Machine Learning Methods for Understanding Social Media Communication: Modeling Irony and Emojis

Francesco Barbieri

TESI DOCTORAL UPF / ANY 2017

DIRECTOR DE LA TESI
Horacio Saggion
Departament DTIC

upf. Universitat Pompeu Fabra
Barcelona

To Andrea,
Semprepresente

Acknowledgements

The last four years have been definitely the best of my life. I had the opportunity to work in a wonderful university, with incredible colleagues, and on unique (and fun) research topics. It was a blessing, and I want to thank many people.

First, thank you Horacio! You made all this possible. You helped me when I needed it and let me free to explore different research areas when I felt like, leading me back to the right track when I was going too far. You have been an extraordinary mentor, thanks.

An important part of this journey was the visit at the University of Trento. I thank Marco Baroni for hosting me, and all members of the Clic group, especially German, The Nghia, Adam, and Angeliki. A very special thanks goes to Angeliki, you taught me so much: it was really important working with you.

I want to thank all the Snap researchers I have worked with in the last four months. Working in such fresh, dynamic and creative environment is incredible. Plus, seeing that a research can actually work on real life applications, is immensely rewarding. A special thank to Luis Marujo who gave me this opportunity (and also Aletta, who proofread part of this thesis).

I also want to thank all the UPF/TALN researchers, in particular my co-authors, Francesco Ronzan (I consider you my second supervisor, thanks to be excited every time I came up with a new idea), Mich Ballesteros (doing machine learning with you was great, I learned a lot from you), and Sergio Oramas (working on the music domain was very fun, even if we have to find someone to take care of the writing part, to be the perfect team).

I also want to thank the people who were with me most of the time during the last four years. Thank you Ruppero Carlini for your non-technical and technical support (which was about asking more questions than actually solving problems), thank you Dr Sun. Soler for all the kind and cordial words you always had for me, I know you believe in me, and also thank you Luis Pinox, I would not be as strong and successful as I am, if you would have not taught me the six rules of success.

Thank you to my Italian friends. I travel and move often, but you are always there for me.

I thank all the nice people I met here in Barcelona, from the sadakos and helmanas to the Casamor-Vidal family. Thank you Max, Mercedes, Marga and Carlos, you made me feel at home during these four years.

I also thank my family, who always believed in me. Thank you dad, mum, Samo and Geky: you are the best, I am so grateful to have you in my life.

Finally, I want to thank Maria, there is not much to say, you are my life.

Abstract

In this dissertation we propose algorithms for the analysis of social media texts, focusing on two particular aspects: irony and emojis. We propose novel automatic systems, based on machine learning methods, able to recognize and interpret these two phenomena. We also explore the problem of topic bias in sentiment analysis and irony detection, showing that traditional word based systems are not robust when they have to recognize irony on a new domain. We argue that our proposal is better suited for topic changes. We then use our approach to recognize another phenomenon related to irony: satirical news in Twitter. By relying on distributional semantic models, we also introduce a novel method for the study of the meaning and use of emojis in social media texts. Moreover, we also propose an emoji prediction task that consists in predicting the emoji present in a text message using only the text. We have shown that this emoji prediction task can be performed by deep-learning systems with good accuracy, and that this accuracy can be improved by using images included in the post.

Resumen

En esta tesis proponemos algoritmos para el análisis de textos de redes sociales, enfocándonos en dos aspectos particulares: el reconocimiento automático de la ironía y el análisis y predicción de emojis. Proponemos sistemas automáticos, basados en métodos de aprendizaje automático, capaces de reconocer e interpretar estos dos fenómenos. También exploramos el problema del sesgo en el análisis del sentimiento y en la detección de la ironía, mostrando que los sistemas tradicionales, basados en palabras, no son robustos cuando los datos de entrenamiento y test pertenecen a dominios diferentes. El modelo que se propone en esta tesis para el reconocimiento de la ironía es más estable a los cambios de dominio que los sistemas basados en palabras. En una serie de experimentos demostramos que nuestro modelo es también capaz de distinguir entre noticias satíricas y no satíricas. Asimismo, exploramos con modelos semánticos distribucionales cómo el significado y el uso de emojis varía entre los idiomas, así como a través de las épocas del año. También nos preguntamos si es posible predecir el emoji que un mensaje contiene solo utilizando el texto del mensaje. Hemos demostrado que nuestro sistema basado en deep-learning es capaz de realizar esta tarea con buena precisión y que se pueden mejorar los resultados si además del texto se utiliza información sobre las imágenes que acompañan al texto.

Contents

Figure Index **xiii**

Table Index **xvi**

1 INTRODUCTION **1**
1.1 Organization of this Dissertation 6
1.2 Contributions 7

2 RELATED WORK **9**
2.1 Irony in Social Media 9
2.1.1 Definition of Irony 9
2.1.1.1 Verbal Irony 10
2.1.1.2 Sarcasm 12
2.1.1.3 Satire 12
2.1.2 Irony Detection in Social Media 13
2.1.2.1 Definition of the Task 13
2.1.2.2 Short Text 14
2.1.2.3 Long Text 16
2.1.3 Computational Models 17
2.1.3.1 Rule Based 19
2.1.3.2 Feature Engineering 20
2.1.3.3 Deep Learning 21
2.2 Visual aspects of Social Media Content 22
2.2.1 Vision and Language 23
2.2.2 Emojis 24
2.2.2.1 Emoticons 25
2.2.2.2 Emoji Sentiments 26
2.2.2.3 Emoji Semantics 26

3 #IRONY AND #SARCASM DETECTION IN SOCIAL MEDIA 29
3.1 Introduction . . . 29
3.2 Dataset and Text Processing . . . 30
3.3 Model . . . 31
3.3.1 Frequency . . . 32
3.3.2 Written-Spoken . . . 32
3.3.3 Intensity . . . 33
3.3.4 Synonyms . . . 33
3.3.5 Ambiguity . . . 34
3.3.6 Sentiments . . . 35
3.3.7 Structure . . . 35
3.3.8 Bag of Words Baseline . . . 36
3.4 Experiments and Results . . . 36
3.4.1 First Experiment . . . 36
3.4.2 Second Experiment . . . 37
3.5 Discussion . . . 40
3.6 Conclusions . . . 43

4 TOPIC BIAS IN IRONY DETECTION AND SENTIMENT ANALYSIS 45
4.1 Introduction . . . 45
4.2 Dataset and Text Processing . . . 46
4.3 Model . . . 47
4.3.1 Word-Based . . . 47
4.3.2 Synsets . . . 47
4.4 Experiments and Results . . . 48
4.4.1 Task 1: Subjectivity Classification . . . 48
4.4.2 Task 2: Polarity Classification . . . 50
4.4.3 Task 3: Irony Detection . . . 51
4.4.4 Cross-Domain Experiments . . . 51
4.5 Discussion . . . 52
4.6 Conclusions . . . 53

5 SATIRE DETECTION 55
5.1 Introduction . . . 55
5.2 Dataset and Text Processing . . . 56
5.3 Model . . . 58
5.3.1 English Resources . . . 58
5.3.2 Spanish Resources . . . 58
5.3.3 Italian Resources . . . 59
5.3.4 Model Summary . . . 59

5.4 Experiments 61
5.4.1 Monolingual Experiments 61
5.4.1.1 English Experiments 62
5.4.1.2 Spanish Experiments 62
5.4.1.3 Italian Experiments 63
5.4.2 Cross-Lingual Experiments 64
5.5 Discussion 65
5.6 Conclusions 67

6 EMOJI SEMANTICS 69
6.1 Introduction 69
6.2 The Twitter Dataset 70
6.2.1 Size of the dataset and usage of emojis 71
6.2.2 Repeated and combined use of emojis in a tweet 72
6.3 Comparing the semantics of words and emojis: the skip-gram vector model 74
6.4 Experimental Framework 75
6.4.1 Nearest-Neighbour experiment 75
6.4.2 Similarity-matrix experiment 77
6.5 Results and discussion 77
6.5.1 Language-and-location analysis of emojis 77
6.5.1.1 Nearest-Neighbour experiment for language-and-location analysis 78
6.5.1.2 Similarity-matrix experiment for language-and-location analysis 78
6.5.2 Season-based analysis of emojis 83
6.5.2.1 Nearest-Neighbours experiment for season-based analysis 83
6.5.2.2 Similarity-matrix experiment for season-based analysis 84
6.6 Conclusions 86

