BRK) durant l'entrenament combinat.

15.8 CONCLUSIONS

Les dades experimentals confirmen que l'entrenament actiu permet reduir progressivament l'*ASR*, la proporció d'exemples que cal anotar d'un corpus determinat. En condicions normals, a mesura que el model millora i el classificador guanya experiència, més exemples es consideren redundats i cal anotar menys exemples. Però per diferents motius aquest valor no acostuma a baixar indefinidament, i acostuma a assolir uns valors finals que poden situar-se fàcilment al voltant del 20% o, segons la tasca i el grau de redundància del corpus, a valors inferiors a l'1%.

Un dels motius evidents que limiten l'*ASR* mínim és el valor del paràmetre `Rand` en l'entrenament actiu amb mostreig aleatori; per això és recomanable no utilitzar mostrejos superiors a l'1% o, directament, optar per l'endarreriment de l'entrenament actiu.

Però la causa més important que pot impedir assolir una bona eficiència és la selecció d'un valor subòptim del llindar de certesa, paràmetre `Confidence`. El motiu és que quan el model arriba al seu límit d'error, i aquest error és troba per sobre del llindar de `Confidence`, comencen a disparar-se els exemples que no arriben a aquest nivell, molts més exemples són seleccionats per ser anotats i, per tant, l'*ASR* comença a augmentar.

També cal destacar que el punt on l'*ASR* fa mínim i comença a augmentar, segons suggereixen les corbes de les tasques T2 i T3, podria ser un indicador del moment on el model comença a arribar al seu límit i el pendent decreixent de la corba d'error comença a reduir-se per saturació. De totes maneres per poder confirmar-ho, caldria estudiar aquesta hipòtesi amb molta més cura en altres tasques i mitjançat experiments específics.

Finalment, com a regla general sembla que l'entrenament combinat obté nivells d'eficiència més baixos, ja que els seus *ASR* acostumen a obtenir valors un 50% superiors als de l'entrenament actiu pur. Possiblement perquè el model pre-entrenat, tot i millorar l'error de classificació, no millora de la mateixa manera la capacitat del classificador per quantificar amb precisió el seu grau de certesa; als capítols següents s'aprofundeix una mica més al voltant de l'evolució del nivell de certesa i la seva precisió.

De totes maneres les corbes d'*ASR* de l'entrenament combinat suggereixen que aquesta tècnica d'entrenament sí ofereix un petit benefici: la possibilitat d'estalviar-se tant el mostreig aleatori (`Rand=0`) com l'endarreriment de l'entrenament actiu (`First=0`). El fet de començar l'entrenament amb un model pre-entrenat permet utilitzar-lo per seleccionar els exemples informatius des del primer moment. Aquest estalvi fa que, dintre de l'entrenament combinat, els millors resultats els obtingui aquesta variant que prescindeix tant del mostreig aleatori com de l'endarreriment.

La principal conclusió que es pot extreure dels experiments mostrats en aquest capítol és que la selecció del valor del paràmetre `Confidence` és crític per obtenir un model precís i eficient: un llindar excessivament baix (<0,9) impedeix minimitzar l'error i assolir classificadors precisos, però un llindar excessivament alt (>0,999) en relació a la dificultat

de la tasca impedeix minimitzar l'*ASR* i aconseguir entrenaments eficients. Queda oberta la possibilitat d'investigar tècniques d'entrenament actiu amb llindars de certesa variables en el temps, de manera que el sistema ajusti dinàmicament aquest valor de manera automàtica per minimitzar l'error final i maximitzar l'eficiència de l'entrenament.

16 DISTRIBUCIÓ DEL NIVELL DE CERTESA

16.1 INTRODUCCIÓ

Seguint la línia d'intentar entendre amb més profunditat el que succeeix dins del classificador al llarg d'un entrenament actiu, es modifica l'entorn d'entrenament perquè, aprofitant les fases d'avaluació, obtingui una sèrie de mesures dels graus de certesa amb què el classificador assigna les etiquetes.

L'objectiu és comprovar si, a mesura que avança l'entrenament, no només es redueix l'error de classificació sinó si també augmenta el grau de certesa amb què realitza les classificacions. Aquest augment dels nivells de certesa justificaria que augmentés la quantitat d'exemples classificats amb un grau de certesa superior al llindar de `Confidence` i, per tant, causaria la disminució de l'*ASR*.

En aquest capítol es descriu la metodologia utilitzada per quantificar aquest nivell de certesa i representar-ho gràficament. Es mostren les gràfiques de quatre entrenaments actius representatius, un per a cada tasca, i s'analitzen els resultats.

16.2 DISTRIBUCIÓ DEL NIVELL DE CERTESA

El nivell de certesa és una propietat associada a cada una de les classificacions individuals feta pel model. Per obtenir valors estadísticament significatius és important quantificar de forma agregada el nivell de certesa del conjunt dels exemples. Però promitjar aritmèticament els diferents valors podria ocultar fenòmens interessants i distorsionar la interpretació dels resultats. Per això es va optar per obtenir un histograma que comptabilitzés les freqüències relatives dels llindars de certesa situats en determinats intervals.

A les primeres proves es va generar un histograma lineal amb 10 segments situats entre un 0% i un 100% de certesa, però els resultats no eren gaire informatius ja que la pràctica totalitat de les classificacions obtenien un grau de certesa superior al 90%.

Histograma	**Llindars**										
Lineal	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Logarítmic	0%	-	-	-	50%	90%	99%	99.9%	99.99%	99.999%	100%

TAULA 66: Valors llindar corresponents als histogrames lineal i logarítmic del nivell de certesa.

Així doncs, es van definir uns llindars equidistants en escala logarítmica, de manera que permetessin discriminar graus de certesa molt elevats. A més, donat que pràcticament no apareixien graus de certesa inferiors a un 50%, tots aquests es van agrupar en un únic segment. Els llindars utilitzats en la creació del histograma lineal i logarítmic poden veure's a la **[Taula 66]**. Els límits inferiors i superiors dels intervals corresponents i el símbols utilitzats a les llegendes de les gràfiques d'aquest capítol poden consultar-se a la [**Taula 67**]:

Histograma	**Segments**						
Llindars	50%	90%	99%	99.9%	99.99%	99.999%	100%
Símbol	<50%	<90%	<99%	<99.9%	<99.99%	<99.999%	<100%
Segments	**0%-50%**	**50%- 90%**	**90%- 99%**	**99%- 99.9%**	**99.9%- 99.99%**	**99.99%- 99.999%**	**99.999- 100%**

TAULA 67: Símbols utilitzats a les gràfiques i intervals coberts als segments de l'histograma logarítmic.

A partir d'aquests intervals, i per a cada una de les avaluacions realitzades al classificador, s'obté un histograma que representa com es distribueixen els graus de certesa assignats a cada una de les classificacions. La **[Fig. 131]** mostra un exemple genèric d'un histograma on, després d'haver estat entrenat amb 1000 exemples, gairebé la meitat dels exemples utilitzats durant l'avaluació (49%) han estat etiquetats amb un grau de certesa comprès entre el 99,999% i el 100%.

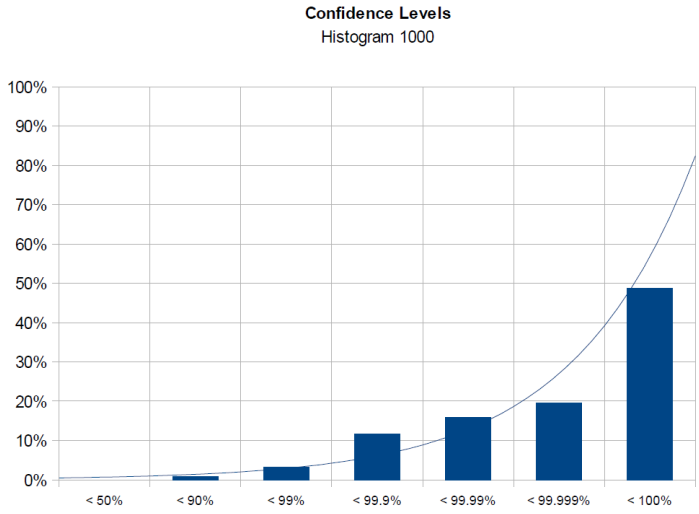

FIG. 131: Exemple genèric de distribució dels nivells de certesa d'una avaluació.

Si es concatenen en una sèrie els histogrames obtinguts en les avaluacions realitzades al llarg de tot l'entrenament, s'obté un gràfic tridimensional que mostra la progressió del nivell de certesa pel mateix conjunt de dades de referència.

Es pot veure un exemple d'aquest gràfic a la **[Fig. 132]**, on es pot observar com durant tot l'entrenament els valors superiors al 99,999%, sèrie blava, són majoritaris, o també com el

nombre d'exemples classificats amb nivells de certesa inferiors al 90%, sèries taronja i groga, segueixen camins decreixents a mesura que el classificador va aprenent.

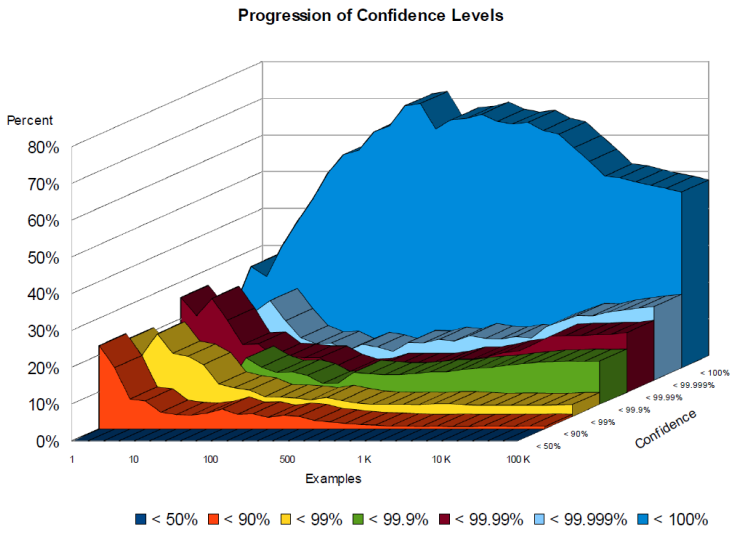

FIG. 132: Evolució genèrica al llarg d'un entrenament (*Examples*) de la distribució dels exemples (*Percent*) per a diferents nivells de certesa (*Confidence*).

16.3 EVOLUCIÓ EN LES DIFERENTS TASQUES

En aquesta secció es comenten algunes característiques de l'evolució del nivell de certesa per a les quatre tasques representatives durant uns entrenaments actius, tots ells amb un mostreig aleatori de l'1% però amb diferents llindars de certesa.

Els histogrames superiors, [**Fig. 133**; **Fig. 135**; **Fig. 137**; **Fig. 139**], mostren la distribució dels nivells de confiança en quatre moments puntuals de l'entrenament. L'eix vertical, escala lineal de 0% a 100%, indica el tant per cent d'exemples anotats amb un determinat interval de nivell de certesa; l'eix horitzontal indica el límit superior d'aquest interval. La línia vertical discontinua de color blau marca la referència del valor utilitzat pel paràmetre `Confidence` en cada una de les tasques.

Les gràfiques tridimensionals inferiors, **[Fig. 134; Fig. 136; Fig. 138; Fig. 140]**, mostren la progressió d'aquests histogrames. L'eix horitzontal etiquetat com a *Examples* fa referència a la quantitat d'exemples seleccionats amb els que ha estat entrenat el model, un número molt inferior al dels exemples analitzats; l'eix vertical *Percent* la proporció d'exemples amb aquest nivell de confiança; i l'eix de profunditat *Confidence* el límit superior del nivell de certesa corresponent.

Als següents apartats es mostren els comentaris (pàgines esquerra) i les gràfiques (pàgines dreta) dels resultats per a cada una de les tasques.

EVOLUCIÓ DE T1 (POS)

A la **[Fig. 133]** es poden observar quatre histogrames corresponents a diferents moments de l'entrenament de la tasca T1(POS). En un primer moment, abans que el classificador hagi començat a veure exemples, s'observa com els nivells de certesa són molt extrems, molts d'ells del 100%. Però a mesura que l'entrenament avança apareixen els matisos fins que en el quart histograma s'observa com segueixen valors creixents segons el grau de certesa.

Si observem la **[Fig. 134]** es pot veure com l'evolució és molt suau i no presenta cap variació abrupta. Cal recordar que es tracta d'una tasca multietiqueta amb un entrenament incomplet que lluny d'arribar a saturar podia haver continuat amb un corpus diverses vegades més gran.

[Espai intencionadament en blanc per alinear el text amb les figures.]

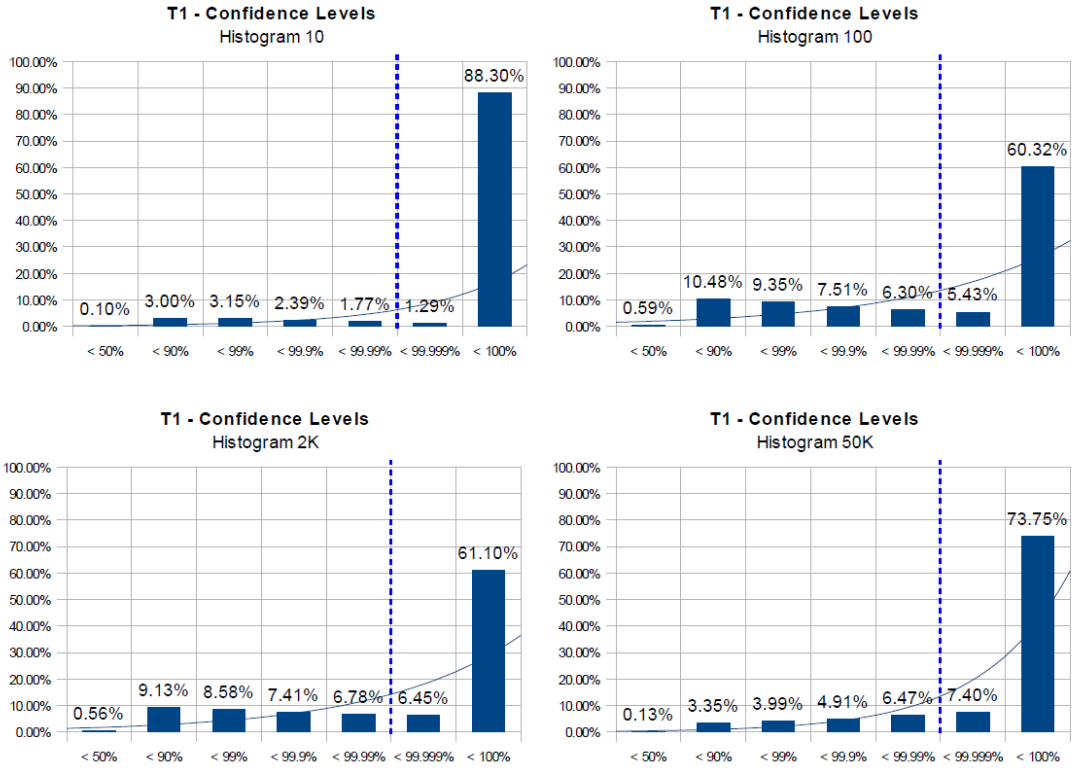

FIG. 133: Distribució del nivell de certesa en diferents moments d'un entrenament de T1(POS).

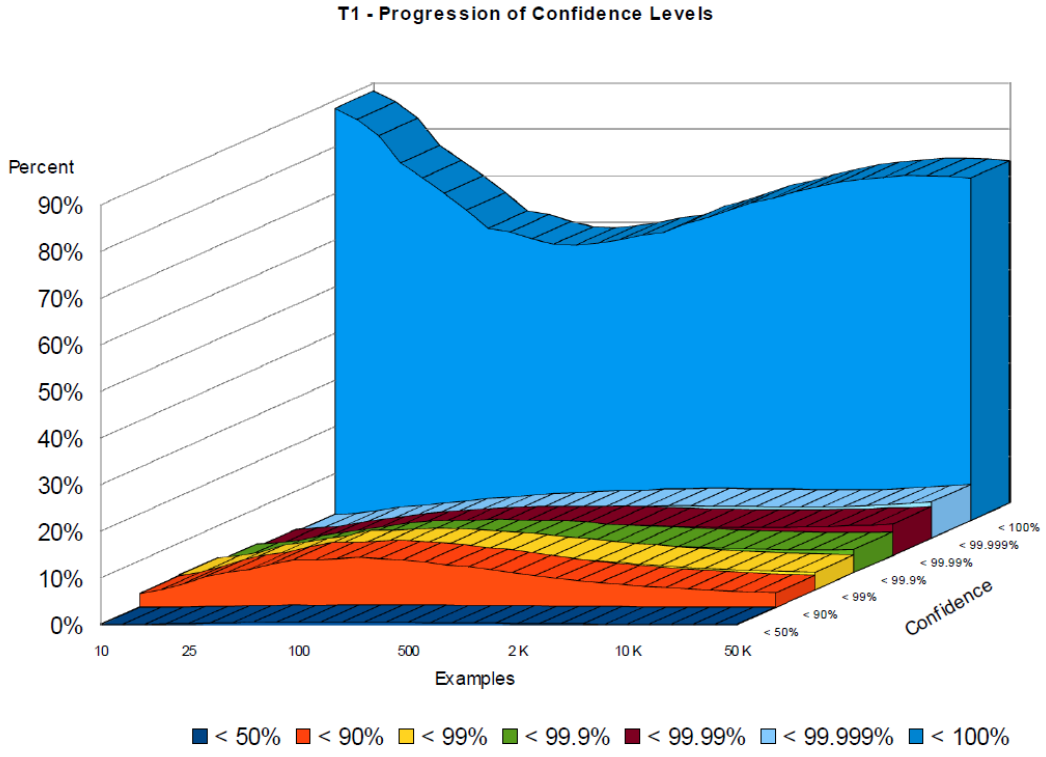

FIG. 134: Evolució del nivell de certesa al llarg d'un entrenament de T1(POS).

EVOLUCIÓ DE T2 (ENT)

A la **[Fig. 135]** s'observen els quatre histogrames corresponents a la tasca T2(ENT). En aquest cas la polarització inicial dels valors de certesa és molt extrema, però a mesura que avança l'entrenament la predominança de la franja de més certesa es va diluint fins que al final el valor més freqüent se situa en la franja 99%-99,9%, just després del llindar de `Confidence`, on coincideixen 1 de cada 3 exemples (34,65%).

En observar **[Fig. 136]** es veu com l'entrenament no s'inicia significativament fins apropar-se al primer centenar d'exemples. A partir d'aquest punt la tendència de cada una de les franges sembla molt clara: la franja de més certesa decreix de manera continuada, mentre que la franja de 99%-99,9%, de color verd, augmenta fins convertir-se en el pic de l'histograma.

És interessant que en aquest tasca on el model sí que començava a saturar i, per tant a estabilitzar-se, la moda de nivell de certesa coincideixi amb el llindar de certesa controlat pel paràmetre `Confidence`. A les conclusions del capítol s'analitza una possible causa d'aquesta coincidència.

[Espai intencionadament en blanc per alinear el text amb les figures.]

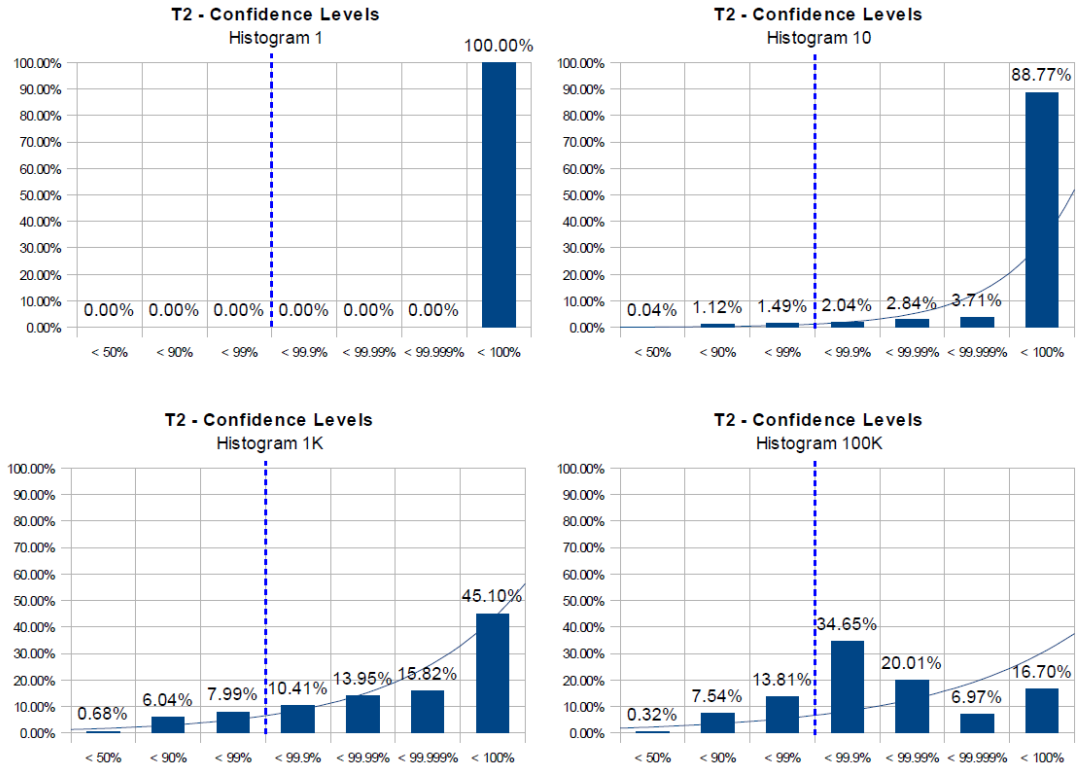

FIG. 135: Distribució del nivell de certesa en diferents moments d'un entrenament de T2(ENT).