7 EMOJI DETECTION 89
7.1 Introduction 89
7.2 Dataset and Task 90
7.2.1 Dataset 90
7.2.2 Task 91
7.3 Models 92
7.3.1 Textual Models 92
7.3.1.1 Bi-Directional LSTMs 92
7.3.1.2 FastText 94

7.3.1.3 Skip-Gram Vector Average 94
7.3.1.4 Bag of Words 95
7.3.2 Visual Models 95
7.4 Unimodal Twitter Experiments 95
7.4.1 First Textual Experiment 95
7.4.2 Second Textual Experiment 99
7.5 Multimodal Instagram Experiments 100
7.5.1 Feature Extraction and Classifier 100
7.5.2 Multimodal Emoji Prediction 101
7.5.3 What Does Each Modality Learn? 102
7.5.4 Which portions of an image are used to predict an emoji? . 103
7.6 Conclusions 105

8 CONCLUSIONS 107

List of Figures

1.1 Most retweeted tweets of Barack Obama 4
1.2 Tweets with emojis of Roger Federer 5

2.1 Multimodal content of news articles 23
2.2 Set of the 176 First Emojis 25
2.3 Set of the 140 Most Frequent New Emojis 26

6.1 Distribution of Repetition of Same Emoji 74
6.2 t-SNE of USA Emoji Embeddings 76

7.1 Confusion Matrix Human Evaluation 99
7.2 Confusion Matrix of Visual and Textual Systems 104
7.3 Class Activation Mapping Heatmap 104

List of Tables

2.1 List of irony detection models 18
2.2 List of irony detection models 19

3.1 Examples from the Dataset (Irony and Sarcasm Detection) 31
3.2 Comparison with State of the Art on irony detection 38
3.3 Cross topic experiments - Irony Detection Short Text 38
3.4 Cross topic experiments - Irony Detection Long Text 39
3.5 Irony Detection Information Gain 39
3.6 Irony Detection Information Gain 40
3.7 Irony and Sarcasm Differences 41

4.1 Results on SENTIPOLC Shared Task 48
4.2 Feature Analysis Irony Detection in Italian 49
4.3 Most Informative Words for SENTIPOLC 50
4.4 Cross-Domain Experiment on SENTIPOLC 51

5.1 List of real news and satirical news Twitter accounts 56
5.2 English Monolingual Experiments 63
5.3 Spanish Monolingual Experiments 63
5.4 Italian Monolingual Experiments 64
5.5 Cross-Languages experiments 65
5.6 Information Gain for Satire 66

6.1 Distribution of Tweets in the Emoji Dataset 72
6.2 Most Frequent Emojis over different Countries and Seasons 73
6.3 Nearest-Neighbour experiment with language-and-location datasets 79
6.4 10 Nearest-Neighbour Emojis of the Clover Symbol 80
6.5 Similarity-matrix experiment with language-and-location datasets 80
6.6 Similarity-matrix experiment with language-and-location datasets 81
6.7 Nearest-Neighbour experiment with season-based datasets 82
6.8 The 10 Nearest-Neighbour Emojis of the Pine Emoji 83
6.9 Similarity-matrix experiment with season-based datasets 84

6.10 Similarity-matrix experiment with season-based datasets 85

7.1 Results of Textual Emoji Prediction 96
7.2 Results of B-LSTM on 20 Emoji Prediction task 97
7.3 Results of Human Emoji Prediction Evaluation 98
7.4 Results of the Multimodal Emoji Prediction Task 101
7.5 Detailed Results of the 20 Emojis Multimodal Emoji Prediction Task 103

Chapter 1

INTRODUCTION

Throughout the last decade, our methods of communication have experienced a revolution comparable to the introduction of the postal service or the telegraph. In fact, many examples in the recent history of our society point to the fundamental role that new forms of social media communication have played in areas such as politics, disaster management, and citizen security. For instance, social networks were instrumental in coordinating “Yes We Can” volunteers for the 2008 U.S. election campaign, social outbreaks claiming justice during the Arab Spring in 2011, and rescue teams during the recent Hurricane Harvey. Social networks have impacted the way we communicate, as they enable us to instantly share ideas and experiences with people worldwide.

However, language in social media differs from standard written language because it is usually shorter and noisier, a sort of spoken written language, with slangs, shortened words, and grammar violations. Automatic systems tend to struggle to process social media content because it is not structured by written language. Moreover, jokes and figurative language such as metaphors, irony, humor, sarcasm, or similes are common practices in social media, making the language processing tasks of automatic systems difficult.

Figurative language is one of the most powerful tools of human language. The listener has to understand both literal and figurative meanings in order to grasp the intended meaning of figurative expressions. Social media users themselves have developed formulas to convey the intended meaning without explicitly explaining the context. For example, users introduced a specific paralinguistic device called “hashtag” in order to complement a figurative language message with tags such as #irony and #sarcasm and to avoid confusion in communication.

The other great paralinguistic resource for making explicit the implicit are **emojis**, they became a key component to reveal the implicit in social media content. Emojis[1] recently started to be used to emphasize or express the true intention behind an ambiguous message. In this dissertation we study the language of social media, focusing on two particular aspects: irony and emojis. We propose novel automatic systems based on machine learning algorithms that are able to recognize and interpret these two phenomena. We first describe the dilemma of irony and later we will explore our research on the use of emoji.

Although the linguistic device of irony is complex, we argue that ironic words signify meanings that are alternative or even contrary to their literal definitions. Indeed, ironic expressions hold two meanings: the superficial literal meaning and the intended meaning. Only the people who understand irony are able to grasp the real meaning of an ironic expression. As an example, we can consider the release of the movie *Sharknado: Enough Said* in July 2013 by the American channel Syfy. Sharknado is meant to be a surrealistic comedy-disaster movie with a simple plot: a tornado made of sharks hits and destroys Los Angeles. On the night of the film's release, nearly 20% of the Twitter users communicating about TV shows mentioned Sharknado. During the show, more than 500 tweets per minute were posted about the film. What is relevant is that people did not write serious tweets, but rather tweets like the following:

1. *There are sharks in the street! omg this is beautiful!!* ***#irony*** #sharknado
2. *I've just watched Sharknado, best-movie-ever* ***#irony***

As we can see, these two users wrote one thing but meant another. They wrote that Sharknado is "beautiful" and "the best movie ever," but they were not serious about their positive reviews. In fact, they even flagged the tweet as ironic with the hashtag "#irony." Therefore, in order to understand these perceptions of Sharknado, it is necessary to understand the irony of the tweets.
Irony in social media is used not only in regard to comedies like Sharknado but also in a variety of other contexts. For instance, after the release of the iPhone X in October 2017, a Twitter user posted: *"I'm sure I will use iphone X in 2025, the future is so bright!"*. This tweet is ironic and thus should not be interpreted literally. The real meaning of the tweet is not that the user is going to use the new iPhone in 2025, but rather, that the user is disinterested in buying the new iPhone in the near future.

The problem with this mode of communication is that it is difficult for machines to understand irony. Machines are designed to process large amounts of data but not to understand jokes. This is why our work focuses on converting these tools from "implicit semantics" into explicit codes that are interpretable for

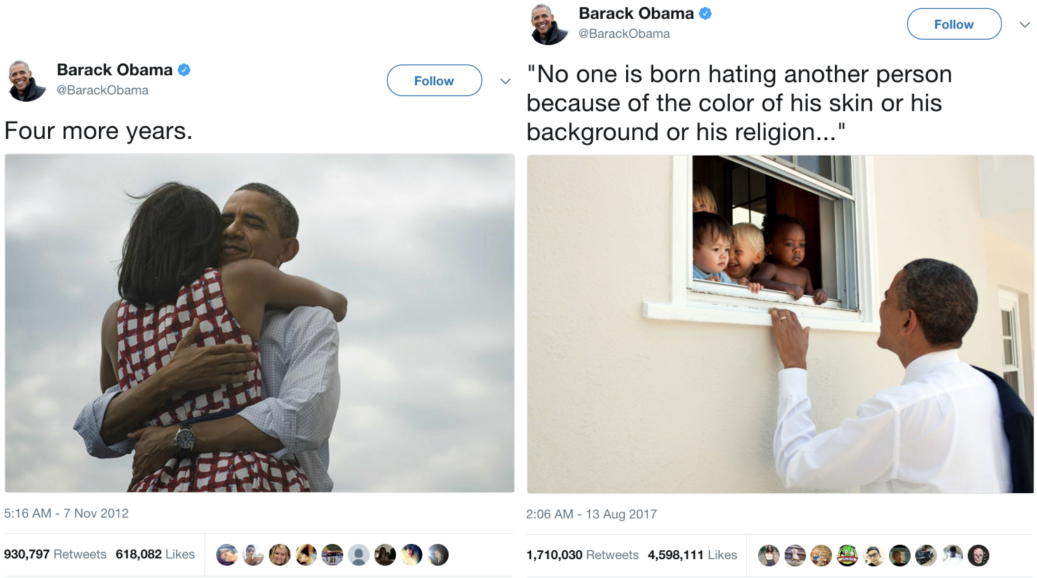
[1] Emojis were initially used in Japanese mobile phones in 1999, in Section 2.2.2 we report more details on the history of emojis.

the machine, so that it can better understand the intended meaning of the message. For these reasons, we propose a new method to recognize irony in social media and determine whether or not a tweet is ironic. **Irony detection** systems have many applications, especially in *opinion mining* (also known as *sentiment analysis*). Opinion mining aims to identify the attitude and sentiment of a speaker through automatic systems. Systems that automatically recognize human opinions are widely used in social media data sets that are too large to be manually processed. For example, a company may be interested in public impressions and opinions of its new product launch. Similarly, a politician may want to understand which potential electors care most about his or her campaign. In these contexts, the understanding of irony is crucial. If irony is not understood, a negative sentence may be erroneously interpreted as positive and provide incorrect data to companies or individuals. Opinion mining systems that are able to deal with irony are limited.

When we started this research, we set out to fill gaps in the important research field of computational models of irony. Existing irony detection systems had two noteworthy limitations: (1) most of the existing systems relied on words and tended to be topic-dependent, (2) most of the research on irony detection was focused on American English and other languages or cultures were little considered.

In this dissertation we propose a new approach that attempts to fill these two gaps. We focus our studies on more than one language, including American English, British English, Iberian Spanish and Italian. Also, we propose a machine learning system that is designed to be topic-independent, as it does not rely on specific words to recognize irony. Instead of statistically learning the probable combinations of words present in ironic and non-ironic examples, we rely on features that instead aim to model a key factor of irony: "unexpectedness" [Lucariello, 1994]. We believe that ironic tweets include unexpected incongruous elements to surprise the reader and suggest that the tweet should not be taken literally. As demonstrated by the iPhone example ("I will use iphone X in 2025"), the year 2025 is unexpected in this context and conveys that the tweet is not literal. We model unexpectedness in various ways, such as by examining register inconsistencies. In particular, we look at written and spoken registers used in the same tweet. Usually a tweet is either written with a formal register like that of a newspaper ("An angry confrontation. Fears of recordings. Trump's lawyers are clashing over cooperating with the Russia inquiry."[2]) or with an informal spoken register ("Sorry can't make it dude haha.."). However, the presence of two registers in the same tweet creates a surprise effect that suggests irony ("haha dude! me and my gf never have *angry confrontations...*"). We argue that within the framework of

[2] From the New York Times Twitter account, @nyt



Barack Obama @BarackObama

Four more years.

5:16 AM - 7 Nov 2012

930,797 Retweets 618,082 Likes

Barack Obama @BarackObama

"No one is born hating another person because of the color of his skin or his background or his religion..."

2:06 AM - 13 Aug 2017

1,710,030 Retweets 4,598,111 Likes

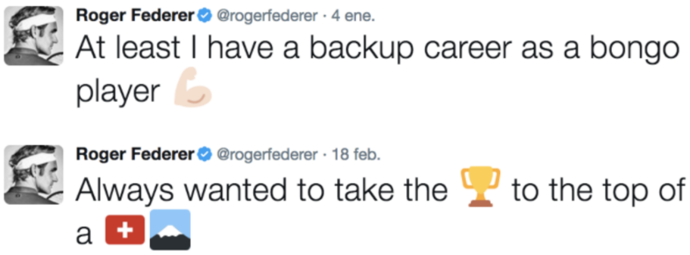
Figure 1.1: The two most re-tweeted tweets of Barack Obama's Twitter account (@BarackObama), they both include pictures.

these features we can model irony with improved accuracy.

It should be acknowledged that in this thesis we focus on signals of irony instead of analyzing semantics of the tweets. At the same time, several researchers claim that irony can not be understood without understanding the context of an ironic utterance. [Burgers, 2010], supported by findings of [Sperber and Wilson, 1981, Grice, 1978], states that "irony is never found in a context-free vacuum and it is not possible to say that an individual utterance is ironic or not." [Wallace, 2015] defends a similar idea, conveying that people infer irony when they recognize an incongruity between an utterance and what is expected about the speaker and the environment (the context). However, he also adds that "only listeners with a sufficient understanding of this context will recognize irony, unless the speaker signals ironic intent in other ways, e.g., via surfaces cues." Supported by our experiments, we demonstrate that these surfaces cues are important for interpreting social media communication. The authors of social media messages must delineate irony clearly because they tend to communicate with an extensive and anonymous audience[3].

In regard to the visual content of social media, photos and images are also becoming extremely important to online communications. Take as an example the two most re-tweeted tweets of Barack Obama in Figure 1.1: they both include a

[3] Specially in Twitter, where every user can see the content of any other user.



Roger Federer @rogerfederer · 4 ene.

At least I have a backup career as a bongo player 💪

Roger Federer @rogerfederer · 18 feb.

Always wanted to take the 🏆 to the top of a 🇨🇭🏔

Figure 1.2: Two tweets of Roger Federer (@rogerfederer) that include emojis.

picture. In both tweets the image is essential to understanding the message's emotional content because it provides easily legible information about the moment and emotions. Over the past few years, the sharing and exchanging of images has driven social media activity across the popular platforms Snapchat[4] and Instagram[5]. In this visual context, pure text communications are converted into a novel way of communication. Social media users combine text messages and visual enhancements, the so-called emojis, in order to make implicit messages explicit.

This visual language is as of now a *de-facto* standard for online communication on social network platforms. The tweet of Roger Federer in Figure 1.2 demonstrates that emojis can be used for various reasons, such as quickly responding in chats (👍) or suggesting the emotion of a text message. The fist emoji 💪 conveys the feeling of strength that is essential to the user's message. The second tweet about Roger Federer reveals that emojis can be used instead of words. In this example, emojis are used in place of the words "cup" and "Swiss Alps."

Despite the widespread use of emojis, this form of communication has been scarcely approached from a computational standpoint. In this dissertation, we aim to investigate how emojis are used and how words and emojis are related.

To study the semantics of emojis we employ unsupervised machine learning approaches that learn the meaning of a word by examining the co-occurrences of words. We use distributional semantics algorithms that rely on the Distributional Hypothesis [Harris, 1954, Miller and Charles, 1991], that is: "linguistic items with similar distributions have similar meanings." For instance, the words "fork" and "knife" share similar meanings because they often appear in similar contexts. We were able to study the meaning of the emojis used across the U.S. and Europe by using an extensive amount of tweets. For example, we demonstrate that the emoji 🍀 is frequently used in the context of friendships and romantic relationships in Spain. Meanwhile, in England, this symbol is mainly used to represent Ireland and "good luck."