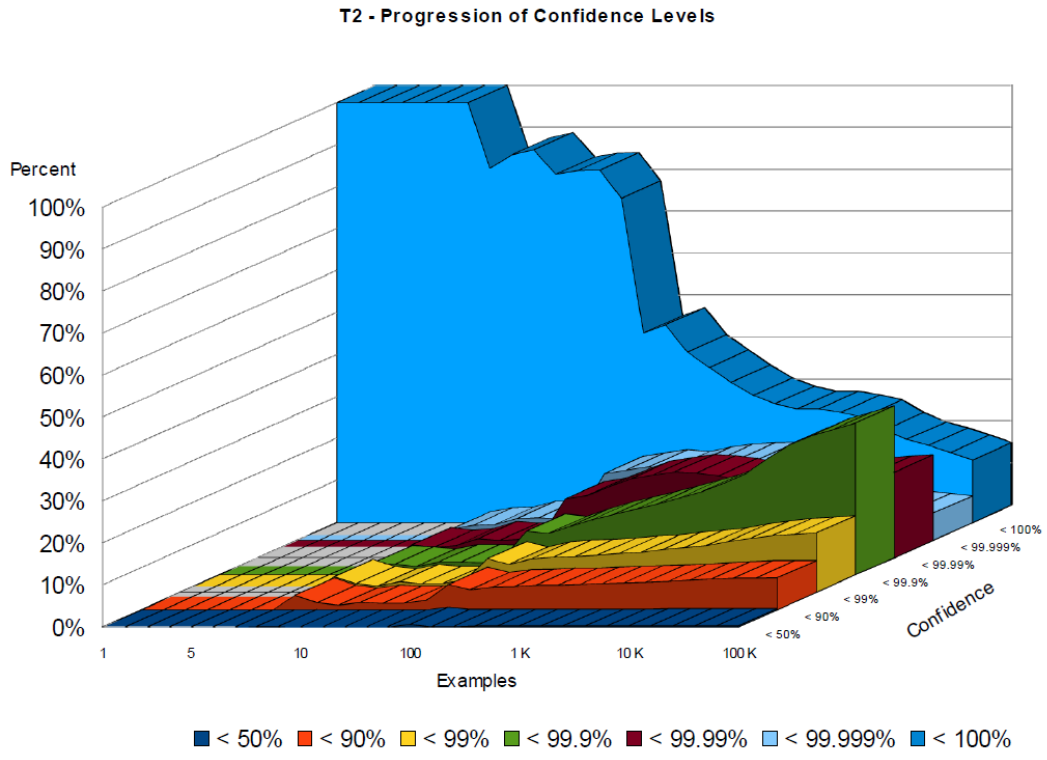

FIG. 136: Evolució del nivell de certesa al llarg d'un entrenament de T2(ENT).

EVOLUCIÓ DE T3 (SPC)

A continuació es pot observar com a la **[Fig. 137]** la tasca T3(SPC) es comporta de manera molt semblant. El que inicialment és un histograma molt polaritzat es va difuminant progressivament a mesura que el model millora. Fins que, com es veu en el quart histograma, la distribució final presenta un pic central, en aquest cas a la franja 99,9%-99,99% de color grana, on cauen bona part dels exemples (41,67%).

La progressió general de la **[Fig. 138]** es molt similar, fins als primers centenars d'exemples l'evolució no és clara, però a partir d'aquest punt s'observa clarament la mateixa tendència: la franja de més certesa decreix de manera continuada, dues franges centrals (color verd i grana) que cobreixen l'interval 99%-99,99% creixen lentament, i la resta de franges continuen decreixent.

Seguint el mateix patró que la tasca anterior i tractant-se també d'un model saturat i estable, es torna a donar la coincidència que la distribució final dels nivells de certesa s'apropa a una distribució normal al voltant del llindar de certesa utilitzat durant l'entrenament. A les conclusions del capítol s'analitza una possible causa d'aquesta coincidència.

[Espai intencionadament en blanc per alinear el text amb les figures.]

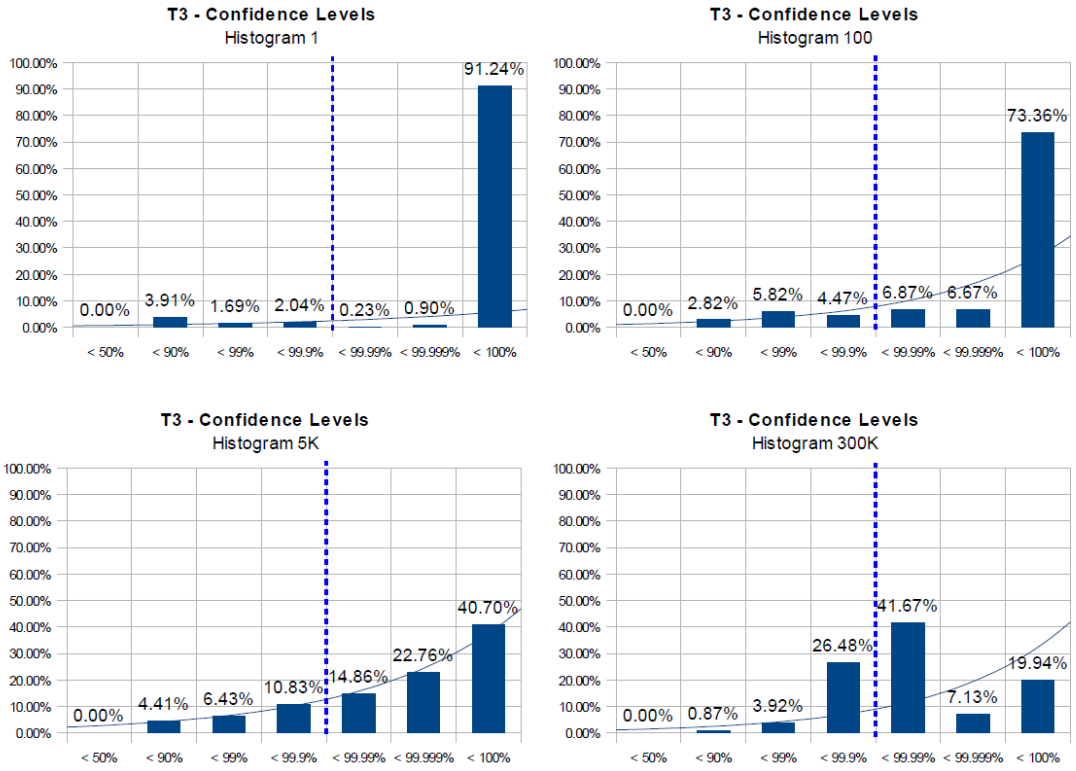

FIG. 137: Distribució del nivell de certesa en diferents moments d'un entrenament de T3(SPC).

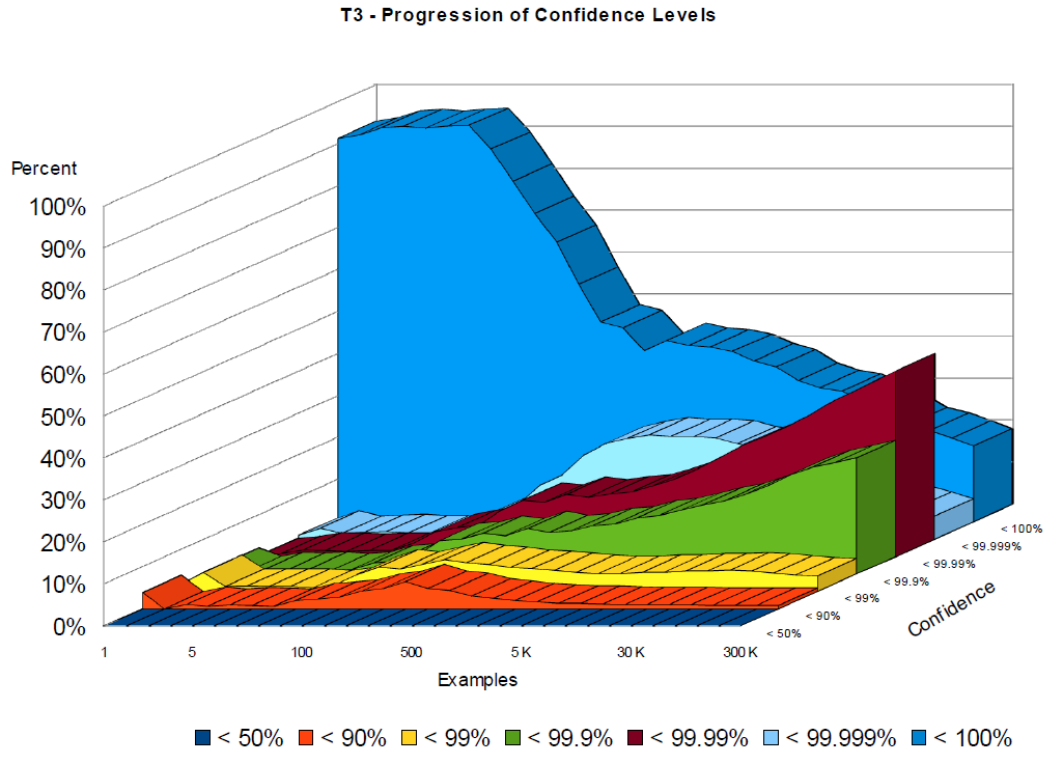

FIG. 138: Evolució del nivell de certesa al llarg d'un entrenament de T3(SPC).

EVOLUCIÓ DE T5 (BRK)

Finalment, els resultats de la tasca T5(BRK) mostrats a la **[Fig. 139]** indiquen un comportament molt diferent al dels casos anteriors. La polarització inicial del primer histograma, que en els dos histogrames següents sembla que pot relaxar-se, acaba reforçant-se i assolint una polarització pràcticament absoluta, on el 98,72% dels exemples s'anoten amb una certesa superior al 99,999%. Sembla clar que la senzillesa de la tasca i la gran quantitat d'exemples permet al classificador modelar de manera gairebé perfecta les dades.

L'evolució global de la **[Fig. 140]** indica que tot i seguir un procés similar a les dues tasques anteriors al voltant dels 500 exemples, on la franja de certesa més elevada sembla disminuir i les dues centrals augmentar lleugerament, acaba transformant-se en una distribució "ideal" on totes les classificacions les fa amb una certesa màxima. A diferència de en les dues tasques anteriors, en aquest cas, el llindar de certesa utilitzat (99,9%) no sembla haver tingut cap influència.

[Espai intencionadament en blanc per alinear el text amb les figures.]

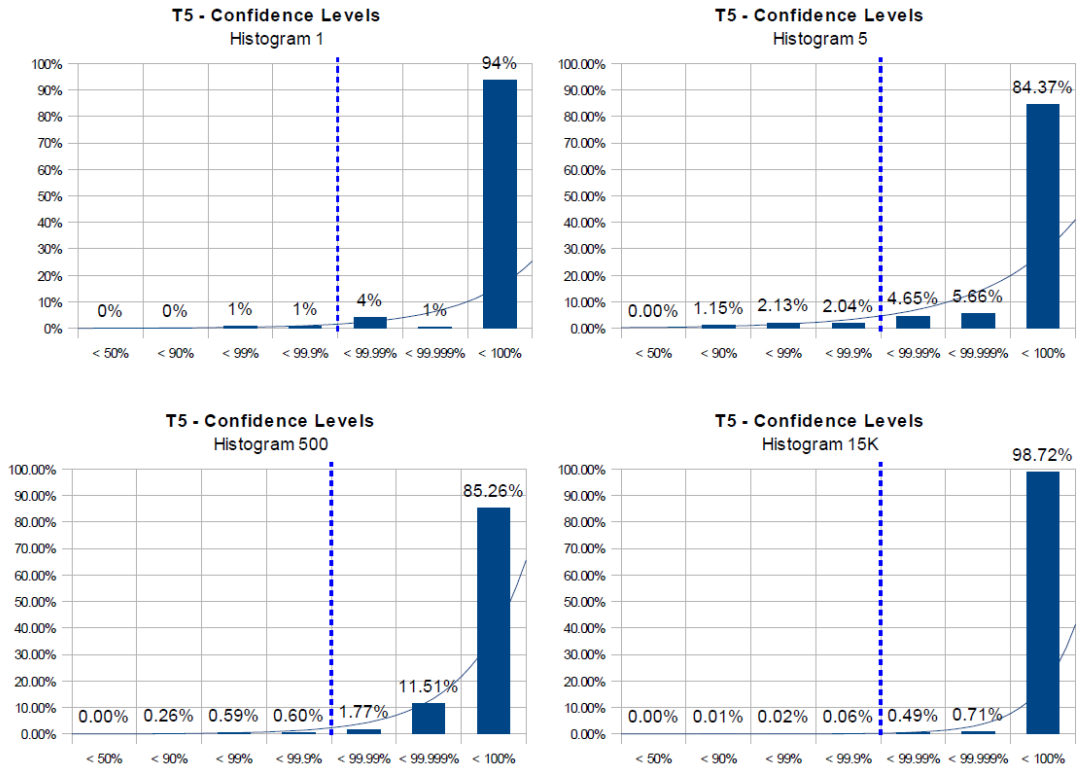

FIG. 139: Distribució del nivell de certesa en diferents moments d'un entrenament de T5(BRK).

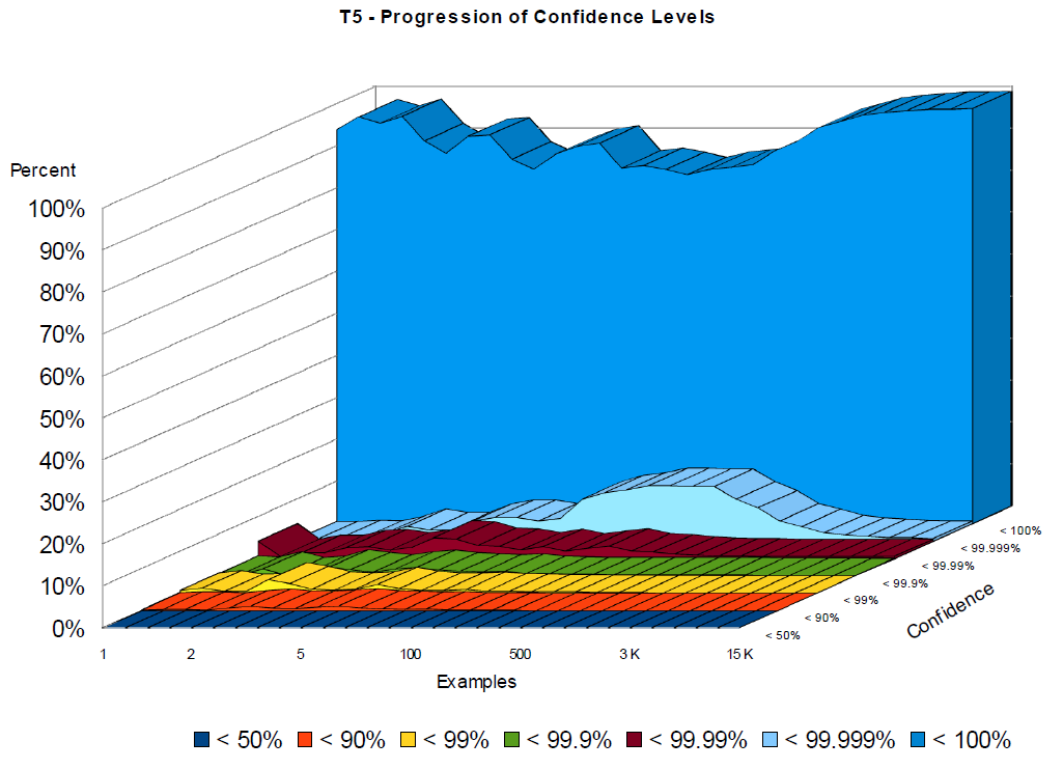

FIG. 140: Evolució del nivell de certesa al llarg d'un entrenament de T5(BRK).

16.4 CONCLUSIONS

En aquest capítol s'ha volgut confirmar empíricament una idea que semblava raonable: que a mesura que un model és entrenat i redueix l'error de classificació, augmenta la certesa amb la que classifica els exemples. És aquest augment dels nivells de certesa el que fa créixer el nombre d'exemples que superen el llindar i per tant el que permet al classificador, a mesura que guanya experiència, descartar més exemples i augmentar l'eficiència.

Tot i que no pugui generalitzar-se a tots els algorismes, sembla que el *Naïve Bayes* presenta una tendència a polaritzar els graus de certesa, ja que tant a l'inici de l'entrenament com al final d'un entrenament complert (on ha modelitzat les dades perfectament), s'observa una distribució gens homogènia amb un únic pic en la franja de certesa més elevada.

També sembla que quan un model satura hi ha una coincidència entre el valor de certesa més freqüent i el valor del llindar de certesa utilitzat per l'algorisme d'entrenament actiu. Tot i que el més és raonable que la causalitat sigui en direcció contrària, és a dir, que el llindar de certesa òptim[1] sigui aquell que coincideix amb el valor de certesa més freqüent, ja que és el punt on maximitza la quantitat d'exemples descartats.

Cal recordar que en aquest capítol s'ha analitzat el nivell de confiança calculat pel propi classificador, un nivell de confiança "subjectiu" que pot coincidir o no amb la probabilitat d'etiquetar correctament un exemple determinat; aquest punt és precisament el que s'intenta esbrinar en el proper capítol.

[1] Cal recordar que els llindars de confiança triats per aquests quatre experiments eren els que havien obtingut millors resultats, en error i eficiència, de totes les variants i proves anteriors.

17 EXACTITUD DEL NIVELL DE CERTESA

17.1 INTRODUCCIÓ

La capacitat d'un classificador per estimar el grau de certesa real davant la classificació d'un exemple és determinant per poder aplicar qualsevol tècnica d'entrenament actiu. Si la certesa calculada pel classificador no reflectís la realitat l'entrenament actiu podria descartar exemples valuosos i centrar-se en exemples redundants, cosa que afectaria tant l'error final com a l'eficiència de l'entrenament.

Per això, s'ha volgut mesurar la precisió d'aquestes estimacions. En aquest capítol es mostren els resultats d'una ampliació feta als quatre experiments anteriors, en la que s'han analitzat les dades i calculat diverses mesures per determinar la proporció d'encerts i d'errors per a diferents nivells de certesa.

En les següents seccions s'expliquen les metodologies utilitzades per obtenir aquestes mesures així com la seva representació gràfica. Posteriorment es comenten els resultats dels experiments de les quatre tasques de referència i s'extreuen algunes conclusions interessants.

17.2 PRECISIÓ SEGONS EL NIVELL DE CERTESA

El plantejament inicial va ser assumir que existia una correspondència entre el nivell de certesa estimat pel classificador i la probabilitat real que la classificació fos correcta. És a dir, que pel conjunt de classificacions amb un determinat nivell de certesa, aquest nivell coincidia amb el tant per cent de classificacions correctes.

Per tant, mantenint els intervals utilitzats als histogrames anteriors, es va calcular la precisió mitjana en cada un dels intervals. Per exemple, l'interval central que cobreix un nivell de certesa d'entre un 99% i un 99,9%, hauria d'obtenir una precisió situada entre aquests dos valors, que en el cas de distribuir-se homogèniament suposaria classificar correctament el 99,45% d'exemples situats en aquests nivells de certesa.

Precisió i Intervals de Certesa del Histograma							
Símbol	<50%	<90%	<99%	<99.9%	<99.99%	<99.999%	<100%
Certesa	0% - 50%	50% - 90%	90% - 99%	99% - 99.9%	99.9% - 99.99%	99.99% - 99.999%	99.999% - 100%
Mitjana	**25%**	**70%**	**94.5%**	**99.45%**	**99.945%**	**99.9945**	**99.9995**

TAULA 68: Precisió esperada idealment en cada un dels segments per diferents nivells de certesa.

A la taula anterior **[Taula 68]** es mostren els límits de tots els intervals i les precisions mitjanes esperades a cada un.

Durant l'avaluació del classificador, a més d'obtenir els histogrames corresponents a la seva distribució, es van comptabilitzar per a cada interval el nombre de classificacions correctes, que dividits entre el nombre de classificacions va permetre obtenir la precisió o tant per cent d'encerts obtingut. Per poder observar la seva similitud, aquests valors reals es van comparar amb els "ideals" o esperats: com més semblants siguin els valors obtinguts i els esperats, més exactes seran els nivells de certesa.

Per comparar visualment aquests valors s'ha utilitzat un diagrama de barres, veure **[Fig. 141]**, on per cada franja de nivell de certesa es mostra: la precisió esperada o ideal (en verd), la precisió obtinguda o real (en taronja), i la diferència entre els dos valors (en vermell).

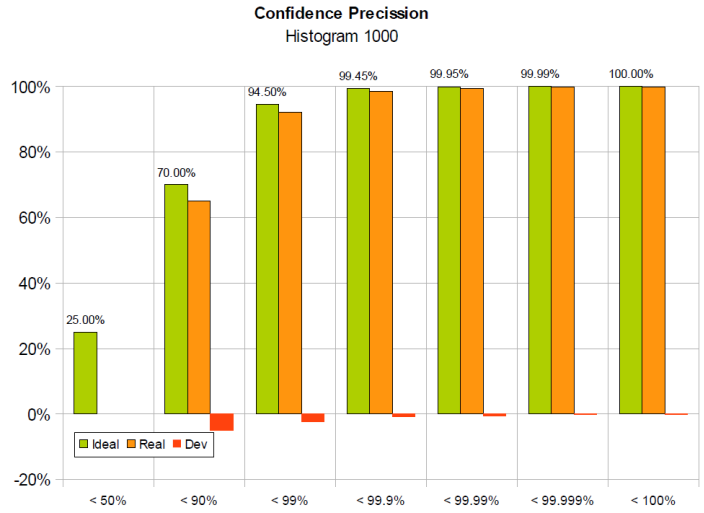

FIG. 141: Exemple genèric de comparació de la precisió ideal i observada segons els nivells de certesa.

A la següent secció es mostren, per a cada un dels quatre experiments, l'histograma comparatiu en diferents moments de l'entrenament. En ells es pot veure com les precisions obtingudes van augmentant progressivament fins apropar-se a les precisions esperades, veure **[Fig. 142; Fig. 144; Fig. 146; Fig. 148]**.

Pera a cada tasca, a sota de cada grup d'histogrames, s'inclou també una representació tridimensional de l'evolució de la precisió per cada interval de nivell de certesa, veure **[Fig. 143; Fig. 145; Fig. 147; Fig. 149]**.

Tot i que es pot observar com la precisió augmenta per cada un dels intervals fins assolir un valor proper a l'esperat, al final de l'entrenament, i especialment en els nivells de certesa propers al 100%, no és fàcil apreciar les diferencies entre el valor real i l'ideal. Per això es va optar per representar una mètrica complementària, veure **[17.4 Error segons el nivell de Dubte]**.

17.3 GRÀFIQUES PRECISIÓ

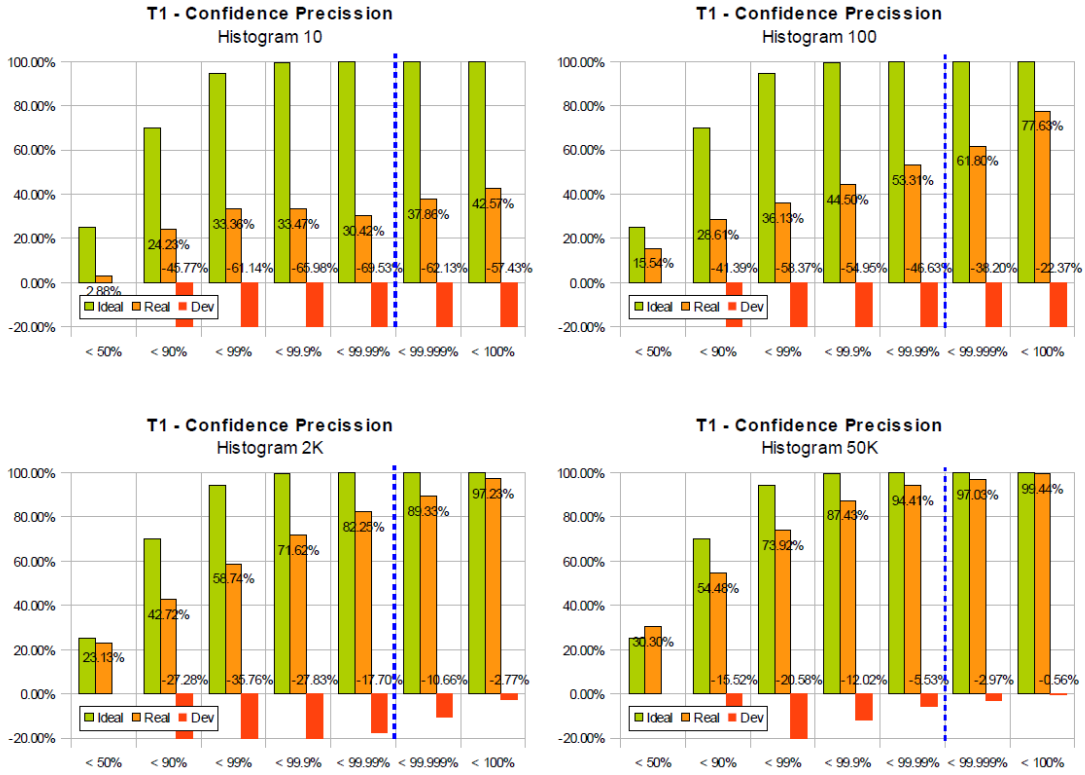

FIG. 142: Precisió esperada i observada en diferents moments d'un entrenament de T1(POS).

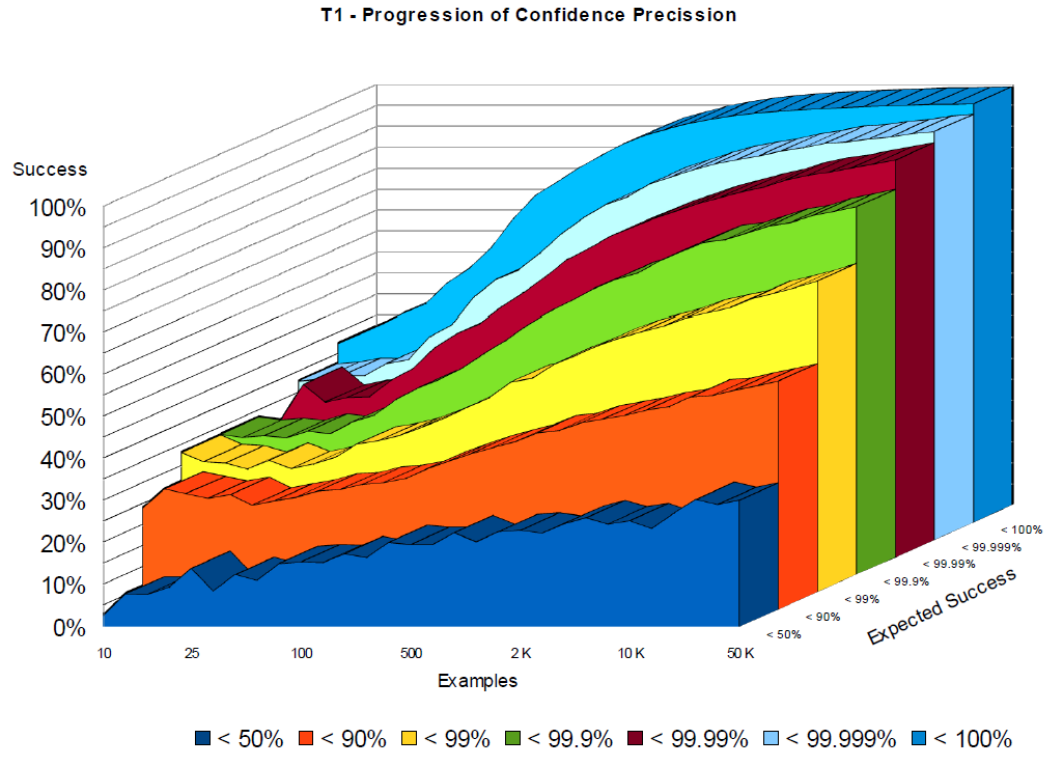

FIG. 143: Evolució de la precisió observada segons el nivell de confiança a T1(POS).

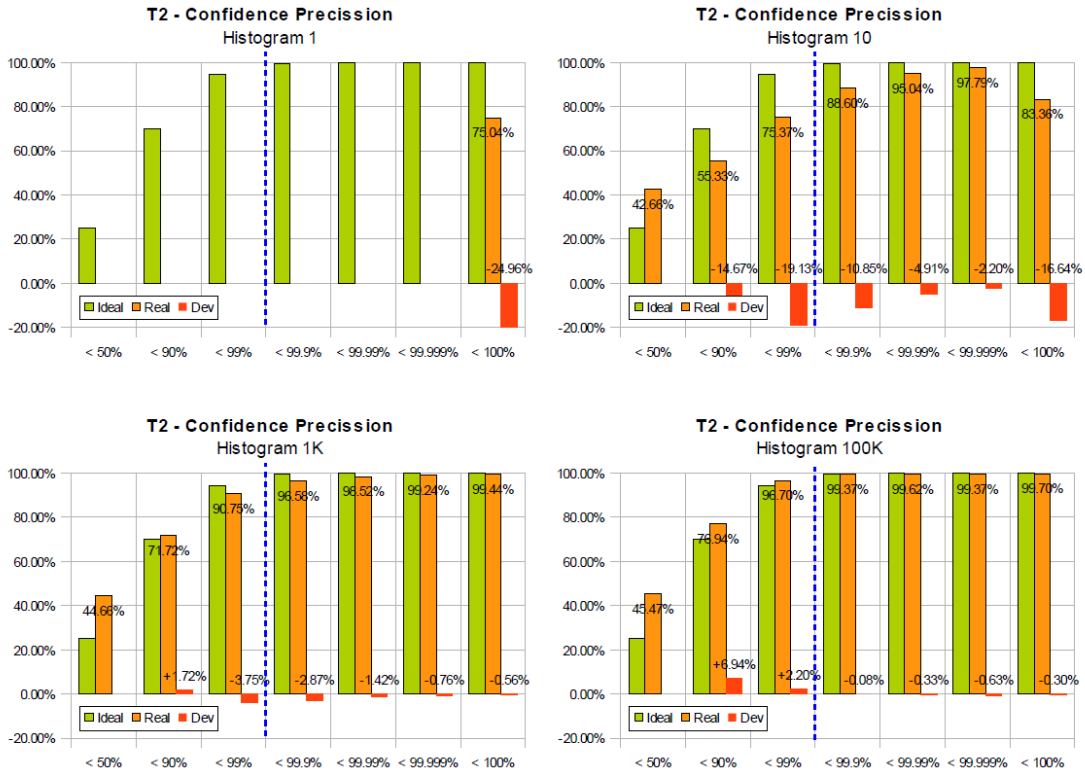

FIG. 144: Precisió esperada i observada en diferents moments d'un entrenament de T2(ENT).

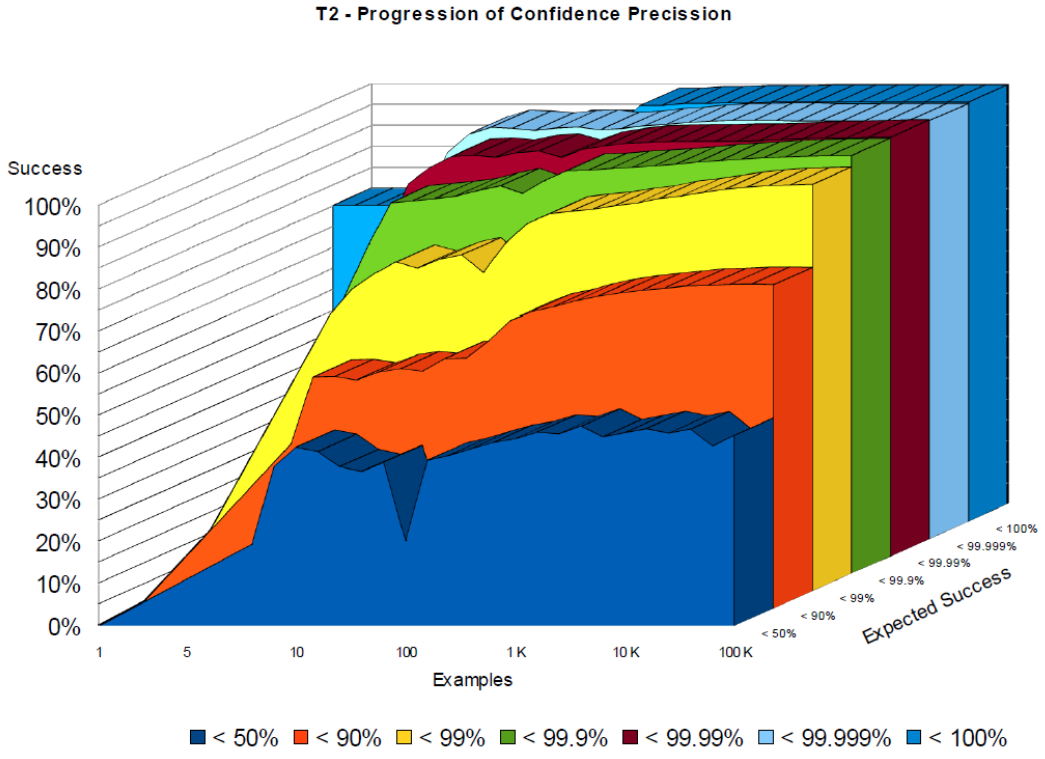

FIG. 145: Evolució de la precisió observada segons el nivell de confiança a T2(ENT).

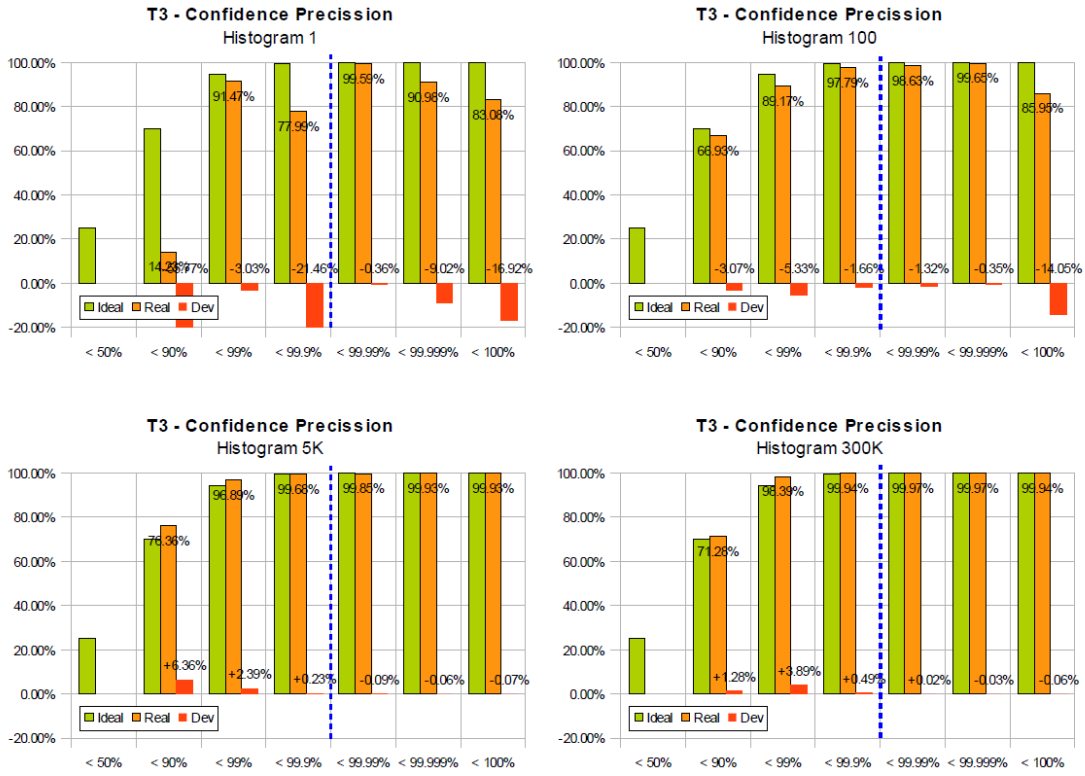

FIG. 146: Precisió esperada i observada en diferents moments d'un entrenament de T3(SPC).

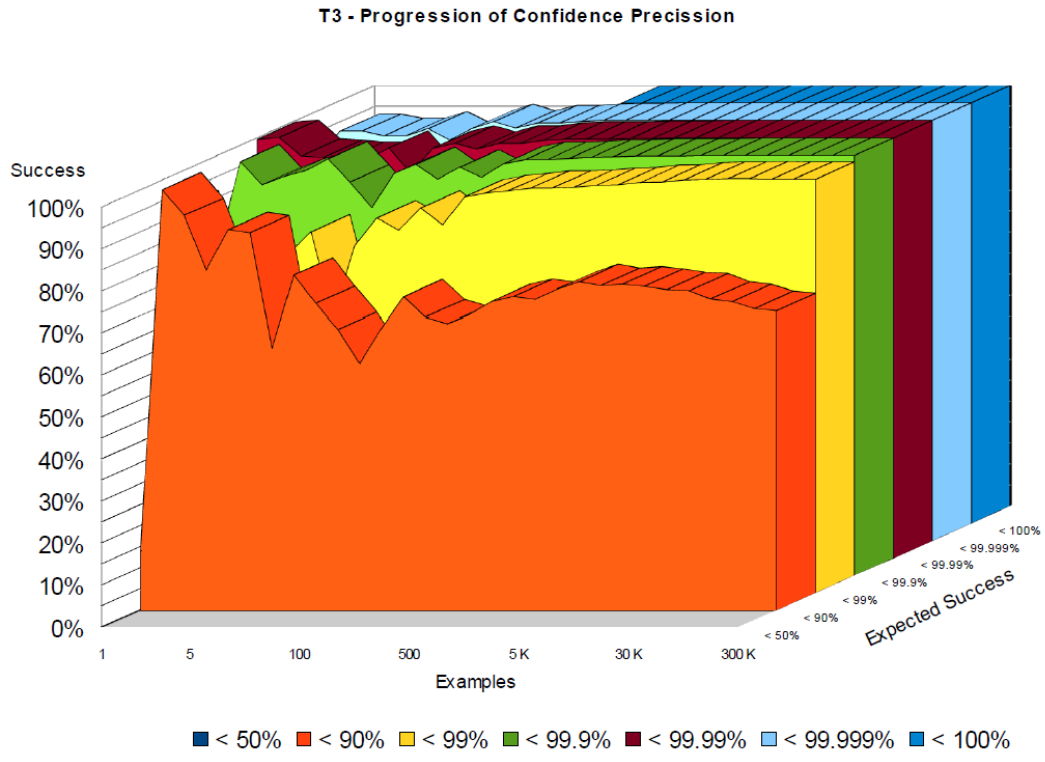

FIG. 147: Evolució de la precisió observada segons el nivell de confiança a T3(SPC).