[4] https://www.snap.com

[5] https://www.instagram.com

Another challenge is predicting the emojis that are most likely to be associated with a social media post (a text or an image, or both). This task is important in social media language interpretation because it helps machines to understand the emotional content of messages, as emojis are often used to explicitly label and reveal the intended emotion of the message. Overall, an artificial intelligence system that communicates with humans should be able to process and use this new visual language of emojis.

1.1 Organization of this Dissertation

In this section we provide a brief summary of the nine chapters that compose this dissertation.

Chapter 2 introduces the core topics of irony and emojis on social media. We first describe the linguistic device of irony and its past definitions, before overviewing previous studies on automatic irony detection of social media posts. Additionally, we discuss previous attempts to model on multimodal contents, such as the processing of combined images and text to build better computational systems. Finally, we introduce the emoji phenomenon by explaining the history of emojis and computational systems that aim to model emojis.

In Chapter 3, we aim to determine whether or not a tweet is ironic. We present a novel approach to detect ironic language, showing that without the direct use of words (typical Bag of Words systems) it is possible to more accurately predict emojis. It is also possible to build context-independent classifiers that predict irony independently of context.

In Chapter 4, we explore the possibility that irony detection and sentiment analysis systems are biased by their domains. We observe that state of the art systems focus on predicting topics instead of predicting actual phenomena like irony. However, our proposed model is significantly less affected by the domain.

In Chapter 5, we describe experiments in regard to satirical news detection. We demonstrate that it is possible to recognize if a news post on Twitter is satirical or genuine.

In Chapter 6, we study emojis by exploring its meanings and usages across seasons and across the countries of USA, UK, Spain and Italy. In Chapter 7, we propose an emoji detection task. By evaluating our classifiers, we analyze which emojis are easy to predict and, conversely, which are difficult (or perhaps impossible) to predict.

Finally, in Chapter 8 we conclude our dissertation with a summary of the results and also possible venues for future research.

1.2 Contributions

We successfully published several papers in scientific conferences regarding the topics of this dissertation, and also participated and organized shared task challenges on sentiment analysis and irony detection. The list of the works is ordered chronologically, starting with the most recent ones.

1. Barbieri F, Ballesteros M. and Saggion H. **Are Emojis Predictable?**, European Chapter of the Association for Computational Linguistics, EACL, 2017 - Chapter 7.

2. Barbieri F, Ballesteros M. and Saggion H. **Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes**, Proceedings of the 3rd Workshop on Noisy User-generated Text at EMNLP, 2017, (Best Paper Award)

3. Barbieri F, Kruszewski G, Ronzano F, Saggion H. **How cosmopolitan are emojis? Exploring emojis usage and meaning over different languages with distributional semantics**, ACM Multimedia Conference, ACMMM, 2016 - Chapter 6.

4. Barbieri F, Basile B, Croce D, Nissim M, Novielli N, Patti V. **Overview of the EVALITA 2016 SENTiment POLarity Classification Task**, Organization of the Shared Task on Sentiment Analysis and Irony detection in Italian at EVALITA, 2016

5. Barbieri F, Espinosa-Anke L, and Saggion H. **Revealing patterns of Twitter emoji usage in Barcelona and Madrid**, Catalan Conference on Artificial Intellicence, CCIA, 2016 - Chapter 6.

6. Barbieri F, Ronzano F, and Saggion H. **What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis**, Language Resources and Evaluation, LREC, 2016 - Chapter 6

7. Barbieri F, Ronzano F, and Saggion H. **Do We Criticise (and Laugh) in the Same Way? Automatic Detection of Multi-Lingual Satirical News in Twitter**, International Join Conference on Artificial Intelligence, IJCAI, 2015 - Chapter 5

8. Barbieri F, Ronzano F, and Saggion H. **Is This Tweet Satirical? A Computational Approach for Satire Detection in Spanish**, 2015, Spanish Society for Natural Language Processing Conference, SEPLN, (Best Paper Award) - Chapter 5.

9. Barbieri F, Ronzano F, and Saggion H. **Summarization and Information Extraction in your Tablet**, SEPLN, 2015
10. Barbieri F, Ronzano F, and Saggion H. **How Topic Biases Your Results? A Case Study of Sentiment Analysis and Irony Detection in Italian.**, RANLP, 2015 - Chapter 4
11. Guerrero I, Verhoeven B, Barbieri F, Martins P, Pérez R. **TheRiddlerBot A next step on the ladder towards creative Twitter bots**, International Conference on Computational Creativity, ICCC, 2015
12. Barbieri F, Ronzano F, and Saggion H. **UPF-taln: SemEval 2015 Tasks 10 and 11 Sentiment Analysis of Literal and Figurative Language in Twitter**, SemEval at NAACL, 2015 (Second Best System)
13. Barbieri F, Ronzano F, and Saggion H. **Italian irony detection in twitter: a first approach**, Conference on Computational Linguistics CLiC-it, 2015 - Chapter 5
14. Barbieri F, Ronzano F, and Saggion H. **Automatic Detection of Irony and Humour in Twitter**, International Conference on Computational Creativity, ICCC, 2014
15. Barbieri F, Ronzano F, and Saggion H. **Modelling Sarcasm in Twitter, a Novel Approach**, Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, WASSA at ACL, 2014 - Chapter 3
16. Barbieri F, Ronzano F, and Saggion H. **Modelling Irony in Twitter: Feature Analysis and Evaluation**, Language Resources and Evaluation, LREC, 2014 - Chapter 3
17. Barbieri F and Saggion H. **Modelling Irony in Twitter**, Student Research Workshop at EACL, 2014 - Chapter 3

Chapter 2

RELATED WORK

In order to provide a thorough overview, we present the most relevant studies for this dissertation. We begin by reviewing the irony detection problem (Section 2.1), introducing the theories of irony, and presenting the most salient computational approaches to model irony. In Section 2.2, we review previous attempts to model visual and textual information together, with a special focus on emojis (Section 2.2.2). The use of emojis is relatively recent and research on this field is still at an early stage; however, there are already some interesting findings on the use of emojis that we report in the dedicated section.

2.1 Irony in Social Media

Irony is a diffuse phenomenon in social media text, and it has been studied in various works. In this section we focus on the theories of irony (Section 2.1.1), reviewing the definitions of irony proposed in the past. In Section 2.1.2 we describe the problem of irony detection in social media, and in the last section (Section 2.1.3) we report previous attempts to model irony computationally in social media contents.

2.1.1 Definition of Irony

In the past, numerous studies from different fields (e.g. philosophy, psychology, linguistics) attempted to describe irony proposing various definitions, however, an common definition accepted by the whole research community does not seem to exist yet.

In the literature, two types of ironies are considered: *situational irony* and *verbal irony*. We now introduce the two types of ironies and we describe in detail verbal irony (Section 2.1.1.1) as it is the type of irony studied in this dissertation.

Situational irony is an unexpected or incongruous event in a specific situation that fail to meet an expectation [Lucariello, 1994, Littman and Mey, 1991, Shelley, 2001]. Consider the example of the firefighters of Station 20 in Las Vegas, who left some chicken fingers cooking when they left the station for a fire alarm [Shelley, 2001]. This situation is quite unexpected, as the firemen are usually the ones who extinguish the fire, not the ones who start the fire. For this reason the situation fail to meet this expectation and is considered ironic. Also, we can note that situational ironies are not necessarily intentional or planned. Ironic situation can occur through unintended and unexpected circumstances or through the evolution of situations.“Situational irony focuses on the surprising and inevitable fragility of the human condition, in which the consequences of actions are often the opposite of what was expected.” [Grant, 2004]

On the other side, **verbal irony** does not relate to an unplanned situation. In general we can say that verbal irony is a figure of speech where the intended meaning of a statement differs from the meaning that the words seem to express. For example:

> *“Bush sent more troops than Obama to create Peace in Afghanistan but Obama got the Nobel!”*

In the above example the speaker ironically underlines that the use of military troops to create peace is conflicting and does not make sense. President Obama received the Nobel Peace price even though he sent less troops than Bush to *create peace* in Afghanistan.

In this dissertation we focus on computational models for verbal irony, for this reason we will review in detail this type of irony in the next Section (Section2.1.1.1).

2.1.1.1 Verbal Irony

Verbal irony is typically defined as a rhetorical trope where the speaker says the opposite of what he/she means [Quintilian and Butler, 1959]. As simple example consider the situation in which a speaker is talking in front of someone who is not listening:

> *“I love when people do not listen to me.”*

This sentence is likely to be ironic as most of the people like to be listened, and not be ignored, when they talk. Hence, in this case the speaker is saying the exact opposite of what he/she means: “I do not like when people do not listen to me”. This case is then covered by the [Quintilian and Butler, 1959] definition of irony. However, this definition is limited and does not cover all the examples of irony.

Consider for instance the following fictional situation, where Jack says to Sam during a rainy day at the beach:

> *"Damn, we forgot the sunscreen!"*

In this case, Jack does not mean the opposite of what he literally says (as the opposite of the sentence would be "we brought the sunscreen"). What Jack really means is that the weather is bad, but how can Sam know that? [Grice et al., 1975] improves the [Quintilian and Butler, 1959] definition, stating that verbal irony is a violation of the conversational *maxim of Quality*, that is: "do not say what you believe to be false" (any form of negation with no negation markers). In the example above the listener (Sam) can understand that Jack is ironic by knowing that Jake can not be serious and really believe that the sunscreen is necessary as it is raining, and that Jack is transgressing the maxim of Quality: he is saying something he believe to be false.

[Grice et al., 1975] theory was criticized by [Wilson and Sperber, 1992] who state that not all ironies are negation of what is thought. They report various examples, like the following:

> Two guests are invited to Tuscany in May, but at they arrival it is windy and rainy, thus one guest says: *"Ah, Tuscany in May!"*

In this example the speaker is being ironic, but he is not violating the maximum of quality as the opposite of the sentence *"Ah, Tuscany in May!"* does not make sense. Hence, [Sperber and Wilson, 1981, Wilson and Sperber, 1992] propose a new theory to cover and explain also these types of ironies. They state that ironic sentences are cases of *echoic mention*. Ironic statements implicitly allude to some real or hypothetical proposition previously said, to demonstrate its absurdity in the current context. This is the case of the ironic speaker that says "Ah Tuscany in May", as this is an echo of someone who might have said this in the past, as in fact, weather in Tuscany is usually nice. A more straight example of echoic irony would be:

> Jack: *"Nice party, what do you think?"*
> Sam (who did not like the party): *"Yeah, nice party..."*

In this case Sam is echoing the statement of Jack in an ironic way as the party was not fun in his opinion.

Irony was also defined as a *pretense* (among others, [Clark and Gerrig, 1984, Kumon-Nakamura et al., 1995, Currie, 2006]). In this view, as summarized in [Burgers, 2010], irony has two voices: a speaker who pretend to be ignorant, and a listener, who is able to see this pretense of ignorance, and understand that the speaker is ironic. However, also this view is weak in some cases, as not all the pretenses of being ignorant hide ironic intents.

Following the *pretense* idea [Kumon-Nakamura et al., 1995] propose the *allusional pretense* theory, where a speaker is recognized as ironic as he is insincere and fail some expectation. The insincere aspect is added to cover the examples (unlike in [Grice et al., 1975]) where the ironic speaker says something true, but the intention is to say something else. In the example reported by [Kumon-Nakamura et al., 1995], there is a talented but arrogant student who is dominating the classroom discussion, and another student says "You sure know a lot", which is probably literally true, but the sentence is ironic for the pragmatic insincerity [Wallace, 2015].

2.1.1.2 Sarcasm

Among theorists there is no agreement on the exact definition of sarcasm, and on what is its relation to verbal irony. For [Fowler, 2010], the essence of sarcasm is *"the intention of giving pain by (ironical or other) bitter words"*, and that sarcasm does not necessarily involve irony. However, sarcasm is often associated to verbal irony, since the literal meaning of sarcastic sentences is usually in opposition with the intended meaning [Grice et al., 1975]. Sarcasm is considered as a subset of verbal irony by many researchers (among others [Brown, 1980, Gibbs and O'Brien, 1991, Kreuz and Roberts, 1993, Veale et al., 2013]). In particular, sarcasm is defined as a meaner form of irony as it tends to be offensive and directed towards other people. [Marina and Peter, 2010] add that sarcasm is an insincere form of politeness, used to insult or hurt the listener. As example of sarcasm consider Jack who says something obvious and Sam who replays:

> *"Really Sherlock? You are so clever!"*

Sam with this utterance is ironic as he does not mean what he literally says because what Jack said was not clever at all, but just obvious. But sam is being also sarcastic as there are intentions of hurting jack by saying that he is not clever.

Other researchers state that the two phenomena should not be separated [Littman and Mey, 1991, Dynel, 2014, Camp, 2012], using the terms sarcasm and verbal irony interchangeably.

2.1.1.3 Satire

In this dissertation we also explore a specific case of irony usage: satire (see Chapter 5). Satire is a form of communication where irony is used to criticize someone's behaviour and ridicule it. Satirical authors may be aggressive and offensive, but they *"always have a deeper meaning and a social signification beyond that of* *the humour"*[Colletta, 2009]. Satire does not make sense when the audience does not understand the real intents hidden in the ironic/funny dimension, as the key message of a satirical messages lays in the figurative interpretation of the message.

Satire is an important literature genre[1] used to criticize foolishness and corruption of an individual or a society. This genre was studied in several contributions in the past [Peter, 1956, Mann, 1973, Knight, 2004, LaMarre et al., 2009]. A well known example of satirical document is the essay “A Modest Proposal” written by Jonathan Swift in 1729. In this essay, the author pretends to be a member of the English ruling class, who proposes to solve the Irish economic struggles by serving poor Irish children as food to the rich people. The reader has to understand the pretense of the writer to be ignorant (who does not know that children cannot be eaten), in order to get the ironic intents of the essay, and understanding the real message the author wants to send. At page 207, Swift writes:

> *I am assured by a very knowing American of my acquaintance in London; that a young healthy child, well nursed, is, at a year old, a most delicious, nourishing, and wholesome food; whether stewed, roasted, baked or boiled, and I make no doubt, that it will equally serve in a fricassee, or ragout*

The proposal of Swift is clearly no sense, and very shocking. By shocking the reader, Swift wants him to know that people should be threaten with kindness and compassion. Even if very interesting, we do not focus on literature in this dissertation, as we only cover aspect of satire in short messages in the context of social media.

2.1.2 Irony Detection in Social Media

In this section we describe how the problem of recognizing irony has been approached computationally in the past. We show the different type of tasks proposed (Section 2.1.2.1), and the datasets employed (Section 2.1.2.2 and Section 2.1.2.3).

2.1.2.1 Definition of the Task

The automatic irony detection problem is typically tackled as a classification problem which consists in recognizing if a sentence is ironic or not. This is a binary classification where the classifier has to distinguish between an ironic and a non ironic class. There are several works in this direction (the one presented in this dissertation included), which we describe in detail in Section 2.1.3.

[1] To name a few popular satirical author: Terry Pratchett, Tom Sharpe, Francisco de Quevedo

An extension of this task was proposed by [Joshi et al., 2016c]: instead of classifying a single sentence, they classify several sentences in a dialogue. In a dialogue, irony can be detected by considering the sequence of the sentences. This can be done by tackling the problem of irony detection as a sequence labelling task, where each sentence is an element of the sequence, that can be ironic or not.

Another task to model irony was proposed by [Ghosh et al., 2015]. They propose to tackle irony detection as a sense disambiguation task. Given a sentence the task consist in detecting if a target word is used ironically or not (they call this task "Literal/Sarcastic Sense Disambiguation task"). They report this example:

" I am so happy that I am going back to the emergency room"

The task consists in detecting if the target word "happy" is used in an ironic way or not. In this case "happy" is used ironically, since the speaker does not mean that is he/she is literally happy, indeed, he/she is somehow sick in an emergency room.