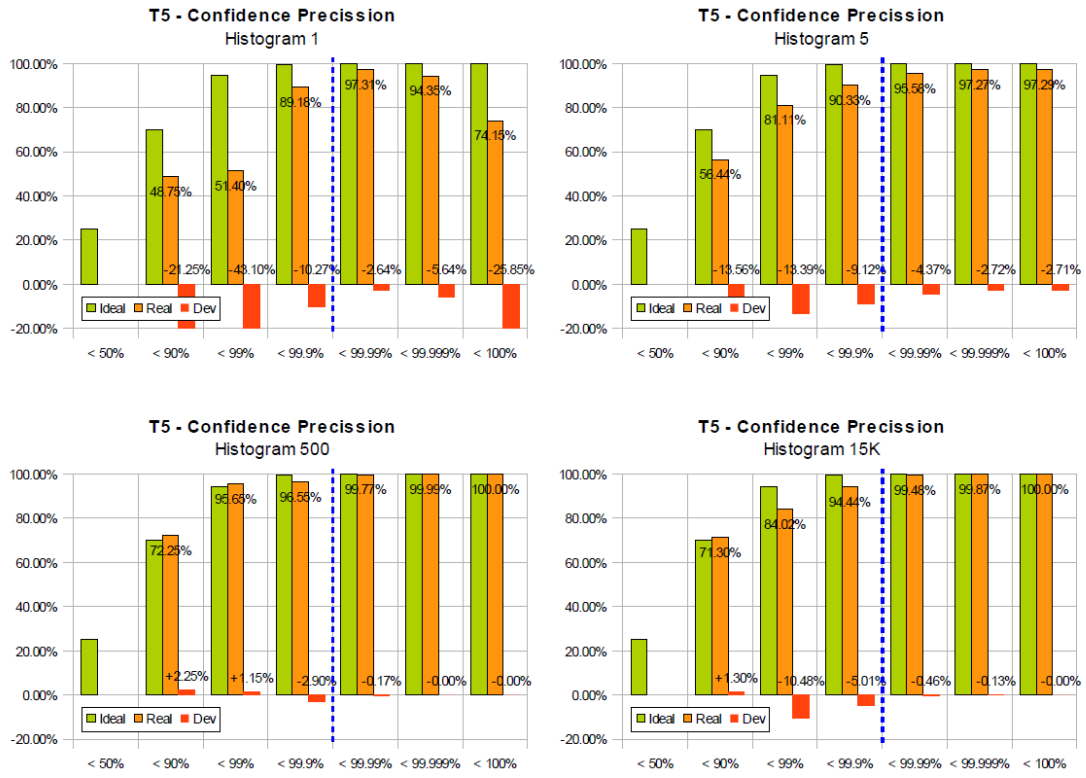

FIG. 148: Precisió esperada i observada en diferents moments d'un entrenament de T5(BRK).

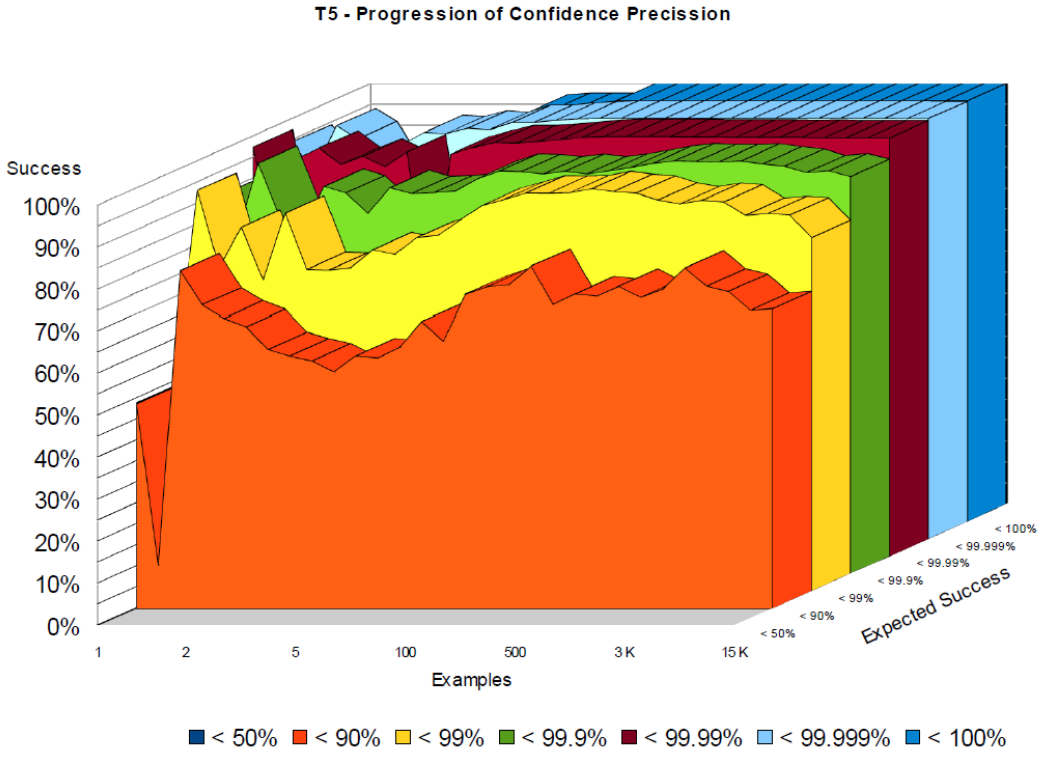

FIG. 149: Evolució de la precisió observada segons el nivell de confiança a T5(BRK).

17.4 ERROR SEGONS EL NIVELL DE DUBTE

Després d'obtenir els gràfics corresponents a la precisió i constatar que no permetien discriminar les diferències entre els valors esperats i els observats quan es tractava de valors propers al 100%, es va optar per analitzar una mètrica complementaria. És a dir, tornar a fer el mateix que amb la precisió però utilitzant l'error, és a dir, el tant per cent de classificacions incorrectes per cada interval de nivell de certesa. De la mateixa manera els nivells de certesa es transformen en el seu valor complementari com a *nivells de dubte*.

A la [**Taula 69**] es mostren els valors utilitzats, els intervals de precisió han estat transformats en els seu invers, intervals d'error, i per cada un s'han calculat els valors mitjans. Per exemple, l'interval central amb una precisió d'entre el 90% i el 99% es transforma en un interval amb un error d'entre el 10% i l'1% i, per tant, amb un error esperat del 5,5%.

Error i Intervals de Dubte del Histograma							
Símbol	<75%	<50%	<10%	<1%	<0.1%	<0.01%	<0.001%
Dubte	100% - 50%	50% - 10%	10% - 1%	1% - 0.1%	0.1% - 0.01%	0.01% - 0.001%	0.001% - 0%
Mitjana	75%	30%	5.5%	0.55%	0.055%	0.0055%	0.0005%

TAULA 69: Error esperat idealment en cada un dels segments segons el nivell de dubte.

Tot i que aparentment no suposa cap millora, la representació gràfica de l'error permet utilitzar eixos logarítmics, cosa que facilita observar clarament les petites diferències dels intervals que tenen un nivell de certesa més elevat. A la **[Fig. 150]** es mostra el diagrama equivalent de l'exemple genèric anterior, en el que s'observen clarament les diferències entre els errors esperats o ideals, barres verdes, i els errors obtinguts o reals, barres taronges.

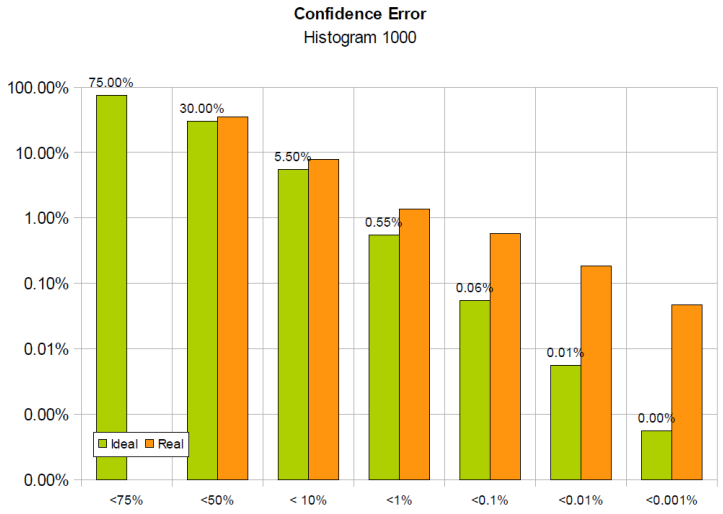

FIG. 150: Exemple genèric de comparació de l'error ideal i observat segons els nivells de dubte.

17.5 EVOLUCIÓ EN LES DIFERENTS TASQUES

Els resultats d'aquestes mesures es poden consultar en els següents apartats d'aquest capítol. Per a cada un dels quatre experiments es mostren a la part superior quatre histogrames amb escala logarítmica on es poden comparar els errors esperats i els errors obtinguts per cada un dels intervals de nivell de certesa en diferents moments de l'entrenament. A la part inferior les representacions tridimensionals que mostren l'evolució dels histogrames al llarg de tot l'entrenament.

Aquestes gràfiques permeten observar amb molt més detall la similitud entre els valors esperats i observats i, per tant, determinar si els nivells de certesa/dubte calculats pel classificador són valors propers a la realitat. A continuació s'analitzen aquests resultats per cada una de les tasques de referència i es comenten algunes conclusions.

EVOLUCIÓ DE T1 (POS)

El gràfic de la **[Fig. 151]** permet comparar l'error obtingut i l'error esperat, per cada franja de nivell de confiança, al llarg de l'entrenament de T1. S'observa que a mesura que l'entrenament avança els errors observats es redueixen i s'apropen als errors esperats. De totes maneres la distància entre ells és considerable, probablement per tractar-se d'un entrenament incomplert on l'error encara podria reduir-se considerablement.

A la **[Fig. 152]** es veu com la progressió de l'error és molt contínua i, independentment dels valors absoluts, segueix una tendència decreixent, tant respecte al nombre d'exemples de l'entrenament com respecte al nivell de dubte.

[Espai intencionadament en blanc per alinear el text amb les figures.]

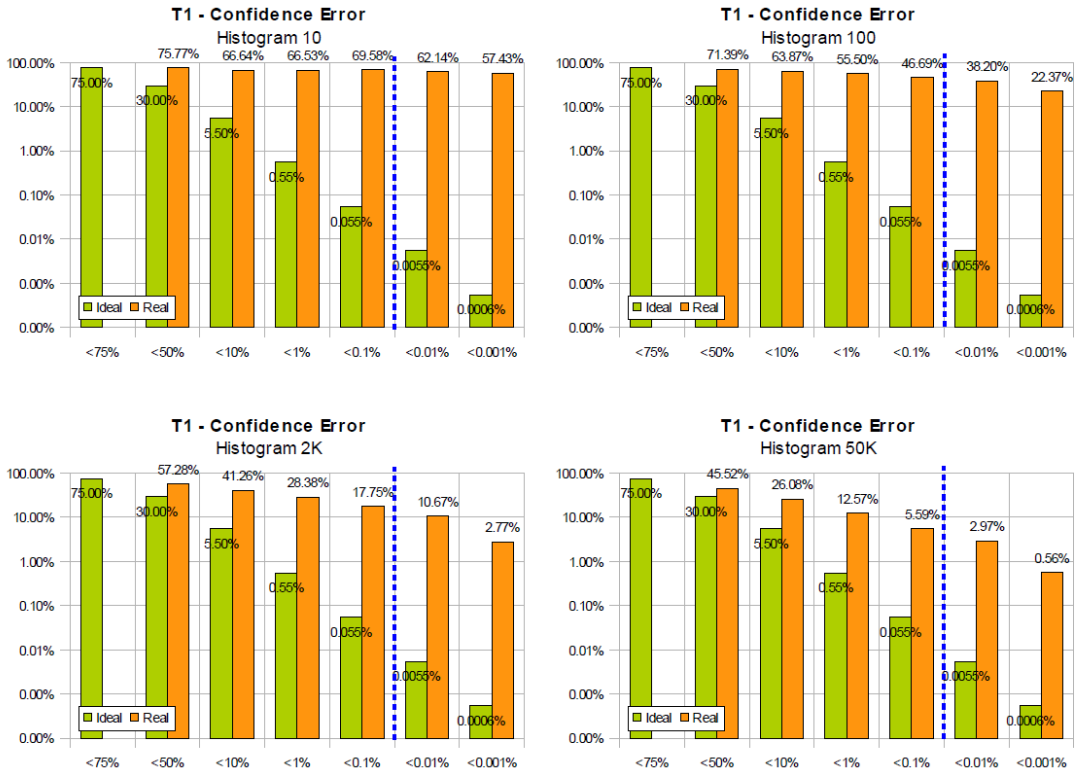

FIG. 151: Ràtios d'error esperats i observats en diferents moments d'un entrenament de T1(POS).

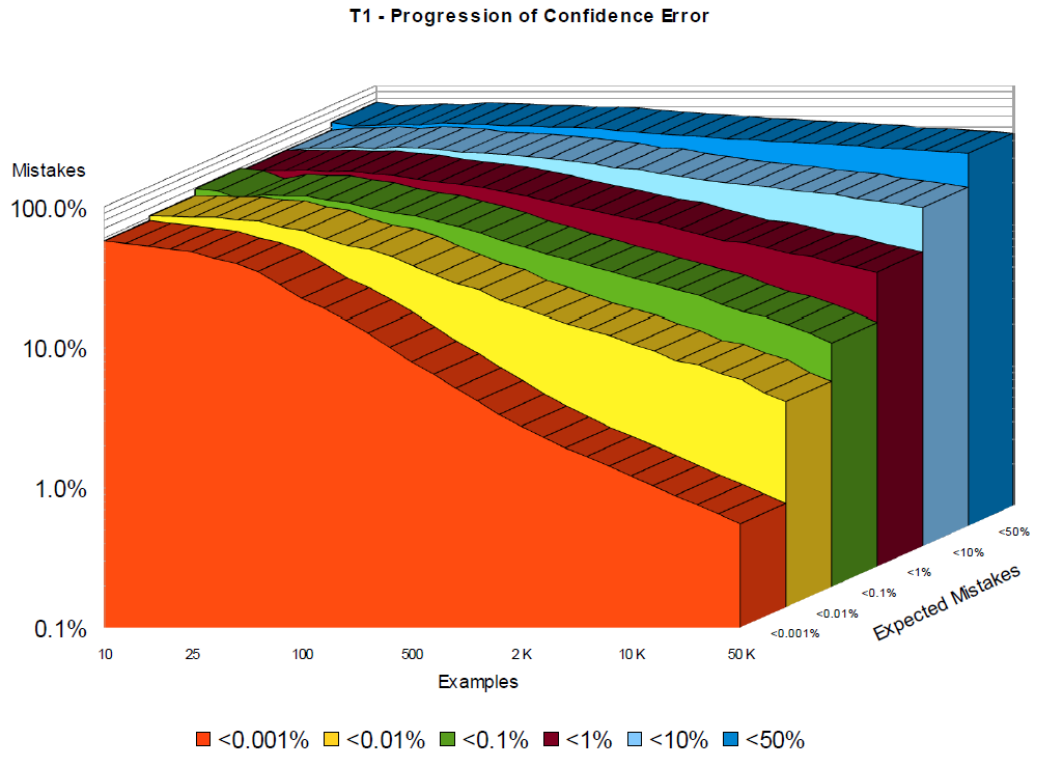

FIG. 152: Evolució del ràtio d'error observat segons el nivell de confiança a T1(POS).

EVOLUCIÓ DE T2 (ENT)

La **[Fig. 153]** mostra els resultats de l'entrenament actiu de T2; en aquest cas els errors observats s'apropen molt més als errors esperats, cal recordar que els histogrames estan representats amb escala logarítmica.

Al finalitzar l'entrenament la correspondència és total fins a un nivell de dubte inferior a un 1%, per nivells inferiors l'error esperat i l'observat es distancien. És interessant veure com a partir d'aquest punt la taxa d'error s'estabilitza al voltant del 0,5%, probablement el límit inferior de la tasca a causa de la saturació del model. L'evolució de la **[Fig. 154]** presenta alguna inestabilitat inicial, però a partir d'uns centenars d'exemples la progressió és clarament continua i decreixent.

[Espai intencionadament en blanc per alinear el text amb les figures.]

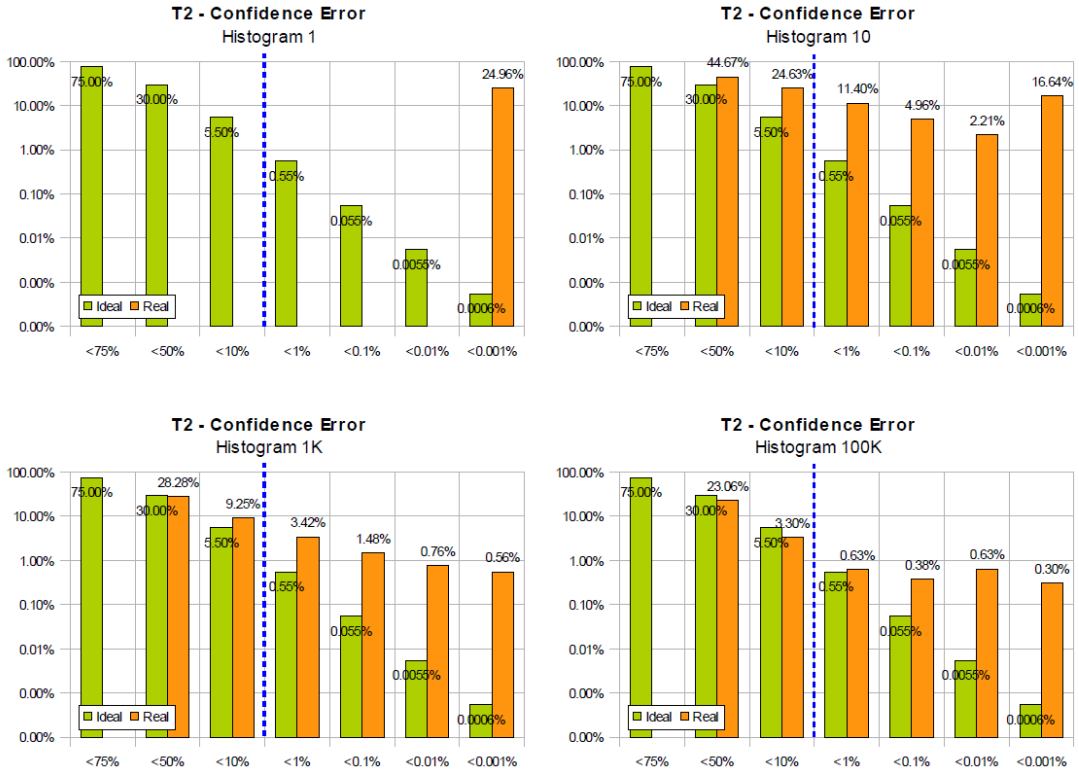

FIG. 153: Ràtios d'error esperats i observats en diferents moments d'un entrenament de T2(ENT).

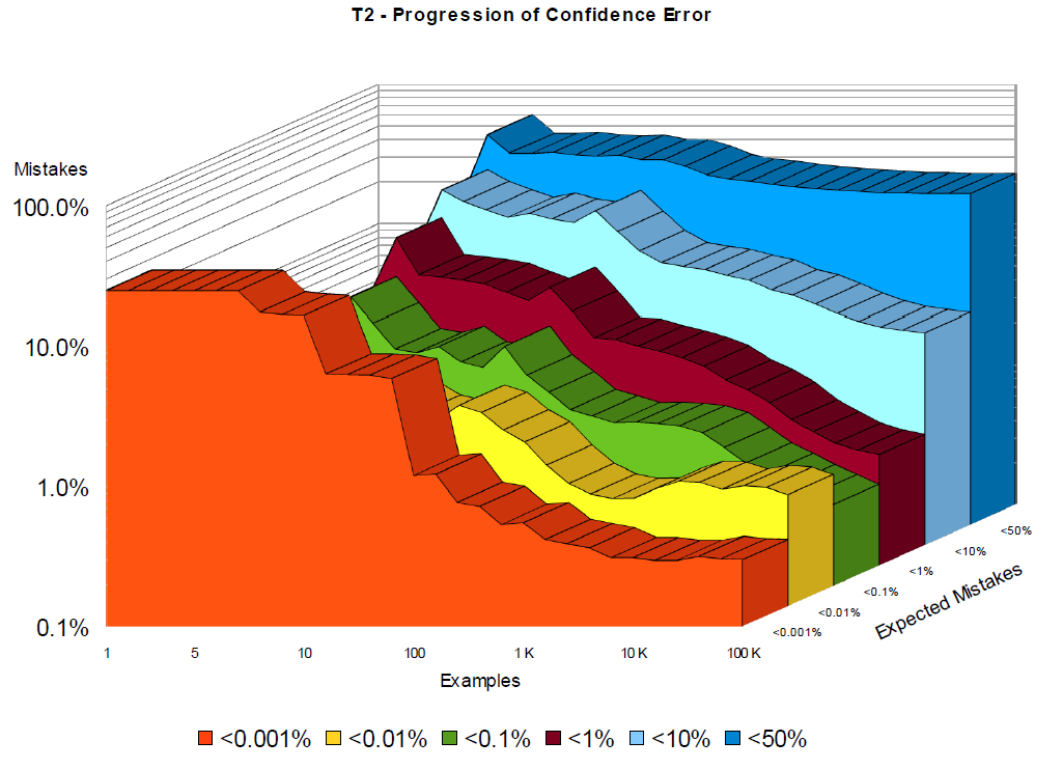

FIG. 154: Evolució del ràtio d'error observat segons el nivell de confiança a T2(ENT).