2.1.2.2 Short Text

Most of the automatic detection models proposed in the past focus on microblogs like Twitter. Twitter allows to post messages of 140 characters[2]. Twitter is very popular in academia research since the data are accessible to everyone (no need to be "friends" to see the content posted by a user), and also because the Twitter APIs allow to download big amounts of data easily. Twitter APIs allow to retrieve tweets posted in a certain geographic area, posted by a certain user or that contain some special keyword. The latter option has been used by many researchers to retrieve ironic tweets by downloading tweets containing the tokens "#irony", "#ironic", "#sarcasm" or "#sarcastic". The assumption of the researchers using this kind of retrieval was that if a user includes "#irony" in the tweet, it means that the tweet is ironic. However, this is not always the case. For example the tweet "The hashtag #irony has increased popularity" is not ironic. For this reason [González-Ibánez et al., 2011], who retrieve tweets with the hashtag "sarcasm", manually check the tweets to assure the quality of the data (removing spam and not sarcastic examples). The non-sarcastic dataset was composed by tweets that contained hashtags relative to positive emotions ("#happy", "#joy") and negative emotions ("#sad", "#anger").

Other researchers used a similar approach by retrieving tweets with the hashtag #irony or #sarcasm and later manually assure the quality of the tweets, removing non-ironic examples from the dataset (among others [Riloff et al., 2013, Maynard

[2] On May 24, 2016 emojis, links and user mentions are not counted in this limit.

and Greenwood, 2014, Bamman and Smith, 2015, Fersini et al., 2015, Abercrombie and Hovy, 2016]). Several other studies use the hashtag to retrieve ironic tweets, but do not assure the quality of the data manually [Davidov et al., 2010, Reyes et al., 2012, Liebrecht et al., 2013, Maynard and Greenwood, 2014, Bouazizi and Ohtsuki, 2015, Bharti et al., 2015].
[Ghosh et al., 2015] also retrieve 8,000 tweets that include the keywords "#irony", "#sarcasm", and "#not" and they manually annotate these tweets with sentiment scores from -5 to 5 (-5 very negative, +5 very positive), and use this dataset to perform sentiment analysis on figurative language (SemEval task 11 in 2015, "Sentiment Analysis of Figurative Language" in Twitter).

[Abercrombie and Hovy, 2016] study whether the #sarcasm based datasets are somehow biased by the use of the hashtag, i.e. to what extent the #sarcasm datasets cover the sarcasm phenomena on social media. They compare the performances of a state of the art system (see Section 2.1.3) trained on an hashtag (#sarcasm) based datasets and train on another dataset, manually annotated for sarcastic tweet. They show that the automatic system performs better on the hashtag based dataset. As possible interpretation, they state that this type of data might be more homogeneous and easy to model since it is often certain types of users (who do not know their audience personally), who feel the need to label their sarcastic statements with hashtags. Hence, the #sarcasm datasets might cover only examples of sarcasm in which the target of the message is not know (big audience), but do not cover sarcastic examples where the messages are targeted to restricted group of people.

Some studies also aimed to contextualize tweets, retrieving for each ironic tweet also additional tweets from the same user. [Bamman and Smith, 2015] and [Khattri et al., 2015] include a maximum of 32,000 tweets from the author timeline, while [Rajadesingan et al., 2015] include 80 tweets from the author timeline.

Most of the dataset studied in the literature are in English, but some studies has been carried out to automatic detect irony in other languages. [Carvalho et al., 2009] study irony in Portuguese, [Liebrecht et al., 2013] in Dutch, [Lunando and Purwarianti, 2013] in Indonesian, [Liu et al., 2014] in Chinese, [Ptácek et al., 2014] in Czech, [Charalampakis et al., 2016] in Greek, [Desai and Dave, 2016] in Hindi, [Karoui et al., 2017] in French (and we explored also Italian and Spanish, see Chapter 4 and 5).

In this dissertation we compiled several datasets for carrying out our experiments, however, we also use existing available datasets to compare the ironic detection system proposed here, with state of the art, that we described below. The first one was compiled by [Reyes et al., 2013], who retrieve 10,000 ironic tweets with the hashtag "#irony" and other 30,000 tweets for the negative non -ironic class. The negative class was composed by three topics equally divided: "#politics", "#education" and "#humor". They suggest that in these last three topics the

ironic example are not very frequent, and they perform a binary classification, in order to show the effectiveness of their model. We use part of this dataset in the experiments shown in Chapter 3.
The second one was published by [Bosco et al., 2013b]. It consists of 1,159 tweets from a famous twitter account called "Spinoza"[3] that post ironic tweets in Italian. [Bosco et al., 2013b] state that there is a sort of collective agreement about the fact that Spinoza's posts include ironic intents, and these posts represent a natural way to extend the sampling of ironic expressions. In Chapter 5 we exploit a similar technique, retrieving tweets from ironic/satirical accounts (in English, Spanish and Italian).

The dataset described in this section are either balanced (50% ironic and 50% non-ironic), or unbalanced (more non-ironic examples than ironic, to simulate real world applications). Regarding the unbalanced datasets [Abercrombie and Hovy, 2016] carried out experiments to study the evaluation issues of irony detection systems where the presence of ironic examples is narrow, in comparison with the non-ironic examples. They show that the use of unbalanced datasets (20% ironic and 80% non-ironic) led to large drops in F1 scores, due to this metric not taking into account true negatives. They state that the ratio of true negatives is necessarily large for effective sarcasm detection on data where positive examples are rare. For this reason they suggest to use AUC instead of F1 in unbalanced datasets, which, in their experiments, seems to be more consistent independently from the dataset (balanced or unbalanced).

2.1.2.3 Long Text

In this dissertation we only tackle the problem of irony detection in short text, however we briefly report here the long text datasets used in automatic detection of irony of on-line sources (like forum or Amazon reviews). [Tsur et al., 2010] collected 66,000 amazon reviews and manually annotated them for irony. Also [Filatova, 2012] create a dataset of Amazon reviews, manually annotating 437 ironic reviews and 817 regular reviews. [Lukin and Walker, 2013] collected a dataset of ironic online comments from forum posts (where the irony is manually flagged with a label). [Reyes and Rosso, 2012] download 3163 reviews from product where most of the reviews are ironic, or in general not serious. One of the products was the "Three Wolf Moon T-shirt", that become very popular because of the viral ironic/humorous reviews. [Liu et al., 2014] create a dataset of manually annotated documents from various sources including Amazon (5,491), Twitter (40,000), News Articles (4,233), and the Chinese Sina (3,859), Trecent (5,487) and Netease (10,356). The dataset included imbalanced classes,

[3] https://twitter.com/spinozait

the ironic examples were between 10% and 17% in each dataset. Finally, also data from Reddit has been exploited for irony detection: [Wallace et al., 2014] collect 10K comments labeled as ironic or non-ironic from 6 reddit subgroups: political news and editorials (r/politics), community for political conservatives (r/conservative), community for political progressives (r/progressive), community for non-believers (r/atheism), news and viewpoints on the Christian faith (r/Christianity), Technology news and commentary (r/technology).

2.1.3 Computational Models

We review in this section the models proposed to automatically recognize irony. We divide the systems in three parts: rule based systems, machine learning systems with handcrafted features, and deep-learning systems. In order to summarize the systems in a compact way we compiled Table 2.1 (short text) and Table 2.2 (long text). These two tables are modified versions of Table 1 of [Joshi et al., 2016a]. We show the type of annotation (manual, hashtag based, or other), the type of approach (rule based, feature extraction, or deep learning), the features used (described in the sub-section 2.1.3.2).