EVOLUCIÓ DE T3 (SPC)

Els resultats de la tasca T3, veure histogrames a la **[Fig. 155]**, són molt similars a la tasca anterior. Els errors observats s'apropen progressivament als errors esperats fins estabilitzar-se en un valor al voltant de 0,05%, que també coincideix amb el límit inferior del model saturat. Això suposa una correspondència absoluta fins els nivells de dubte inferiors al 0,1%, i un distanciament en els intervals situats a la seva dreta.

L'evolució mostrada a la **[Fig. 156]** és pràcticament idèntica a la de la tasca anterior, estabilitzada a partir del primer miler d'exemples, amb tendència decreixent i contínua.

[Espai intencionadament en blanc per alinear el text amb les figures.]

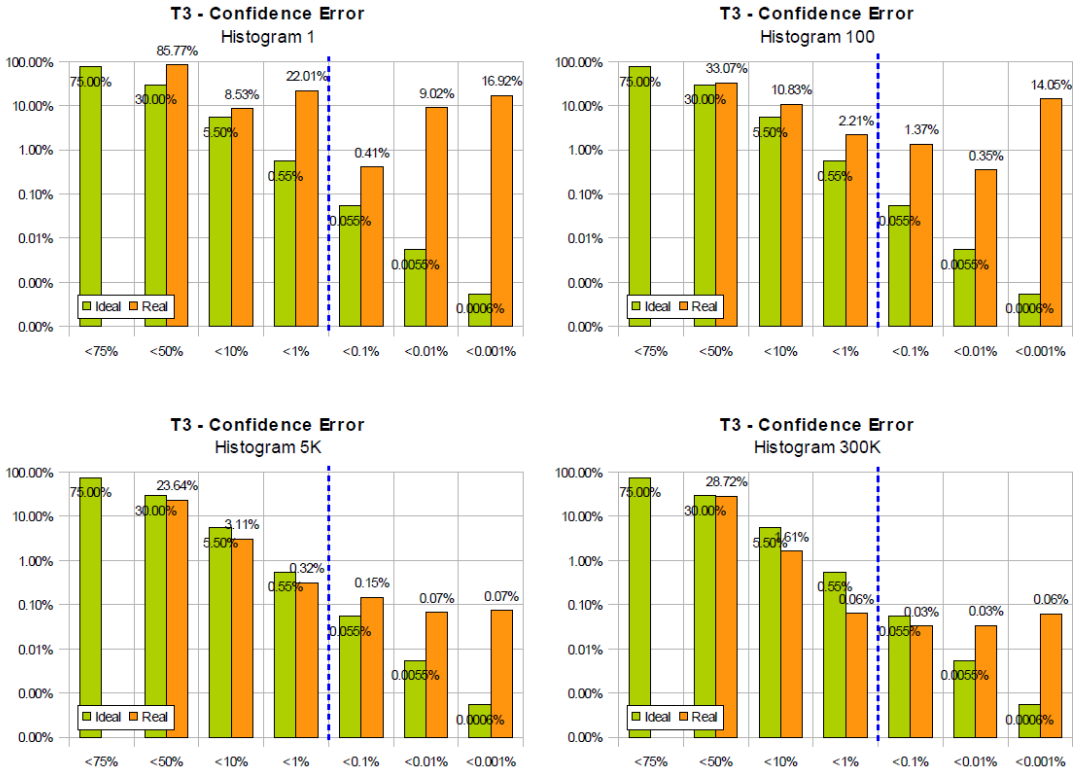

FIG. 155: Ràtios d'error esperats i observats en diferents moments d'un entrenament de T3(SPC).

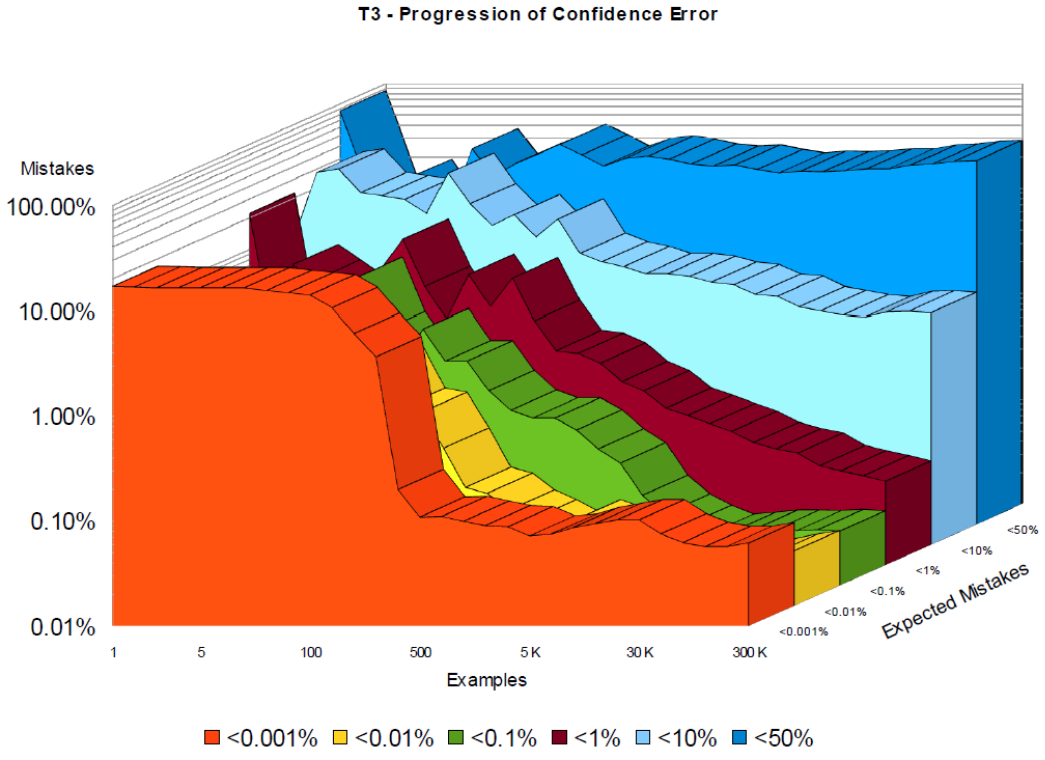

FIG. 156: Evolució del ràtio d'error observat segons el nivell de confiança a T3(SPC).

EVOLUCIÓ DE T5 (BRK)

Finalment, la tasca T3 és la que ofereix millors resultats al final de l'entrenament **[Fig. 157]**, mostrant una elevada correlació entre els errors observats i els errors esperats. Tot i que els valors absoluts poden divergir en un ordre de magnitud, la correlació és molt elevada per tots els intervals de dubte, sense mostrar cap límit inferior.

Per contra, i a diferència de les tres tasques anteriors, en l'evolució de la [**Fig. 158**] es pot veure com els errors observats assoleixen un mínim a meitat de l'entrenament, al voltant dels 500 exemples, i a partir d'aquest punt empitjoren allunyant-se dels errors esperats. És possible que el model hagi assolit un cert nivell de sobre-entrenament, no tant en relació a la seva capacitat de classificar correctament, sinó en la seva capacitat de determinar el seu propi grau de certesa.

[Espai intencionadament en blanc per alinear el text amb les figures.]

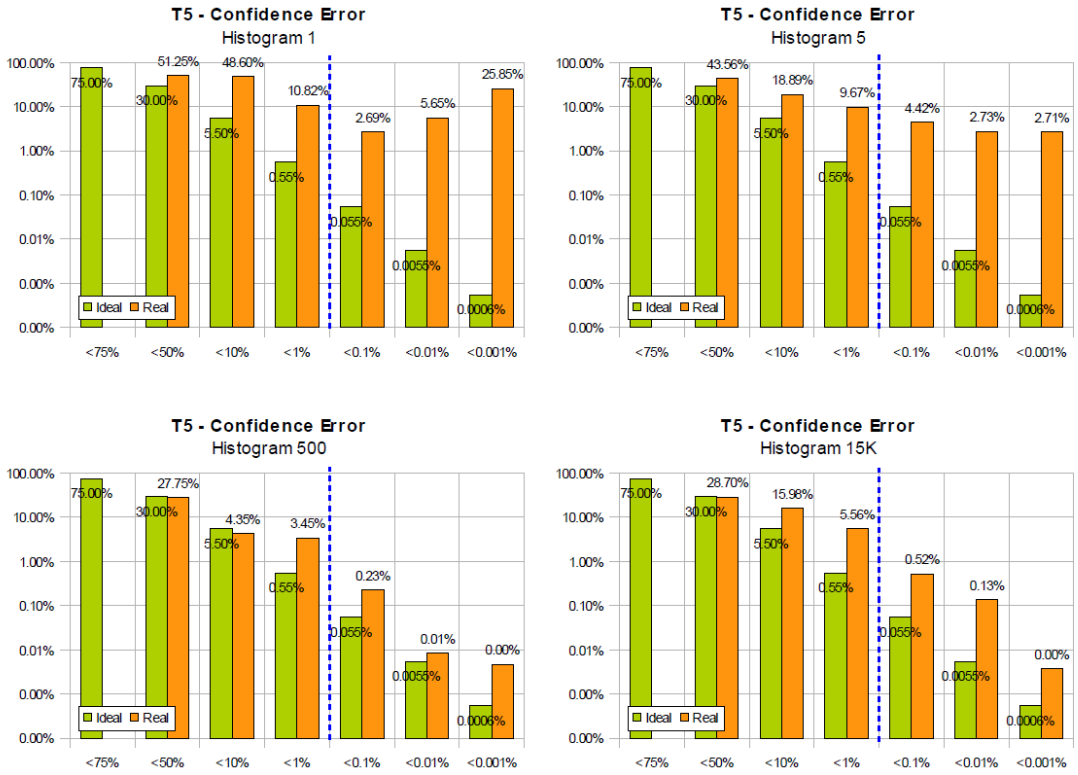

FIG. 157: Ràtios d'error esperats i observats en diferents moments d'un entrenament de T5(BRK).

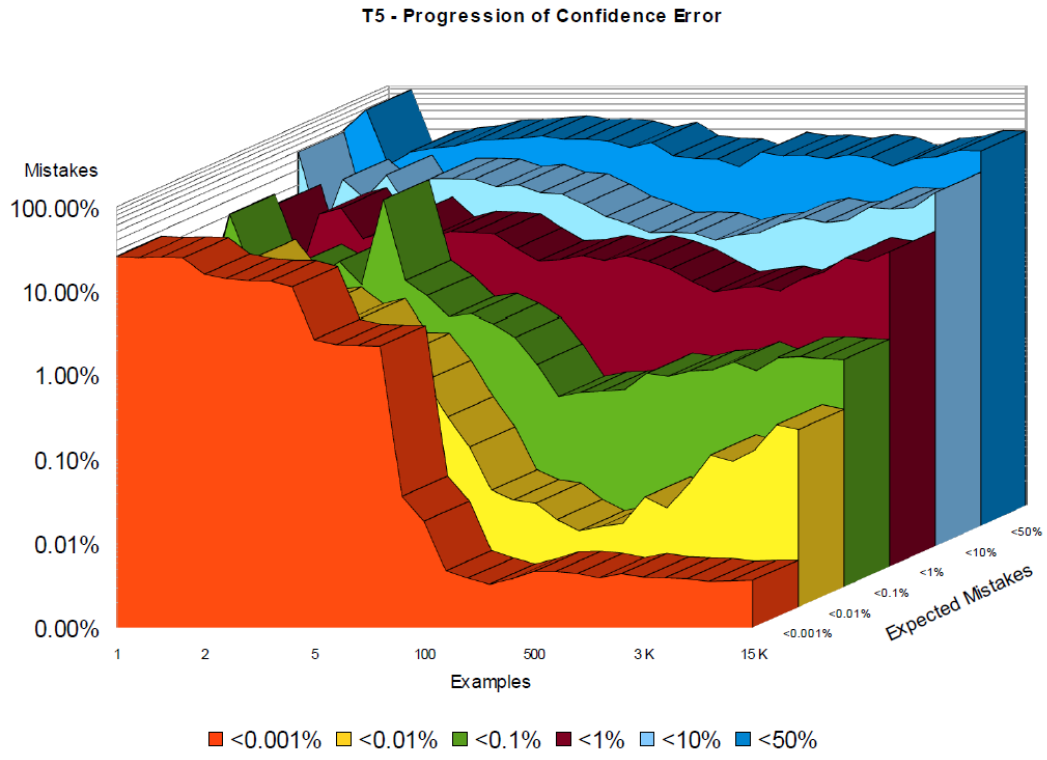

FIG. 158: Evolució del ràtio d'error observat segons el nivell de confiança a T5(BRK).

17.6 CONCLUSIONS

En aquest capítol s'ha mesurat la precisió amb que el classificador estima el nivell de certesa de les seves pròpies classificacions. Inicialment la metodologia emprada ha consistit en comparar el nivell de certesa estimat amb la proporció d'encerts o classificacions correctes. Però per facilitat la seva representació gràfica s'han transformat les dades per poder comparar el nivell de dubte estimat amb la proporció d'errors o classificacions incorrectes.

Els resultats dels experiments mostren clarament que a mesura que avança l'entrenament, no només es redueix l'error de classificació, sinó que també millora la capacitat del classificador de determinar el nivell de dubte o de certesa associat a cada una de les classificacions.

En els models que arriben a saturar i, per tant, presenten un límit inferior en l'error de classificació, la correlació entre el nivell de dubte i la probabilitat d'error és molt bona fins arribar a aquests límit, i es manté constant però per a nivells de dubte inferior.

En models no saturats, on l'entrenament encara és possible perquè l'error no ha arribat al seu mínim, la correspondència entre el nivell de dubte i la probabilitat d'error no és tan bona, però es continua mantenint una correlació positiva. És a dir, que classificacions amb un nivell de dubte inferior presenten una probabilitat d'error inferior. I, per tant, mantenen el que podríem anomenar un *criteri d'ordinalitat*, en el que els N exemples amb nivells de certesa més elevat es corresponen amb els N exemples amb més probabilitat de classificar correctament. Aquest *criteri d'ordinalitat* és l'únic requisit imprescindible per poder aplicar de forma satisfactòria qualsevol tècnica d'entrenament actiu.

Finalment, en la tasca T5, una tasca senzilla i amb un corpus sintètic prou redundant com per modelitzar la tasca de manera gairebé perfecta, la correlació entre la precisió i l'error esperat i observat és absoluta. Cosa que fa pensar que, si la tasca ho permet, no existeix límit en la precisió que pot assolir un classificador *Naïve Bayes* a l'hora d'estimar el seu nivell de certesa.

Per tant, es pot concloure que els nivells de certesa proporcionats per un classificador *Naïve Bayes* són molt exactes, es corresponen considerablement amb la precisió prevista a l'hora de classificar un exemple i, per tant, són vàlids per intentar augmentar l'eficiència de l'entrenament actiu mitjançat tècniques més elaborades.

PART IV:

L'APRENENTATGE INCREMENTAL EN PLN: CONCLUSIONS

CONCLUSIONS

Al llarg d'aquest treball de recerca aplicada s'ha defensat la importància que té per a la Lingüística Empírica (LE) i per al Processament del Llenguatge Natural (PLN) facilitar i potenciar el desenvolupament de grans corpus anotats lingüísticament. S'ha argumentat que moltes de les limitacions actuals del paradigma *batch* podrien reduir-se mitjançant tècniques d'Aprenentatge Automàtic Incremental (AAI). També s'ha proposat la utilització d'entorns d'anotació Inter-Activa que combinin algorismes d'AAI amb tècniques d'Aprenentatge Actiu com la tecnologia que permetria explotar millor els beneficis de l'entrenament incremental.

També s'ha explorat la viabilitat tècnica de l'AAI, investigant l'estat de la qüestió dels algorismes incrementals d'inducció de classificadors, de les diferents arquitectures i de les tècniques auxiliars compatibles amb aquest paradigma. La recerca feta indica que l'AAI és un camp força desenvolupat que ofereix un gran ventall d'algorismes i tècniques aplicables a tasques de PLN, i que, per tant, pot recolzar amb solidesa un aprofundiment en aquesta línia.

Finalment, un exhaustiu conjunt d'experiments realitzats sobre quatre tasques representatives de PLN han permès quantificar els beneficis de tres tècniques d'entrenament incremental: el pre-entrenament amb corpus auxiliar, l'entrenament actiu amb selecció d'exemples i l'entrenament combinant les dues tècniques anteriors.

Les conclusions d'aquest treball segueixen la mateixa estructura en què s'ha dividit el treball. La primera part recorda les limitacions del paradigma *batch* i descriu els avantatges del paradigma incremental, especialment en entorns d'anotació Inter-Activa. En la segona part es conclou que l'estat de la qüestió recolza la viabilitat tècnica de solucions basades en l'aprenentatge incremental. En la tercera part, els experiments realitzats demostren que l'eficiència de l'anotació Inter-Activa permet obtenir millors models mitjançant l'anotació de molts menys exemples. I, finalment, la darrera secció apunta algunes línies de recerca futura en les que es podria aprofundir la investigació en aquest àmbit.

I. MOTIVACIONS I AVANTATGES: JUSTIFICACIÓ CONCEPTUAL

La primera part del treball ha justificat la necessitat de superar les limitacions del paradigma d'Aprenentatge Automàtic *batch* a l'hora de desenvolupar sistemes de Processament de Llenguatge Natural. També ha presentat els avantatges que pot suposar la utilització de l'Aprenentatge Automàtic basat en algorismes incrementals.

Com s'ha explicat, la Lingüística actual és una ciència moderna que té com a objectiu crear models i teories que expliquin el funcionament del llenguatge humà. Per fer-ho necessita poder analitzar quantitativament i estadísticament el seu objecte d'estudi; d'aquí neix la importància de disposar de corpus de llenguatge natural amb anotacions lingüístiques.

El cost associat a l'anotació manual va suposar un primer *coll d'ampolla* que va poder superar-se gràcies al desenvolupament d'eines d'anotació automàtiques basades en algorismes d'Aprenentatge Automàtic que indueixen models predictius a partir d'exemples d'anotacions. Aquest fet subratlla la importància de disposar de corpus anotats lingüísticament, no només en recerca Lingüística, sinó també en el desenvolupament d'aplicacions de PLN.

Aquests mètodes empírics han anat desenvolupant-se des de finals del segle XX fins arribar a ser totalment dominants a inicis del segle XXI. Aquest creixement ha anat de la mà del desenvolupament tecnològic; de fet, la potència computacional i la capacitat d'emmagatzemament dels ordinadors actuals han facilitat progressivament el processament estadístic de les grans quantitats de text electrònic que hi ha disponibles. Però la necessitat de corpus anotats per poder entrenar aquests models ha fet reaparèixer el *coll d'ampolla* del cost d'anotació, encara que sigui en forma de revisió manual de les anotacions fetes pels sistemes automàtics.

Les limitacions encara són més greus del que pot semblar. Diferents experiments han demostrat que la disponibilitat de corpus massius, inimaginables amb la tecnologia actual, podria suposar un salt qualitatiu a l'hora d'entrenar sistemes de PLN. Per això, diferents autors subratllen la importància d'invertir en el desenvolupament de corpus molt més grans que els utilitzats actualment per la pràctica totalitat d'investigadors.

El problema és que el paradigma actual, basat en l'Aprenentatge Automàtic *batch*, presenta limitacions més profundes, més enllà del *coll d'ampolla* de l'anotació manual que no deixa de ser una limitació *simplement* econòmica. Si parlem de corpus massius cal afegir limitacions tecnològiques causades per la no escalabilitat dels algorismes d'aprenentatge. I, fins i tot, limitacions metodològiques, conseqüència dels problemes derivats de voler tractar un problema incremental, l'aprenentatge del llenguatge, com una tasca no-incremental.

L'Aprenentatge Automàtic *batch* assumeix l'existència d'un conjunt d'exemples, prou representatiu de la tasca com per poder induir un model generalitzable a la totalitat de les dades. Per tant, aplicar aquest paradigma per resoldre tasques de PLN suposa assumir que existeix un conjunt finit d'anotacions que pot capturar tota la productivitat lingüística. Només cal pensar en la distribució de les dades lingüístiques, segons la llei de Zipf, per veure que no és possible obtenir una mostra finita que sigui prou representativa, per tant no sembla fàcil que un algorisme *batch* pugui obtenir tot el coneixement necessari per resoldre la tasca en textos oberts.