We will not report performances of these models, as the type of data (short-/long, noisy/not noisy) and the settings (balanced/unbalanced, different negative classes) make the comparison hard. One of the limitations in irony detection is indeed the lack of a standard datasets/tasks where to compare different systems [Joshi et al., 2016a]. We can say that the models we describe in the following sections were improving the state of the art at the moment of publication. Moreover, we can say that the feature engineering based system worked better than rule based ones, but worse that more recent deep leaning systems, that report the best scores in irony detection at the moment of writing.

[Wallace, 2015] suggest that all the theories of irony imply that the choice between literal and ironical interpretation of an utterance, must be based on contextual knowledge (i.e. external information of the utterance). However, in this dissertation we do not study contextual information, but only focus on intrinsic signs of irony. However, for completeness, we also report systems that use contextual additional information. Generally the contextual information added is about the conversational context, including tweet replies [Bamman and Smith, 2015, Joshi et al., 2015, Wang et al., 2015], and author information [Rajadesingan et al., 2015, Bamman and Smith, 2015, Amir et al., 2016].

In this section we will only mention irony detection, but we cover automatic detection of both irony and sarcasm. We use only the term irony as most of the papers we show, use the two terms interchangeably, and do not refer to the two phenomena as different ones.

Model	Annotat.	Type			Features					Context		
		Rule Based	Feat. Eng.	Deep Learn.	BoW	Sentiment	Pragmatic	Patterns	Other	Author	Convers.	Other
[Davidov et al., 2010]	M		x		x			x				
[González-Ibánez et al., 2011]	#		x		x	x	x					
[Reyes et al., 2012]	#		x		x	x	x		x			
[Riloff et al., 2013]	M	x			x		x	x				
[Liebrecht et al., 2013]	#		x		x	x		x		x		
[Reyes et al., 2013]	#		x		x	x	x		x			
[Reyes and Rosso, 2014]	#		x		x	x	x		x			
[Maynard and Greenwood, 2014]	M	x			x	x			x			
[Liu et al., 2014]	#		x		x	x	x		x			
[Joshi et al., 2015]	#			x	x	x	x	x				
[Khattri et al., 2015]	M	x			x	x				x		
[Khattri et al., 2015]	#		x		x	x			x	x	x	
[Bamman and Smith, 2015]	#		x		x	x	x		x	x	x	x
[Farıas et al., 2015]	#		x		x	x	x		x			
[Wang et al., 2015]	#		x		x						x	
[Bharti et al., 2015]	#	x			x	x		x				
[Fersini et al., 2015]	#		x		x		x		x			
[Frenda, 2016]	#	x			x		x	x				
[Muresan et al., 2016]	#		x		x	x	x					
[Abercrombie and Hovy, 2016]	M		x		x					x	x	x
[Joshi et al., 2016b]	#		x		x							
[Amir et al., 2016]	#			x						x		
[Ghosh and Veale, 2016]	M			x								
[Poria et al., 2016a]	M			x								

Table 2.1: List of models for the automatic detection of irony.

Model	Annotat.	Type			Features					Context		
		Rule Based	Feat. Eng.	Deep Learn.	BoW	Sentiment	Pragmatic	Patterns	Other	Author	Convers.	Other
[Tsur et al., 2010]	M		x		x			x				
[Reyes and Rosso, 2012]	M		x		x	x			x			
[Lukin and Walker, 2013]	M		x		x		x	x				
[Rakov and Rosenberg, 2013]	M		x		x				x			
[Buschmeier et al., 2014]	O		x		x	x	x			x		
[Wallace et al., 2015]	M		x		x	x			x		x	x
[Ghosh et al., 2015]	M		x		x				x			

Table 2.2: List of models for the automatic detection of irony.

2.1.3.1 Rule Based

This section is an overview of rule based systems proposed to recognize irony. In general such systems do not perform as well as statistical systems, however, it is interesting to see how researchers designed these systems starting from existing theories of irony. [Riloff et al., 2013] aim to recognize positive words in negative sentences, as they state that a common form of irony consists of a positive sentiment contrasted with a negative situation. They present a bootstrapping algorithm that automatically learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets, and show that this approach improve recall for irony recognition. [Maynard and Greenwood, 2014] design a number of rules to improve the accuracy of sentiment analysis when irony is known to be present. In particular they develop an ad-hoc hashtag tokenizer. They state that hashtags (without considering #irony/ #sarcasm) are often used to highlight irony, and if the sentiment expressed by hashtag is not the same as in the rest of the tweet, the tweet is predicted as sarcastic. Finally, [Frenda, 2016] propose a rule based system based on simple linguistic pattern rules, including positive interjections, the pattern “you”+ verb “to be”, disjunctive Conjunction (like “or”), quotation and exclamation marks, “onomatopoeic expressions for laughter” (“hahaha”) and dialectal expressions that suggest a colloquial register infrequent when talking about political issues. This rule based system outperformed many statistical systems participating to the SENTIPOLC irony detection task [Barbieri et al., 2016a], ranking second best system.

2.1.3.2 Feature Engineering

Most of the approaches to irony detection (including the one proposed in this dissertation) use a statistical approach composed of two phases. In the first phase each document is represented with a feature vector, where the features are calculated using various methods (we describe them later in this section). After the features calculation, a machine learning algorithm is used to classify ironic and non-ironic texts. The most used algorithms are Support Vector Machine [Cortes and Vapnik, 1995], Decision Tree [Salzberg, 1994], Random Forest [Liaw et al., 2002], and also Naive Bayes [John and Langley, 1995]. As these algorithm are well known in literature, we only focus on the description of the features.

In Table 2.1 and Table 2.2 is reported a summary of the features employed by the irony detection models. As we can see in both tables the most used feature is the bag of words (BoW, see Section 3.3.8). All the systems that exploit statistical learning methods use it as a base for their learning systems. Using words as features results convenient as some words or topic are often used with in ironic sentences (e.g. "I love *Mondays*", or '*Rain* is wonderful if you don't have an umbrella'). In Chapter 4 we show that using words as feature (BoW) has some limitation. Instead of learning to detect irony, word based models might learn to recognize the most ironic topics. For example, the term "Trump" can be a frequent term in ironic examples in Twitter, but this does not mean that this is a good feature to model irony as linguistic phenomena: using the term "Trump" as ironic feature work only in a limited domain.

One of the first attempts to use feature engineering and statistical classifiers to detect irony was proposed by [Tsur et al., 2010]. They first compile a set of ironic patterns (common combination of words present in the ironic examples), then extract features from the documents that they want to classify. The pattern-based features they use are real numbers, based on four rules: (1) the ironic pattern is in the document; (2) exact match but in the document there are additional words; (3) partial match with the pattern; (4) no match with any pattern. Then, in a second phase they use these features to build a classifier for ironic and non-ironic sentences.

[González-Ibánez et al., 2011] propose another model, composed of three pragmatic features: (1) positive emoticons such as smileys; (2) negative emoticons such as frowning faces; and (3) user mentions[4], which indicate that a message is addressed to someone.

The irony detection system proposed by [Reyes et al., 2012] include (1) Ambiguity: structural, morphosyntactic and semantic; (2) Polarity: words that denote either positive or negative semantic orientation; (3) Unexpectedness: contextual

[4] A user mention on Twitter is indicated with @ + username of the person that is being referred in the message

imbalances among the semantic meanings of the words; (4) Emotional scenarios: activation, imagery, and pleasantness scores measured with lexicons.

[Reyes et al., 2013] design another system, based on four dimensions: (1) Signatures: specific textual markers or signatures; (2) Unexpectedness: concerning verb temporal imbalance and contextual imbalance, by measuring the pair-wise semantic similarity of all terms in a text, measured with WordNet (feature also used in [Farıas et al., 2015]); (3) Style: captured by character-grams (c-grams), skip-grams (s-grams), and polarity skip-grams (ps-grams); and (4) Emotional scenarios: same feature proposed in [Reyes et al., 2012].