Cal entendre que els models lingüístics són necessàriament aproximacions asimptòtiques, que, tot i poder millorar-se indefinidament, no poden arribar a ser completes. Per això sembla raonable utilitzar algorismes d'Aprenentatge Automàtic Incremental, que permeten desenvolupar sistemes *vius* que continuïn aprenent durant tota la seva vida, enriquint els seus models i apropant-se asimptòticament al model *ideal*.

Més enllà d'aquest canvi conceptual, la utilització d'algorismes incrementals suposa importants avantatges pràctics:

- Desapareix la divisió entre fase d'entrenament i fase d'explotació; això fa que aquestes eines puguin ser utilitzades des del primer moment, no cal ajornar la seva utilització fins a l'anotació completa d'un determinat corpus d'entrenament. A més, l'prenentatge continuat els permet adaptar el seu model a les noves dades que vagin apareixent. Cosa que indirectament resol el problema de *fora de domini*: en no existir un corpus d'entrenament diferenciat del corpus d'explotació, no existeix la necessitat de la *transportabilitat*.
- A més, la incorporació d'un nou exemple només implica actualitzar el seu model predictiu i, per tant, aquests algorismes presenten una elevada eficiència computacional. Això permet utilitzar-los en entorns interactius, tant d'anotació[1] com d'entrenament, on el coneixement implícit de les correccions de l'anotador sigui incorporat immediatament al model predictiu. La combinació d'aquesta característica amb tècniques d'Aprenentatge Actiu que minimitzin les anotacions necessàries permet desenvolupar entorns Inter-Actius que redueixin significativament les limitacions econòmiques.
- Finalment, la utilització d'algorismes incrementals obre la porta al tractament de corpus massius. Ja que l'escalabilitat d'aquest algorismes (lineals en el temps i sub-lineals en memòria) resol els problemes tècnics corresponents. A més, la possibilitat d'utilitzar els models predictius simultàniament a l'entrenament, resol els inconvenients d'haver d'esperar a la finalització de l'anotació. Això permet distribuir el cost de l'anotació al llarg de tota la seva vida i, per tant, facilita la posada en marxa de projectes d'anotació a mitjà i llarg termini.

II. ESTAT DE LA QÜESTIÓ: VIABILITAT TÈCNICA

La segona part del treball descriu l'estat actual de l'Aprenentatge Automàtic Incremental. Aquest estudi s'ha centrat principalment en els diferents algorismes capaços d'induir incrementalment models predictius. A més a més, ha analitzat arquitectures complexes basades en la combinació d'aquests algorismes i diferents tècniques auxiliars adaptades al paradigma incremental.

La finalitat d'aquesta part ha estat revisar críticament les diferents famílies d'algorismes d'aprenentatge amb els objectius de, primer, conèixer quins algorismes incrementals hi ha disponibles, i, segon, determinar quins són els idonis a l'hora d'utilitzar-los en sistemes de PLN incremental. Com s'ha explicat al llarg del treball, a més de per l'estricta incrementalitat de l'algorisme, el grau d'idoneïtat ve determinat tant per la seva capacitat d'extreure informació de representacions lingüísticament riques, com pel grau d'escalabilitat dels recursos utilitzats, computacionals i de memòria.

[1] També coneguts com eines CATA (*Computer Aided Text Annotation*)

La primera conclusió ha estat comprovar que la incrementalitat no està restringida a unes poques famílies d'algorismes. Tot i que hi ha algunes famílies intrínsecament incrementals, en la majoria de famílies s'han proposat variants que, amb més o menys eficiència, permeten entrenar els models de manera incremental.

Per un costat tenim els mètodes simbòlics basats en la inducció incremental d'arbres (ID4 i ID5R) i regles de decisió (Crystal, PRISM, PDL^2, ILA i RDR). Els seus avantatges són la gran capacitat de generalització i, sobretot, la interpretabilitat dels seus models, una característica important en la recerca. El principal inconvenient és la poca eficiència computacional de les reestructuracions dels arbres de decisió durant la seva actualització.

Per un altre costat hi ha els algorismes de base estadística, entre els quals destaca el *Naïve Bayes* pels bons resultats que dóna malgrat la seva simplicitat. Aquest algorisme pot tractar representacions binàries, simbòliques o numèriques, i ha estat utilitzat satisfactòriament en moltes tasques de PLN. La seva escalabilitat és molt bona i pot tractar representacions d'elevada dimensionalitat. També destaquen els classificadors lineals, representats pel *Winnow* i les seves variants, per la seva incrementalitat intrínseca. La seva escalabilitat és immillorable, models de mida constant, i també tenen la capacitat de treballar en tasques d'elevada dimensionalitat. Una limitació secundària és el requeriment d'haver de binaritzar els trets i que per resoldre tasques multiclasse cal combinar diversos classificadors; limitacions solubles amb diferents tècniques auxiliars.

S'ha deixat pel final la família dels algorismes basats en memòria, representats pels *K-Nearest Neighbour* (IB-1, IB-k, KStar), per l'ambivalència de la seva idoneïtat. No només és intrínsecament incremental, sinó que alguns autors consideren que és el paradigma perfecte per a tasques de PLN; a més, és el model que més pot beneficiar-se de l'entrenament amb corpus massius. Tot i que el cost d'entrenament és mínim, els recursos de memòria necessaris per emmagatzemar els exemples i, especialment, el cost computacional lineal de la classificació qüestionen la seva aplicació en tasques interactives. Caldria investigar si alguna variant proposada per limitar la memòria utilitzada (IB-2) o alguna tècnica per optimitzar el cost computacional (IGTree) pot resoldre satisfactòriament aquestes limitacions.

Per tant, tot sembla indicar que els algorismes *Naïve Bayes* i les variants del Winnow encaixen perfectament en entorns d'aprenentatge incremental. Hi ha indicis que també ho podrien fer els basats en memòria si s'utilitzen versions que minimitzin els recursos necessaris. I també sembla raonable utilitzar algorismes d'inducció de regles si es busca la interpretabilitat dels models induïts. Si s'hagués de descartar *a priori* alguns algorismes, serien possiblement la inducció d'arbres de decisió i, sobretot, els models connexionistes que poden tenir problemes per convergir en ser entrenats incrementalment.

A més, el fet que diferents algorismes estiguin més adaptats a certs tipus de trets, suggereix que la millor opció és utilitzar una combinació de classificadors. Tots els sistemes de votació poden ser aplicats incrementalment, i també la majoria d'arquitectures complexes: el *bagging* per obtenir diversitat a partir del mostreig aleatori, l'*stacking* com a sistema de

metaaprenentatge, o l'arbitratge per monitoritzar el rendiment de diferents classificadors entrenats en paral·lel. Finalment, cal afegir aquelles tècniques auxiliars que també poden ajudar a millorar els resultats: transformació d'exemples, expansió de trets, discretització de trets numèrics, selecció de trets i optimització de paràmetres; totes elles aplicables incrementalment.

Per tant, es pot concloure que la utilització exclusiva de tècniques incrementals d'Aprenentatge Automàtic pràcticament no restringeix la capacitat de tria ni entre les famílies d'algorismes ni entre les arquitectures i tècniques auxiliars. Això recolza la viabilitat tècnica d'aplicar algorismes d'aprenentatge incremental a l'hora de resoldre tasques de PLN.

III. PROVES EXPERIMENTALS: MESURA DE L'EFICIÈNCIA

En la tercera part del treball s'han volgut validar les hipòtesis plantejades inicialment, i per fer-ho ha calgut realitzar una sèrie d'experiments que permetin quantificar els beneficis de diferents tècniques d'entrenament incremental.

A partir de dos corpus bilingües, l'*AnCora* i el *De-News*, s'han preparat cinc tasques de PLN de les quals finalment se n'han seleccionat quatre per a complir els requeriments buscats: que fossin **tasques solubles** i **no saturades**. Les tasques seleccionades han estat la T1 (POS) d'anotació morfosintàctica, la T2 (ENT) de detecció d'entitats anomenades (ENT), la T3 (DOT) de desambiguació de punts i la T5 (BRK) de segmentació d'oracions. Les característiques d'aquestes tasques (mida de l'etiquetari, quantitat d'exemples, grau de saturació o error assolit) presenten una gran varietat i, per tant, són representatives d'un ampli ventall de tasques de PLN.

Durant la preselecció de tasques s'ha relacionat el concepte de saturació amb la proporció entre la quantitat d'exemples disponibles i la mida de l'univers de representació de la tasca. S'ha proposat la creació d'un ***índex de densitat*** d'un corpus que pugui utilitzar-se com a indicador del risc de saturació. Com s'ha dit en el capítol corresponent, tot i que seria necessària una recerca més profunda per a poder confirmar la seva validesa. S'ha observat que les tasques que presentaven corbes d'aprenentatge amb indicis de saturació es corresponien precisament amb les tasques amb *índexs de densitat* més elevats.

Durant els experiments realitzats s'han avaluat tres **tècniques d'entrenament incremental**: el pre-entrenament amb corpus auxiliar, la selecció d'exemples amb aprenentatge actiu i la combinació simultània de les dues tècniques anteriors.

Els resultats confirmen que el **pre-entrenament** de models utilitzant corpus auxiliars permet obtenir millors models i reduir els errors en l'etapa inicial transitòria. També demostren que mitjançant la tria acurada del pes relatiu de cada corpus és possible assolir aquest benefici sense perjudicar el model final. Concretament, en el cas que el corpus auxiliar i el corpus principal presentin un elevat grau de coherència es recomana donar un pes similar a tots dos corpus; però en el cas que presentin diferències significatives és

recomanable infraponderar (1/100, 1/1000) el corpus auxiliar per minimitzar la interferència en el model final.

Els experiments al voltant de l'**entrenament actiu** són els que han proporcionat uns resultats més concloents. Les dades indiquen sense cap dubte que la inducció de models mitjançant l'entrenament actiu incremental permet millorar els resultats de l'entrenament incremental estàndard, especialment en relació a la quantitat d'exemples utilitzats. Concretament, en les diferents tasques l'entrenament actiu ha permès obtenir models més precisos, amb errors entre un 5% i un 50% inferiors, utilitzant una quantitat molt inferior d'exemples, entre 5 i 100 vegades menys exemples. Tot i això, cal tenir en compte que els beneficis presenten variacions molt grans segons el cas, i l'anàlisi de les quatre tasques estudiades suggereix que el factor determinant és el *grau de redundància* que presenti el corpus.

També s'ha pogut comprovar que mitjançant l'**entrenament combinat** és possible aplicar aquestes dues tècniques i obtenir simultàniament els seus beneficis individuals, de manera que es redueix l'error en l'etapa inicial i, al mateix temps, s'accelera l'entrenament obtenint errors més baixos al llarg de tot l'entrenament. De totes maneres, tot i que els resultats són molt millors que els de l'entrenament estàndard, en general no supera els resultats de l'entrenament actiu pur: assoleix errors molt similars però requerint una major quantitat d'exemples.

Dels experiments realitzats també es pot concloure que la variant utilitzada per alimentar el model inicial per a l'entrenament actiu, mostreig aleatori o endarreriment, o els valors dels paràmetres corresponents, no suposa grans diferències en els resultats finals. Per contra, queda demostrada una elevada sensibilitat respecte el valor del paràmetre del llindar de certesa. En la majoria de casos valors de certesa al voltant del 99%, que representa un nivell de "dubte" o error de l'1%, donen bons resultats i superen l'entrenament estàndard. En determinades tasques especialment solubles i amb corpus amb una baixa taxa d'error es poden reduir molt els errors de classificació augmentant els valors a 99,9% o 99,99%. Però cal tenir en compte que si aquest llindar supera l'error assolible a la tasca, l'error final i, sobretot, l'eficiència final pot empitjorar significativament.

L'anàlisi de les corbes d'**ASR** (*Annotated Sampling Ratio*) confirmen que l'entrenament actiu permet reduir progressivament la proporció d'exemples que cal anotar d'un corpus determinat a mesura que el model millora i el classificador guanya experiència. Però aquesta reducció no és indefinida i, per exemple, sembla que quan el model comença a saturar-se i no li és possible reduir l'error, en un primer moment, l'ASR s'estabilitza i, posteriorment, comença a augmentar. La principal conclusió és que la selecció del valor del llindar de confiança és crític per obtenir un model eficient, i que la selecció d'un valor excessivament baix o excessivament alt pot perjudicar molt l'eficiència assolida.

Per un altre costat, en analitzar les distribucions dels **nivells de certesa** assignats pel model durant les classificacions, s'ha pogut confirmar una idea que semblava raonable: a mesura que un model és entrenat i redueix l'error de classificació, augmenta la certesa amb la que

classifica els exemples. I és aquest desplaçament a l'alça dels nivells de certesa el que permet augmentar l'eficiència a mesura que avança l'entrenament i més exemples superen el llindar de certesa.

A més, en mesurar la precisió d'aquestes estimacions s'ha observat que a mesura que avança l'entrenament també millora la capacitat del classificador per determinar amb exactitud el nivell de dubte o de certesa associat a cada una de les classificacions. Per tant, almenys per un classificador *Naïve Bayes*, aquestes estimacions són vàlides per a desenvolupar tècniques de selecció d'exemples més elaborades.

IV. RECERCA FUTURA: LÍNIES DE MILLORA

L'objectiu global de la tesi era demostrar els beneficis que poden proporcionar els algorismes incrementals d'Aprenentatge Automàtic al camp del PLN; especialment mitjançant la seva integració en entorns d'anotació interactiva basada en l'Aprenentatge Actiu. Tot i que aquest objectiu ha estat assolit, el treball realitzat no ha fet més que reobrir una porta a unes tècniques sobre les quals queda molt per aprofundir. De fet, pel camí han aparegut resultats inesperats i algunes qüestions sobre les quals seria molt interessant fer recerques específiques, ja que només s'ha pogut suggerir-ne pinzellades.

La principal línia de recerca que suggereixen els resultats és el desenvolupament d'algorismes d'entrenament actiu que resolguin la limitació més important detectada a les tècniques presentades: l'excessiva sensibilitat de l'eficiència assolida al valor triat com a llindar de certesa. Recordem que independentment de la variant d'entrenament utilitzada l'error i l'eficiència assolida (*ASR*) depenien bàsicament del llindar de certesa utilitzat per seleccionar els exemples per anotar.

El fet d'utilitzar un valor constant, quan s'ha demostrat que la precisió del classificador varia al llarg de l'entrenament, fa que aquest valor pugui ser: 1) excessiu quan el model encara està estabilitzant-se, 2) insuficient quan el model ha madurat i, 3) de nou excessiu si entra en una fase de saturació. Aquestes ineficiències impedeixen minimitzar l'error assolit i l'eficiència final. Per això se suggereix investigar algorismes d'entrenament actiu basats en **llindars de certesa dinàmics**, llindars variables al llarg del temps: a) inicialment baixos, de forma que se seleccionin la majoria d'exemples fins a estabilitzar el model, b) progressivament més alts, per pujar el llistó del classificador centrant-se en els casos més difícils, i c) amb un mecanisme de detecció de saturació que impedeixi que el llindar de certesa superi en excés la precisió màxima de la tasca, ja que això augmentaria innecessàriament l'*ASR* sense reduir l'error.

Una altre línia de recerca interessant, relacionada amb l'anterior, gira al voltant de la utilització de l'*ASR* com a **indicador de saturació del model**. Alguns dels experiments realitzats suggereixen que l'evolució habitual del classificador és anar disminuint l'*ASR* a mesura que avança l'entrenament i millora la precisió del classificador. Però quan el model entra en la fase de saturació, on la seva precisió disminueix més lentament fins estabilitzar-se, l'*ASR* comença a augmentar. Així doncs, per un costat caldria fer experiments específics

per validar aquesta hipòtesi i, en cas de ser certa, desenvolupar alguna tècnica que permetés detectar o quantificar el grau de saturació. Aquesta eina no només permetria determinar quan un model està madur i ha assolit la seva màxima precisió, sinó que permetria implementar el mecanisme c) de l'algorisme d'entrenament dinàmic suggerit al paràgraf anterior.

Des d'un punt de vista de recerca més bàsica, i potser no tan aplicada, hi ha dos aspectes que pot valer la pena investigar: l'*índex de densitat* d'un corpus i el *grau de similitud* entre corpus.

Per un costat caldria aprofundir en la validesa i utilitat de l'***índex de densitat*** proposat en aquest treball. Recordem que es tracta d'un indicador del grau de redundància, i per tant del risc de saturació del model, d'un corpus determinat a partir de la proporció entre la quantitat d'exemples i la mida de l'univers de representació. La validació d'aquest indicador permetria determinar *a priori* el grau de redundància d'un corpus i, per tant, fins a quin punt es pot beneficiar de la utilització d'un entrenament actiu en relació a un entrenament incremental estàndard.

Per un altre costat caldria trobar i validar alguna mètrica que permetés determinar el ***grau de similitud*** entre dos corpus. Aquesta mesura no hauria de basar-se en la similitud superficial dels corpus, sinó en el grau de coherència (o contradicció) entre els exemples descrits en una representació comuna i les etiquetes assignades. La característica que hauria d'intentar capturar aquesta mètrica és fins a quin punt el corpus A pot utilitzar-se amb èxit per entrenar un model que serà validat amb el corpus B. La possibilitat de determinar *a priori* aquesta informació permetria aprofitar tot el potencial del pre-entrenament d'un model mitjançant la utilització d'un corpus auxiliar. Recordem que el *grau de similitud* determina si cal infraponderar o sobreponderar el corpus auxiliar en relació al corpus principal.

També queda obert el debat de fins a quin punt són extrapolables els resultats dels experiments a altres algorismes d'Aprenentatge Incremental diferents al *Naïve Bayes*. En principi haurien de ser aplicables a qualsevol algorisme capaç d'induir incrementalment un model classificador i, molt important, capaç de proporcionar un grau de certesa per cada classificació realitzada. La recerca feta a la segona part d'aquest treball suggereix que tant els classificadors lineals (*Winnow* o *SNoW*) com els algorismes basats en memòria (*K-NN*, *IB-1*, *K-Star*, ...) són bons candidats. També és cert que la facilitat i precisió amb què el *Naïve Bayes* pot determinar la probabilitat de l'etiqueta assignada pot no ser assolible per aquests altres algorismes; caldria obtenir resultats experimentals per poder confirmar-ho.

Finalment, i des d'un punt de vista més tècnic, seria interessant desenvolupar una prototip real d'anotador Inter-Actiu. Tot i que els resultats obtinguts per la "simulació computacional" de l'anotador humà són fiables a l'hora de determinar l'eficiència a la que poden arribar, és difícil conèixer exactament els problemes d'usabilitat que podrien aparèixer en una aplicació real d'aquesta tecnologia.

PART V:

L'APRENENTATGE INCREMENTAL EN PLN: ANNEXOS

ANNEX A: ETIQUETARIS

1. ETIQUETARI MORFOSINTÀCTIC

Enumeració de les 122 etiquetes[1] morfosintàctiques utilitzades a les tasques T1 i T2. Es mostren agrupades en 12 categories majors juntament amb els trets corresponents inclosos a l'*AnCora* i la freqüència absoluta d'aparició als corpus català i espanyol.

ADJECTIVE		gen=m|f|c num=s|p|c posfunction=participle
-	A-	
-	AC	postype=common
-	AM	postype=main
3.542	AO	postype=ordinal
62.277	AQ	postype=qualificative
CONJUNCTION		
-	C-	
29.251	CC	postype=coordinating
21.402	CS	postype=subordinating
DETERMINER		gen=m|f|c num=s|p|c person=1|2|3 possessornum=s|p
-	D-	
100.861	DA	postype=article
6.270	DD	postype=demonstrative
13	DE	postype=exclamative
26.341	DI	postype=indefinite
118	DT	postype=interrogative
4.681	DN	postype=numeral
7.799	DP	postype=possessive
410	DPp	postype=possessive possessornum=p
324	DPs	postype=possessive possessornum=s
55	DR	postype=relative
PUNCTUATION		
-	F-	
-	Fa	punct=apostrophe
1.288	Fl	punct=colon
195	Ft	punct=etc
3.823	Fy	punct=hyphen
3.906	Fm	punct=mathsign
33.990	Fp	punct=period
13.705	Fq	punct=quotation
729	Fs	punct=semicolon

[1] De les 122 etiquetes disponibles només 88 tipus apareixen al corpus de l'*AnCora*.

3.823	Fh	punct=slash
-	Fr	punct=revslash
56.721	FC	punct=comma
-	FCc	punct=comma punctenclose=close
-	FCo	punct=comma punctenclose=open
-	FK	punct=quotation
-	FKc	punct=quotation punctenclose=close
-	FKo	punct=quotation punctenclose=open
-	FB-	punct=bracket
3.528	FBc	punct=bracket punctenclose=close
3.530	FBo	punct=bracket punctenclose=open
-	FBs-	punct=sqbracket
-	FBsc	punct=sqbracket punctenclose=close
-	FBso	punct=sqbracket punctenclose=open
-	FBk-	punct=cubracket
-	FBkc	punct=cubracket punctenclose=close
-	FBko	punct=cubracket punctenclose=open
89	FEc	punct=exclamationmark punctenclose=close
67	FEo	punct=exclamationmark punctenclose=open
590	FQc	punct=questionmark punctenclose=close
387	FQo	punct=questionmark punctenclose=open

INTERJECTION

109	I-	

NOUN gen=m|f|c num=s|p|c

-	N-	
181.483	NC	postype=common
58.878	NP	postype=proper

PRONOUN gen=m|f|c num=s|p|c person=1|2|3 possessornum=s|p polite=yes case=oblique|nominative|accusative|dative

31.052	P-	
1.295	PD	postype=demonstrative
-	PE	postype=exclamative
3.383	PI	postype=indefinite
705	PT	postype=interrogative
871	PN	postype=numeral
4.803	PS	postype=personal
1.992	PSa	postype=personal case=accusative
1.517	PSd	postype=personal case=dative
233	PSn	postype=personal case=nominative
143	PSo	postype=personal case=oblique
83	PSy	postype=personal polite=yes
47	PP	postype=possessive
32	PPp	postype=possessive possessornum=p
14	PPs	postype=possessive possessornum=s
18.436	PR	postype=relative

ADVERB

-	R-	
28.113	RG	postype=general
6.454	RN	postype=negative

PREPOSITION		gen=m num=s|p
-	S-	
133.560	SP	postype=preposition
23.292	SPc	postype=preposition contracted=yes
VERB		gen=m|f num=s|p|c person=1|2|3 posfunction=participle
-	Vm--	postype=main
371	VmZM	postype=main mood=imperative
26.943	VmZI	postype=main mood=infinitive
2.480	VmZG	postype=main mood=gerund
11.769	VmZP	postype=main mood=participle
26.982	VmIP	postype=main mood=indicative tense=present
4.351	VmII	postype=main mood=indicative tense=imperfect
5.663	VmIF	postype=main mood=indicative tense=future
11.349	VmIS	postype=main mood=indicative tense=past
1.092	VmIC	postype=main mood=indicative tense=conditional
3.554	VmSP	postype=main mood=subjunctive tense=present
1.011	VmSI	postype=main mood=subjunctive tense=imperfect
-	VmSF	postype=main mood=subjunctive tense=future
-	VmSS	postype=main mood=subjunctive tense=past
-	VmSC	postype=main mood=subjunctive tense=conditional
5	VxZM	postype=auxiliary mood=imperative
488	VxZI	postype=auxiliary mood=infinitive
9	VxZG	postype=auxiliary mood=gerund
52	VxZP	postype=auxiliary mood=participle
16.699	VxIP	postype=auxiliary mood=indicative tense=present
1.007	VxII	postype=auxiliary mood=indicative tense=imperfect
194	VxIF	postype=auxiliary mood=indicative tense=future
62	VxIS	postype=auxiliary mood=indicative tense=past
251	VxIC	postype=auxiliary mood=indicative tense=conditional
292	VxSP	postype=auxiliary mood=subjunctive tense=present
160	VxSI	postype=auxiliary mood=subjunctive tense=imperfect
-	VxSF	postype=auxiliary mood=subjunctive tense=future
-	VxSS	postype=auxiliary mood=subjunctive tense=past
-	VxSC	postype=auxiliary mood=subjunctive tense=conditional
10	VsZM	postype=semiauxiliary mood=imperative
1.463	VsZI	postype=semiauxiliary mood=infinitive
110	VsZG	postype=semiauxiliary mood=gerund
873	VsZP	postype=semiauxiliary mood=participle
5.431	VsIP	postype=semiauxiliary mood=indicative tense=present
676	VsII	postype=semiauxiliary mood=indicative tense=imperfect
-	VsIF	postype=semiauxiliary mood=indicative tense=future
947	VsIS	postype=semiauxiliary mood=indicative tense=past
190	VsIC	postype=semiauxiliary mood=indicative tense=conditional
375	VsSP	postype=semiauxiliary mood=subjunctive tense=present
103	VsSI	postype=semiauxiliary mood=subjunctive tense=imperfect
641	VsIF	postype=semiauxiliary mood=subjunctive tense=future
-	VsIS	postype=semiauxiliary mood=subjunctive tense=past
-	VsIC	postype=semiauxiliary mood=subjunctive tense=conditional
DATE		
6.023	W-	

NUMBER

```
8.033     Z-
1.483     ZC    postype=currency
1.758     ZP    postype=percentatge
```

2. ETIQUETARI TIPOGRÀFIC

La informació ortotipogràfica de cada token es codifica mitjançant una etiqueta formada per una seqüència de 2 o 3 caràcters segons el token, de manera que els dos primers són obligatoris i indican el tipus i nombre de caràcters que el formen i el tercer, que pot aparèixer o no, indica la presencia de determinats caràcters finals.

Concretament el format de les etiquetes tipogràfiques és el següent:

```
Etiqueta Tipogràfica:
<Type><Length>[<Final>]
```

I els diferents elements poden prendre els següents valors

<Type>	**Meaning**	**Characters**
D	Digits	[0-9]+
N	Numbers	[0-9+-%.,]+
L	LowerCase	[a-z]+
U	UpperCase	[A-Z]+
T	TitleCase	[A-Z][a-z]+
M	MixedCase	[a-zA-Z]+
X	Other	(catch all)

<Length>	**Meaning**	**Pattern**
1	1 character	#
2	2 characters	##
3	3 characters	###
4	4 characters	####
5	5 characters	#####
6	6 characters	######
7	7 characters	#######
8	8 characters	########
9	9 or more	#########...

<Final>	**Meaning**	**Character**
d	Final Dot	[.]
a	Final Apostrophe	[']
p	Final Percent	[%]

A continuació es mostren alguns exemples de tokens arbitraris i la seva etiqueta ortotipogràfica:

Token	Label
"a"	[L1]
"A"	[U1]
"3"	[D1]
"+"	[X1]
"."	[X1]
"aaaa"	[L4]
"AAAA"	[U4]
"Aaaa"	[T4]
"aAaA"	[M4]
"aa'"	[L2']
"000"	[D3]
"000%"	[D3%]
"+0.00"	[N5]
"+0.00%"	[N5%]

3. PUNTUACIÓ CANÒNICA

Forma canònica utilitzada pels diferents signes de puntació:

Ch	Lemma
.	[PERIOD]
,	[COMMA]
:	[COLON]
;	[SEMICOLON]
–	[MINUS]
-	[HYPHEN]
–	[DASH]
_	[UNDERSCORE]
(	[PARENTHESIS_OPEN]
)	[PARENTHESIS_CLOSE]
[	[SQUARE_OPEN]
]	[SQUARE_CLOSE]
{	[CURLY_OPEN]
}	[CURLY_CLOSE]
?	[QUESTION_OPEN]
¿	[QUESTION_CLOSE]

Ch	Lemma
¡	[EXCLAMATION_OPEN]
!	[EXCLAMATION_CLOSE]
'	[APOSTROPHE]
`	[APOSTROPHE]
´	[APOSTROPHE]
\	[QUOTATION]
«	[GUILLEMET_OPEN]
»	[GUILLEMET_CLOSE]
|	[VERTICALBAR]
/	[SLASH]
⁄	[SLASH]
\	[BACKSLASH]
%	[PERCENT]
SPC	[SPACE]
	[EMPTY]

ANNEX B: RESULTATS EXPERIMENTALS

Corpus:

T1:	Tasca 1
T2:	Tasca 2
T3:	Tasca 3
T5:	Tasca 5

Tècnica:

STD:	Standard
PRE:	Pre-entrenament
ACT:	Entrenament Actiu
COMB:	Combinat

Paràmetres:

CONF:	`Confidence`, llindar de certesa.
RAND:	`Random`, mostratge aleatori.
FIRST:	`First`, endarreriment de l'entrenament actiu.

Notes:

REF:	Referència
GOOD:	Bons resultats
BEST:	Millors resultats
DISC:	*Discarded*, dades no utilitzades.
OVRFL:	*Overflow* durant l'entrenament.
***:**	Seleccionat

1. EXPERIMENTS TASCA T1 (POS)

	Minim	Maxim	Mean	ASR	
T1-STD CAT	4.20%	4.52%	4.38%	100%	REF*
T1-STD ESP	4.46%	4.86%	4.71%	100%	
	Minim	**Maxim**	**Mean**	**ASR**	
T1-PRE x1	3.93%	4.44%	4.21%	100%	GOOD*
T1-PRE x10	4.11%	4.52%	4.34%	100%	DISC
T1-PRE x10	4.23%	4.55%	4.39%	100%	GOOD*
T1-PRE x100	4.49%	4.77%	4.65%	100%	
T1-PRE x1,000	4.82%	5.11%	4.96%	100%	
T1-PRE x10,000	5.43%	5.78%	5.56%	100%	
	Minim	**Maxim**	**Mean**	**ASR**	
T1-ACT Conf <0.99	4.57%	5.26%	4.89%	10.14%	DISC
T1-ACT Conf <0.999	4.29%	4.55%	4.44%	14.79%	DISC
T1-ACT Conf <0.9999	3.94%	4.37%	4.19%	20.07%	DISC*

	Minim	Maxim	Mean	ASR	
T1-ACT Rand 1% + Conf <0.999	4.26	4.64%	4.44%	15.53%	
T1-ACT Rand 1% + Conf <0.9999	4.07	4.33%	4.20%	20.72%	GOOD*
T1-ACT Rand 5% + Conf <0.6	6.81%	7.40%	7.02%	6.78%	
T1-ACT Rand 5% + Conf <0.9	5.28%	5.78%	5.61%	10.05%	
T1-ACT Rand 5% + Conf <0.95	5.12%	5.46%	5.28%	11.33%	
T1-ACT Rand 5% + Conf <0.99	4.47%	5.11%	4.83%	14.12%	
T1-ACT Rand 5% + Conf <0.999	4.29%	4.66%	4.46%	18.56%	
T1-ACT Rand 5% + Conf <0.9999	3.99%	4.50%	4.19%	23.46%	GOOD*

	Minim	Maxim	Mean	ASR	
T1-ACT First 100 + Conf <0.95	5.22%	5.67%	5.43%	7.10%	DISC
T1-ACT First 100 + Conf <0.99	4.79%	5.19%	4.94%	10.13%	
T1-ACT First 100 + Conf <0.999	4.23%	4.64%	4.42%	14.81%	
T1-ACT First 100 + Conf <0.9999	4.06%	4.32%	4.17%	20.06%	GOOD*
T1-ACT First 1,000 + Conf <0.99	4.62%	5.01%	4.89%	10.19%	
T1-ACT First 1,000 + Conf <0.999	4.25%	4.58%	4.40%	14.93%	
T1-ACT First 1,000 + Conf <0.9999	3.89%	4.42%	4.20%	20.08%	GOOD*

	Minim	Maxim	Mean	ASR	
T1-COM x1 + Conf <0.9999	4.09%	4.32%	4.19%	19.93%	GOOD*
T1-COM x10 + Conf <0.9999	4.14%	4.39%	4.28%	18.24%	
T1-COM x1 + Rand 1% + Conf <0.9999	3.98%	4.33%	4.18%	20.53%	GOOD*
T1-COM x10 + Rand 1% + Conf <0.9999	4.10%	4.47%	4.33%	18.99%	
T1-COM x1 + Rand 5% + Conf <0.9999	3.98%	4.39%	4.15%	23.41%	BEST*
T1-COM x10 + Rand 5% + Conf <0.9999	4.16%	4.53%	4.33%	21.97%	
T1-COM x1 + First 100 + Conf <0.9999	4.03%	4.36%	4.21%	19.91%	GOOD*
T1-COM x10 + First 100 + Conf <0.9999	4.16%	4.43%	4.29%	18.17%	
T1-COM x1 + First 1,000 + Conf <0.9999	4.07%	4.36%	4.19%	20.04%	GOOD*
T1-COM x10 + First 1,000 + Conf <0.9999	4.20%	4.43%	4.30%	18.42%	

2. EXPERIMENTS TASCA T2 (ENT)

	Minim	Maxim	Mean	ASR	
T2-STD CAT	3.36%	3.65%	3.53%	100%	REF*
T2-STD ESP	2.17%	2.38%	2.28%	100%	DISC*

	Minim	Maxim	Mean	ASR	
T2-PRE x1	3.39%	3.70%	3.50%	100%	
T2-PRE x10	3.36%	3.72%	3.54%	100%	GOOD*
T2-PRE x100	3.43%	3.61%	3.53%	100%	
T2-PRE x1,000	3.39%	3.66%	3.53%	100%	
T2-PRE x10,000	3.48%	3.73%	3.60%	32.7%	OVRFL

	Minim	Maxim	Mean	ASR	
T2-ACT Rand 1% + Conf <0.99	2.69%	2.88%	2.79%	21.18%	GOOD*
T2-ACT Rand 1% + Conf <0.999	2.89%	3.16%	3.05%	45.07%	
T2-ACT Rand 1% + Conf <0.9999	3.15%	3.37%	3.26%	71.83%	
T2-ACT Rand 5% + Conf <0.9	2.89%	3.12%	3.00%	13.01%	
T2-ACT Rand 5% + Conf <0.99	2.72%	3.00%	2.86%	24.41%	GOOD*
T2-ACT Rand 5% + Conf <0.999	2.90%	3.18%	3.08%	47.29%	
T2-ACT Rand 5% + Conf <0.9999	3.21%	3.43%	3.27%	73.16%	

	Minim	Maxim	Mean	ASR	
T2-ACT First 100 + Conf <0.90	2.97%	3.23%	3.15%	7.70%	
T2-ACT First 100 + Conf <0.99	2.71%	2.85%	2.79%	20.48%	GOOD*
T2-ACT First 100 + Conf <0.999	2.91%	3.16%	3.02%	44.41%	
T2-ACT First 100 + Conf <0.9999	3.15%	3.33%	3.24%	71.55%	
T2-ACT First 1,000 + Conf <0.99	2.65%	2.93%	2.79%	20.10%	
BEST**					
T2-ACT First 1,000 + Conf <0.999	2.92%	3.11%	3.02%	43.96%	
T2-ACT First 1,000 + Conf <0.9999	3.15%	3.35%	3.25%	71.92%	

	Minim	Maxim	Mean	ASR	
T2-COM x1 + Conf <0.99	3.24%	3.49%	3.35%	34.28%	
T2-COM x10 + Conf <0.99	2.76%	3.06%	2.91%	36.20%	GOOD
T2-COM x1 + Rand 1% + Conf <0.99	3.27%	3.53%	3.35%	35.72%	
T2-COM x10 + Rand 1% + Conf <0.99	2.84%	3.05%	2.92%	37.43%	GOOD
T2-COM x1 + Rand 5% + Conf <0.99	3.24%	3.45%	3.35%	41.63%	
T2-COM x10 + Rand 5% + Conf <0.99	2.88%	3.05%	2.97%	43.33%	
T2-COM x1 + First 100 + Conf <0.99	3.24%	3.48%	3.35%	34.26%	
T2-COM x10 + First 100 + Conf <0.99	2.85%	3.02%	2.91%	36.08%	GOOD
T2-COM x1 + First 1,000 + Conf <0.99	3.22%	3.52%	3.35%	34.53%	
T2-COM x10 + First 1,000 + Conf <0.99	2.83%	3.03%	2.90%	36.22%	BEST*
T2-COM x100 + First 1,000 + Conf <0.99	2.81%	3.15%	3.01%	38.70%	
T2-COM x1,000 + First 1,000 + Conf <0.99	2.95%	3.16%	3.05%	37.48%	

3. EXPERIMENTS TASCA T3(SPC)

	Minim	Maxim	Mean	ASR	
T3-STD ENG	0.439%	0.489%	0.466%	100%	REF*
T3-STD GER	0.505%	0.598%	0.550%	100%	

	Minim	Maxim	Mean	ASR	
T3-PRE x1	0.591%	0.664%	0.627%	100%	
T3-PRE x5	0.455%	0.524%	0.490%	100%	DISC
T3-PRE x10	0.447%	0.522%	0.476%	100%	
T3-PRE x100	0.413%	0.491%	0.466%	100%	GOOD*
T3-PRE x1,000	0.441%	0.509%	0.467%	100%	GOOD*
T3-PRE x10,000	0.459%	0.584%	0.506%	20.5%	OVRFL

	Minim	Maxim	Mean	ASR	
T3-ACT Rand 1% + Conf <0.999	0.321%	0.396%	0.359%	29.9%	
T3-ACT Rand 1% + Conf <0.9999	0.359%	0.394%	0.377%	53.0%	
T3-ACT Rand 5% + Conf <0.6	0.459%	0.559%	0.514%	5.1%	
T3-ACT Rand 5% + Conf <0.9	0.352%	0.409%	0.379%	6.0%	GOOD
T3-ACT Rand 5% + Conf <0.95	0.316%	0.394%	0.356%	7.0%	GOOD
T3-ACT Rand 5% + Conf <0.99	0.308%	0.363%	0.335%	13.4%	GOOD*
T3-ACT Rand 5% + Conf <0.999	0.361%	0.412%	0.380%	32.2%	
T3-ACT Rand 5% + Conf <0.9999	0.356%	0.393%	0.376%	54.2%	

	Minim	Maxim	Mean	ASR	
T3-ACT First 100 + Conf <0.99	0.342%	0.405%	0.369%	9.1%	GOOD
T3-ACT First 100 + Conf <0.999	0.307%	0.485%	0.364%	28.6%	
T3-ACT First 100 + Conf <0.9999	0.334%	0.404%	0.367%	51.6%	
T3-ACT First 1,000 + Conf <0.99	0.307%	0.404%	0.363%	6.9%	GOOD
T3-ACT First 1,000 + Conf <0.999	0.266%	0.354%	0.317%	27.3%	GOOD*
T3-ACT First 1,000 + Conf <0.9999	0.334%	0.416%	0.377%	52.2%	

	Minim	Maxim	Mean	ASR	
T3-COM x10 + Conf <0.999	0.353%	0.397%	0.377%	24.7%	
T3-COM x100 + Conf <0.999	0.316%	0.426%	0.361%	28.0%	GOOD
T3-COM x10 + Rand 1% + Conf <0.999	0.340%	0.421%	0.380%	25.1%	
T3-COM x100 + Rand 1% + Conf <0.999	0.319%	0.422%	0.387%	29.8%	GOOD
T3-COM x10 + Rand 5% + Conf <0.999	0.350%	0.406%	0.390%	26.5%	
T3-COM x100 + Rand 5% + Conf <0.999	0.365%	0.433%	0.398%	30.4%	GOOD
T3-COM x10 + First 100 + Conf <0.999	0.331%	0.405%	0.376%	24.7%	
T3-COM x100 + First 100 + Conf <0.999	0.312%	0.404%	0.370%	28.3%	GOOD
T3-COM x10 + First 1,000 + Conf <0.999	0.310%	0.456%	0.383%	24.5%	
T3-COM x100 + First 1,000 + Conf <0.999	0.350%	0.421%	0.385%	28.8%	GOOD

4. EXPERIMENTS TASCA T5(BRK)

	Minim	Maxim	Mean	ASR	
T5-STD ENG	0.021%	0.036%	0.027%	100%	
T5-STD GER	0.167%	0.203%	0.183%	100%	

	Minim	Maxim	Mean	ASR	
T5-PRE x1	0.046%	0.077%	0.057%	100%	
T5-PRE x10	0.023%	0.041%	0.030%	100%	
T5-PRE x100	0.019%	0.035%	0.026%	100%	GOOD
T5-PRE x1,000	0.020%	0.035%	0.026%	100%	GOOD

	Minim	Maxim	Mean	ASR
T5-ACT Rand 1% + Conf <0.999	0.010%	0.019%	0.014%	1.15%
T5-ACT Rand 1% + Conf <0.9999	0.008%	0.019%	0.013%	1.25%
T5-ACT Rand 5% + Conf <0.999	0.008%	0.023%	0.016%	5.13%
T5-ACT Rand 5% + Conf <0.9999	0.007%	0.019%	0.014%	5.25%

	Minim	Maxim	Mean	ASR	
T5-ACT First 100 + Conf <0.999	0.004%	0.013%	0.009%	0.23%	GOOD
T5-ACT First 100 + Conf <0.9999	0.004%	0.017%	0.011%	0.64%	GOOD
T5-ACT First 1,000 + Conf <0.999	0.007%	0.016%	0.012%	0.20%	GOOD
T5-ACT First 1,000 + Conf <0.9999	0.007%	0.019%	0.011%	0.60%	GOOD

	Minim	Maxim	Mean	ASR	
T5-COM x10 + Conf <0.9999	0.011%	0.018%	0.015%	0.26%	
T5-COM x100 + Conf <0.9999	0.009%	0.017%	0.012%	0.12%	GOOD
T5-COM x10 + Rand 1% + Conf <0.9999	0.011%	0.019%	0.015%	1.29%	
T5-COM x100 + Rand 1% + Conf <0.9999	0.009%	0.019%	0.014%	1.15%	
T5-COM x10 + Rand 5% + Conf <0.9999	0.012%	0.021%	0.016%	5.32%	
T5-COM x100 + Rand 5% + Conf <0.9999	0.011%	0.018%	0.014%	5.18%	
T5-COM x10 +First 100 + Conf <0.9999	0.007%	0.020%	0.015%	0.26%	
T5-COM x100 +First 100 + Conf <0.9999	0.008%	0.019%	0.013%	0.13%	GOOD
T5-COM x10 +First 1,000 + Conf<0.9999	0.011%	0.021%	0.015%	0.33%	
T5-COM x100 +First 1,000 + Conf<0.9999	0.006%	0.017%	0.012%	0.19%	GOOD

BIBLIOGRAFIA

Aha, D. W. (1992). Tolerating Noisy, Irrelevant and Novel Attributes in Instance-Based Learning Algorithms. *International Journal of Man-Machine Studies*, *36*(2).

Aha, D. W. (1995). *Machine Learning: An Annotated Bibliography*. *Machine Learning*.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-Based Learning Algorithms. *Machine Learning*, *6*(1), 37-66. Springer.

Alpaydin, E. (2004). Introduction to Machine Learning. In *Adaptive Computation and Machine Learning*. The MIT Press.

Aseltine, J. H. (1999). WAVE: An Incremental Algorithm for Information Extraction. In *Proceedings AAAI Workshop on Machine Learning for Information Extraction*.

Banko, M., & Brill, E. (2001). Mitigating the Paucity-of-Data Problem: Exploring the Effect of Training Corpus Size on Classifier Performance for Natural Language Processing. *Computational Linguistics*, 2-6.

Banko, M., & Brill, E. (2001). Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *Annual Meeting of the ACL. Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 26-33). Toulouse, France: Association for Computational Linguistics.

Bayes, T. (1763). An Essay Towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London*, *53*, 370–418.

Blum, A. (1996). On-Line Algorithms in Machine Learning. In *In Proceedings of the Workshop on On-Line Algorithms, Dagstuhl* (pp. 306-325). Springer.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24*(2), 123-140. Springer Netherlands.

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics.

Brill, E., & Wu, J. (1998). Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1* (pp. 191-195). Association for Computational Linguistics.

Busser, B., Netherlands, T., & Morante, R. (2005). Designing an Active Learning Based System for Corpus Annotation. *Sociedad Española para el Procesamiento del Lenguaje Natural*, *35*, 375-381.

Bybee, J. L., & Hopper, P. (2001). Introduction to Frequency and the Emergence of Linguistic Structure. In *Frequency and the Emergence of Linguistic Structure* (pp. 1-24). John Benjamins Publishing Co.

Carlson, A. J., Rosen, J., & Roth, D. (2001). Scaling Up Context-Sensitive Text Correction. In *Proceedings of the Thirteenth Conference on Innovative Applications of Artificial Intelligence Conference* (pp. 45-50). AAAI Press.

Carvalho, V. R., & Cohen, W. W. (2006). Single-Pass Online Learning: Performance, Voting Schemes and Online Feature Selection. In *International Conference on Knowledge Discovery and Data Mining*. New York, USA: ACM.

Cauwenberghs, G., & Poggio, T. (2000). Incremental and Decremental Support Vector Machine Learning. *Advances in Neural Information Processing*, *13*.

Cawley, G. (2011). Baseline Methods for Active Learning. *JMLR: Workshop and Conference Proceedings 16(2011)*, 47-57. *Workshop on Active Learning and Experimental Design.*

Chinchor, N. (1992). MUC-4 Evaluation Metrics. In *Proceedings of the 4th Conference on Message Understanding* (pp. 22-29). Morristown, USA: Association for Computational Linguistics.

Chomsky, N. (1957). *Syntactic Structures* (2nd Edition). Walter de Gruyter.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. The MIT Press.

Ciravegna, F. (2001). (LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*. Seattle, USA.

Ciravegna, F., Dingli, A., Petrelli, D., & Wilks, Y. (2002). *Document Annotation via Adaptive Information Extraction* .

Ciravegna, F., Dingli, A., Petrelli, D., & Wilks, Y. (2002). Timely and Non-Intrusive Active Document Annotation via Adaptive Information Extraction. In *In Proc. Workshop Semantic Authoring Annotation and Knowledge Management (European Conf. Artificial Intelligence* (pp. 7-13).

Ciravegna, F., Dingli, A., Petrelli, D., & Wilks, Y. (2002). User-System Cooperation in Document Annotation Based on Information Extraction. In *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02* (pp. 122-137). Springer Verlag.

Cleary, J. G., & Trigg, L. E. (1995). K*: An Instance-based Learner Using an Entropic Distance Measure. In *In Proceedings of the 12th International Conference on Machine Learning* (pp. 108-114). Morgan Kaufmann.

Cohen, W. W. (1995). Fast Effective Rule Induction. In *In Proceedings of the Twelfth International Conference on Machine Learning* (pp. 115-123). Morgan Kaufmann.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving Generalization with Active Learning. In *Machine Learning* , 15(2), 201-221.

Compton, P., & Jansen, R. (1990). Knowledge in Context: a Strategy for Expert System Maintenance. *Proceedings of the second Australian joint conference on Artificial intelligence.*

Cornuéjols, A. (1993). Getting Order Independence in Incremental Learning. In P. Brazdil, *Proc. European Conference on Machine Learning Lecture Notes in Artificial Intelligence v. 667* (pp. 196-212). Springer-Verlag.

Cover, T., & Hart, P. (1967). Nearest Neighbor Pattern Classification. *Information Theory, IEEE Transactions on*, *13*(1).

Crammer, K., Dekel, O., Shalev-Shwartz, S., & Singer, Y. (2003). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, *7*.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Culotta, A., Kristjansson, T., McCallum, A., & Viola, P. (2006). Corrective Feedback and Persistent Learning for Information Extraction. *Artificial Intelligence*, *170*(14).

Daelemans, W. (2005). *Machine Learning of Natural Language. Machine Learning*. University of Antwerp.

Daelemans, W., Bosch, A. v., & Weijters, T. (1997). *IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms*.

Daelemans, W., Bosch, A. v., Van, A., Bosch, D., & Zavrel, J. (1999). Forgetting Exceptions is Harmful in Language Learning. In *Machine Learning, Special issue on Natural Language Learning* (pp. 34-11).

Dagan, I., & Engelson, S. P. (1995). Committee-Based Sampling For Training Probabilistic Classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning* (pp. 150-157). Morgan Kaufmann.

Dagan, I., Karov, Y., & T, D. R. (1997). Mistake-Driven Learning in Text Categorization. In *In EMNLP-97, The Second Conference on Empirical Methods in Natural Language Processing* (pp. 55-63).

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems* (pp. 1-15). London, UK: Springer-Verlag.

Dietterich, T. G., & Bakiri, G. (1995). Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, *2*, 263-286.

Dredze, M., Crammer, K., & Pereira, F. (2008). Confidence-Weighted Linear Classification. In *Proceedings of the 25th international conference on Machine learning* (Vol. 307, p. 7). Helsinki, Finland: ACM.

Duda, R., & Hart, P. (1973). Pattern Classification and Scene Analysis. *IEEE Transactions on Automatic Control*, *19*(4), 462-463. John Wiley & Sons Inc.

Elman, J. L. (1993). Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition: International journal of cognitive science*, *48*(1), 71-79.

Estévez, P. A., Tesmer, M., Perez, C. A., & Zurada, J. M. (2009). Normalized Mutual Information Feature Selection. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *20*(2), 189-201.

Finn, A., & Kushmerick, N. (2003). Active Learning Selection Strategies for Information Extraction. In *In Proceedings of the ECML-2004 Workshop on Adaptive Text Extraction and Mining*.

Fix, E., & Hodges, J. L. (1951). Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties, (Project 21-49-004, Report Number 4), 261 - 279.

Fleuret, F. (2004). Fast Binary Feature Selection with Conditional Mutual Information. *The Journal of Machine Learning Research*, *5*, 1531-1555.

Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. In *Proceedings of the Second European Conference on Computational Learning Theory* (pp. 23-37). London, UK: Springer-Verlag.

Fujii, A., Tokunaga, T., Inui, K., & Tanaka, H. (1998). Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics*, *24*, 24-4.

Fürnkranz, J. (2002). Pairwise Classification as an Ensemble Technique. In *Proceedings of the 13th European Conference on Machine Learning* (pp. 97-110). Springer Verlag.

Fürnkranz, J., & Hüllermeier, E. (2003). Pairwise Preference Learning and Ranking. In *Proceedings of the 14th European Conference on Machine Learning* (pp. 145-156). Springer-Verlag.

Gaines, B. R., & Compton, P. (1995). Induction of Ripple-Down Rules Applied to Modeling Large Databases. *Journal of Intelligent Information Systems*, *5*(3), 211-228. Kluwer Academic Publishers.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, *26*(5-6), 415-439.

Gasperin, C. (2009). Active Learning for Anaphora Resolution. In *Proceedings of the NAACL Human Language Technology Conference 2009 Workshop on Active Learning for Natural Language Processing* (p. 7). Boulder, USA: Association for Computational Linguistics.

Giraud-Carrier, C. (2000). A Note on the Utility of Incremental Learning. *AI Communications*, *13*(4), 215-223.

Giraud-Carrier, C., & Martinez, T. (1994). An Incremental Learning Model For Commonsense Reasoning. In *in Proceedings of the Seventh International Symposium on Artificial Intelligence (ISAI'94* (pp. 134-141).

Giraud-Carrier, C., & Martinez, T. (1994). Seven Desirable Properties For Artificial Learning Systems. In *In Proceedings of the Seventh Florida AI Research Symposium* (pp. 16-20).

Giraud-Carrier, C., & Martinez, T. (1995). ILA: Combining Inductive Learning with Prior Knowledge and Reasoning. Bristol, UK.

Giraud-Carrier, C., & Martinez, T. R. (2006). A Constructive Incremental Learning Algorithm for Binary Classification Tasks. In *Proceedings of SMCals/06* (pp. 213-218).

Gold, M. (1967). Language Identification in the Limit. *Information and Control*, *10*(5), 447-474.

Golding, A. R., Roth, D., Mooney, J., & Cardie, C. (1999). A Winnow-Based Approach to Context-Sensitive Spelling Correction. In *Machine Learning* (pp. 107-130).

Hall, M., Eiber, F., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. (2009). The WEKA Data Mining Software: An Update; *SIGKDD Explorations*, Vol. 11, Issue 1.

Heckerman, D. (1996). A Tutorial on Learning With Bayesian Networks.

Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. In *Machine Learning* (pp. 63-91).

Hong, J., Mozetic, I., & Michalski, R. S. (1986). AQ15: Incremental Learning of Attribute-Based Descriptions from Examples: The Method and User's Guide. *Department of Computer Science, University of Illinois*.

Horn, K., Lazarus, L., Compton, P., & Quinlan, J. (1985). An Expert System for the Interpretation of Thyroid Assays in a Clinical Laboratory. *Australian Computer Journal*, *17*(3).

Huang, S.J., Jin, R., Zhou, Z.H. (2010). Active Learning by Querying Informative and Representative Examples. In *Neural Information Processing Systems (NIPS) Foundation. Conferences 2010*.

Kalal, Z., Matas, J., & Mikolajczyk, K. (2008). Weighted Sampling for Large-Scale Boosting. *British machine vision conference*.

Kaynak, C., & Alpaydin, E. (2000). MultiStage Cascading of Multiple Classifiers: One Man's Noise is Another Man's Data. In *In Proceedings of the 17th International Conference on Machine Learning (ICML-2000* (pp. 455-462).

Knuth, D. E. (1969). Seminumerical Algorithms. In *The Art of Computer Programming, Vol. 2*. Addison Wesley.

Koehn, P. (2000). German-English Parallel Corpus De-News. v.0.9. *ELRA-A-U-W 0100*. http://homepages.inf.ed.ac.uk/pkoehn/publications/de-news/.

Kohavi, R. (1995). The Power of Decision Tables. In *Proceedings of the European Conference on Machine Learning* (pp. 174-189). Springer Verlag.

Lehnert, W. G. (1987). Case-Based Problem Solving with a Large Knowledge Base of Learned Cases. *AAAI-87 Proceedings*.

Leung, A. P., & Gong, S. (2005). Online Feature Selection Using Mutual Information for Real-Time Multi-view Object Tracking. In W. Zhao, S. Gong, & X. Tang, *AMFG*, Lecture Notes in Computer Science (Vol. 3723, pp. 184-197).

Li, Y., & Long, P. M. (2002). The Relaxed Online Maximum Margin Algorithm. *Machine Learning*, *46*(1-3).

Littlestone, N. (1988). Learning Quickly When Irrelevant Attributes Abound: A new Linear-Threshold Algorithm. *Machine Learning*, *2*(4), 285-318.

Littlestone, N., & Warmuth, M. K. (1994). The Weighted Majority Algorithm. *Information and Computation*, *108*(2), 212-261.

Lu, J., Yang, Y., & Geoffrey, W. (2006). Incremental Discretization for Naive-Bayes Classifier. *Lecture Notes in Computer Science*, *4093*, 223-238.

Macgregor, J. N. (1988). The Effects of Order on Learning Classifications by Example: Heuristics for Finding the Optimal Order. *Artificial Intelligence*, *34*(3).

Magerman, D. M. (1995). Statistical Decision-Tree Models for Parsing. In *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* (pp. 276-283).

Martinez, T. R., & Vidal, J. J. (1988). Adaptive Parallel Logic Networks. *Journal of Parallel and Distributed Computing*, *5*(1).

Martí, M. A., Taulé, M., Bertran, M., & Màrquez, L. (2008). AnCora: Multilingual and Multilevel Annotated Corpora. CLiC-UB (Centre de Llenguatge i Computació, Universitat de Barcelona).

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1* (pp. 307-314). Budapest, Hungary: Association for Computational Linguistics.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Science.

Màrquez, L. (2001). *Aprendizaje Automático y Procesamiento del Lenguaje Natural*. Departamento de Lenguajes y Sistemas Informáticos, Universitat Politècnica de Catalunya.

Ng, H. T. (1997). *Getting Serious about Word Sense Disambiguation*. Workshop On Tagging Text With Lexical Semantics: Why What And How?

Ng, H. T. (1997). Exemplar-Based Word Sense Disambiguation: Some Recent Improvements. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*.

Ng, H. T., & Lee, H. B. (1996). Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. In *In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics* (pp. 40-47).

Nilsson, N. J., Sejnowski, T. J., & White, H. (1965). *Learning Machines*. San Mateo, USA: Morgan Kaufmann.

Oza, N. C., & Russell, S. (2001). Online Bagging and Boosting. In *In Artificial Intelligence and Statistics 2001* (pp. 105-112). Morgan Kaufmann.

Parmanto, B., Munro, P. W., & Doyle, H. R. (1996). Reducing Variance of Committee Prediction with Resampling Techniques. *Connection science*, *8*(3-4), 405-425.

Pham, D. T., & Afify, A. A. (2005). Online Discretization of Continuous-Valued Attributes in Rule Induction. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *219*(8).

Pinto, C., & Gama, J. (2005). Partition Incremental Discretization. In *Portuguese Conference on Artificial Intelligence* (pp. 168-174). IEEE.

Quinlan, J. R. (1986). Simplifying Decision Trees. *International Journal of Man-Machine Studies*, *27*(3), 221-234.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann.

Raykar, V. C. (2005). Computational Tractability of Machine Learning Algorithms for Tall Fat Data. *Matrix*. Maryland, USA.

Raykar, V. C. (2007). Scalable Machine Learning for Massive Datasets: Fast Summation Algorithms. *Strain*. Maryland.

Rivest, R. L. (1987). Learning Decision Lists. *Machine Learning*, *2*(3), 229-246.

Rosenblatt, F. (1958). The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, *65*, 386-408.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books.

Roth, D., & Zelenko, D. (1998). Part of Speech Tagging Using a Network of Linear Separators. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2* (pp. 1136-1142). Montreal, Canada: Association for Computational Linguistics.

Sampson, G. (2002). Introduction to Empirical Linguistics. In *Empirical Linguistics (Open Linguistics)* (pp. 1-12). Great Britain: Continuum International Publishing Group Ltd.

Schapire, R. E., & Singer, Y. (1999). Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, *37*(3), 80-91.

Scheffer, T., Wrobel, S., Popov, B., Ognianov, D., Decomain, C., Hoche, S., et al. (2002). *Learning Hidden Markov Models for Information Extraction Actively from Partially Labeled Text*. Kuenstliche Intelligenz, Themenheft Textmining, 02.

Schlimmer, J., & Fisher, D. (1986). A Case Study of Incremental Concept Induction. In *In American Artificial Inteligence Proceedings* (pp. 496-501). Philadelphia, PA: Morgan Kaufman.

Sculley, D. (2007). Online Active Learning Methods for Fast Label-Efficient Spam Filtering. In *Fourth Conference on Email and AntiSpam*. Mountain View, USA.

Shao, J. (1993). Linear Model Selection by Cross-Validation. *Journal of the American Statistical Association*, *88*(422), 486 - 494.

Shen, D., Zhang, J., Su, J., Zhou, G., & Tan, C. (2004). Multi-Criteria-Based Active Learning for Named Entity Recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Barcelona, Spain: Association for Computational Linguistics.

Siefkes, C. (2005). *Incremental Information Extraction Using Tree-Based Context Representations*. *Computational Linguistics and Intelligent Text Processing* (pp. 510 - 521).

Siefkes, C. (2008). *An Incrementally Trainable Statistical Approach to Information Extraction: Based on Token Classification and Rich Context Model* (p. 220). VDM Verlag.

Simmons, R. F., & Yu, Y. (1992). The Acquisition and Use of Context-Dependent Grammars for English. *Computational Linguistics*, *18*(4).

Smola, A. J., Bartlett, P., Schuurmans, D., & Schölkopf, B. (2000). *Advances in Large Margin Classifiers*. Cambridge, USA: MIT Press.

Soderland, S., Fisher, D., Aseltine, J., & Lehnert, W. (1995). *CRYSTAL: Inducing a Conceptual Dictionary*.

Sung, K., & Poggio, T. (1998). Example-Based Learning for View-Based Human Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 39-51.

Thompson, C. A., Califf, M. E., & Mooney, R. J. (1999). Active Learning for Natural Language Parsing and Information Extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 406-414). San Francisco, USA: Morgan Kaufmann Publishers Inc.

Utgoff, P. (1989). Incremental Induction of Decision Trees. *Machine Learning*, *4*(2), 161-186.

Utgoff, P. E. (1995). Decision Tree Induction Based on Efficient Tree Restructuring. *Machine Learning*. Amherst, USA.

Van Den Bosch, A., & Buchholz, S. (2002). Shallow Parsing on the Basis of Words Only: A Case Study. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 433-440). Philadelphia.

Van Erp, M., Vuurpijl, L., & Schomaker, L. (2002). An Overview and Comparison of Voting Methods for Pattern Recognition. In *Hoboken(NJ), IEEE. Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (WFHR02* (pp. 195-200).

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Viterbi, A. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. {*IEEE*} *Transactions on Information Theory*, 260 - 269.

Weijters, A. (1991). A Simple Look-up Procedure Superior to NETtalk. In *Proc. of the international conference on artificial neural networks*. Espoo, Finland.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)* (2nd.). Morgan Kaufmann.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, *5*(2), 241-259.

Zhang, P. (1992). On the Distributional Properties of Model Selection Criteria. *Journal of the American Statistical Association*, *87*(419), 732 - 737.

Zhang, T. (2000). Large Margin Winnow Methods for Text Categorization. In *KDD-2000 Workshop on Text Mining*.