[Liu et al., 2014] use part of speech tag sequences, punctuation symbols, and semantic imbalance intended to capture inconsistencies within a context. They define the semantic imbalance measure as *the maximum semantic similarity scores*[5] *(across different senses of words in text) divided by the length of the text.*

[Joshi et al., 2015] refer to irony as a phenomena based on incongruity. They designing a system based on implicit and explicit sentiment incongruity. Explicit incongruity occurs when two words are in contrast like "love" (positive) and "ignored" (negative) in the sentence "I love being ignored". They use sentiment lexicons to measure this type of incongruity. The implicit incongruity, like in "I love this paper so much that I made a doggy bag out of it", and the contrast here is between a word (love that is positive) and the sentence "I made a doggy bag out of it" which has negative connotations. They use a sentiment classifier based on lexicons.

2.1.3.3 Deep Learning

Recently Deep Leaning approaches have been successfully applied to several Natural Language Tasks, text classification included. There are mainly three deep learning algorithms used in this context. The first one is to represent word as vectors using a Skipgram model [Mikolov et al., 2013c] or a glove model [Pennington et al., 2014], and using this vectors as features to recognizing irony. The work of [Joshi et al., 2016b] uses word embeddings, and it is somehow in the middle between deep leaning methods and feature engineering. They implement two kind of features based on word embeddings: (1) Similarity-based (maximum/minimum similarity score of most similar/dissimilar word pair respectively), and (2) Weighted similarity-based (maximum/minimum similarity scores of most similar/dissimilar word pairs divided by the linear distance between the two words in the sentence).

In the last few years other well know deep learning architectures have been used in irony detection. Convolutional Neural Networks (CNN) [LeCun et al.,

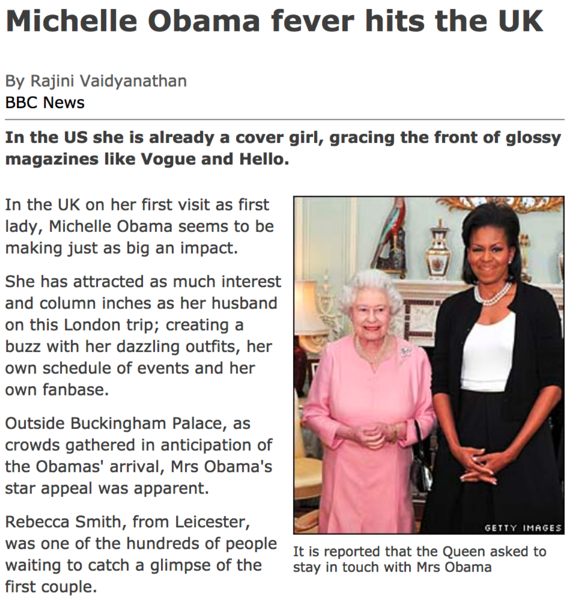
[5] It is not clear what kind of similarity they exploit.

1998] and Long Short Term Memory networls (LSTMs) [Hochreiter and Schmidhuber, 1997] are the two models used more often. Convolutional Neural Networks (CNN) [LeCun et al., 1998] are typically used in computer vision. Explaining how CNNs work is behind the scope of this thesis, but we can think of them as algorithms that process images that can be considered matrices of two or three dimensions (height, length, and pixel color if the image is not monochromatic). CNNs can be applied to text classification [Kim, 2014, Zhang et al., 2015] since given a sentence, we can represent it as a matrix of two dimensions, where each line is the vector of a word (learned for example with an unsupervised system [Mikolov et al., 2013c]) or the vector of the characters of the sentence [Zhang et al., 2015]. [Amir et al., 2016] and [Poria et al., 2016a] use Convolutional Neural Networks to classify irony. In particular, [Amir et al., 2016] use a standard binary classification, while [Poria et al., 2016a] use a combination of CNNs trained on different tasks, namely: sentiment analysis, emotion detection and personality features detection.

Another popular deep learning algorithm for text understanding is LSTM [Hochreiter and Schmidhuber, 1997], that we describe in Chapter 7. [Ghosh and Veale, 2016] propose to use a network based on two layers of LSTMs followed by a CNN. The system shows important improvements in the classification of irony.

2.2 Visual aspects of Social Media Content

With the advancement of communication technology, and large diffusion of smartphones with camera and the increasing number of social media users, a larger and larger amount of data is being uploaded in visual format (photos / videos) in addition to text format [Cambria et al., 2014]. Indeed, social networks such as Instagram[6] and Snapchat[7] heavily rely on multimodal contents. The visual content has becoming of central importance for social media communication, and focusing only on the text might be limiting. For this reason, we focus this thesis also on multimodal form of communication like emojis (small images that can be incorporated with traditional text messages), considering both textual and visual content of social media posts. In this section we overview some studies that explore language and vision (Section 2.2.1), and also we explore computational models for emoji semantics (Section 2.2.2)



Michelle Obama fever hits the UK

By Rajini Vaidyanathan
BBC News

In the US she is already a cover girl, gracing the front of glossy magazines like Vogue and Hello.

In the UK on her first visit as first lady, Michelle Obama seems to be making just as big an impact.

She has attracted as much interest and column inches as her husband on this London trip; creating a buzz with her dazzling outfits, her own schedule of events and her own fanbase.

Outside Buckingham Palace, as crowds gathered in anticipation of the Obamas' arrival, Mrs Obama's star appeal was apparent.

Rebecca Smith, from Leicester, was one of the hundreds of people waiting to catch a glimpse of the first couple.

GETTY IMAGES

It is reported that the Queen asked to stay in touch with Mrs Obama

Figure 2.1: Example of multimodal content of news articles used in [Feng and Lapata, 2010]

2.2.1 Vision and Language

The interplay between visual and textual contents has been studied in various tasks in the past, such as image captioning, visual question answering, multimodal semantic representations. The most explored area is probably the image captioning task, i.e. the description of images using natural language ([Bernardi et al., 2016] compiled an overview of recent datasets and systems about image captioning). Some of recent works are [Karpathy and Fei-Fei, 2015] and [Vinyals et al., 2015b] who propose similar models for image captioning: they use a RNN (LSTM) where the first "word" of the generative network is the vector of the image extracted using a Convolutional Neural Network (the LSTM is conditioned on the image information at the first time step). Many other models have been proposed to describe images, but in this review we focus on other aspects of vision and language.
Indeed, image captioning is not the only research area where visual and textual information are processed together. Various researchers proposed multimodal systems to improve semantic representation of words. [Feng and Lapata, 2010] propose an extension of Latent Dirichlet Allocation (LDA, [Blei et al., 2003, Blei and

[6] https://www.instagram.com/

[7] https://www.snap.com

Jordan, 2003]), a topic model algorithm that can be used to represent meaning as a probability distribution over a set of multimodal topics (a set of BBC news articles and related pictures is used as dataset, an example is shown in Figure 2.1). To extract visual information from pictures, they employ the Scale Invariant Feature Transform (SIFT) algorithm [Lowe, 1999] over segments of the pictures (143 parts). [Bruni et al., 2014] explore the use of visual information in distributional semantics, extending the representation of a word with its co-occurrence with the visual words automatically extracted from images (Bag of visual words, [Sivic and Zisserman, 2003, Csurka et al., 2004, Nister and Stewenius, 2006]). [Kiela and Bottou, 2014] construct multimodal concept representations by simply concatenating a word embedding representation [Mikolov et al., 2013c] with a visual concept representation vector computed using a Convolutional Neural Network (they use Alexnet [Krizhevsky et al., 2012]) pre-trained on a large labeled object recognition dataset (Imagenet [Deng et al., 2009a, Russakovsky et al., 2015b]). They also use this method to automatically recognize metaphors [Shutova et al., 2016]. [Lazaridou et al., 2015] propose another way to learn multimodal embeddings. They extend the skip-gram model for word embeddings [Mikolov et al., 2013c] by taking visual information into account: their multimodal model builds vector representation for words by learning to predict linguistic contexts in text corpora, and, for a restricted set of words, the model is also exposed to visual representations of the objects they describe (same CNN as the work described above [Krizhevsky et al., 2012]) and predicts linguistic and visual features jointly. This way they improve the semantic representation quality of these words.

Another area that mixes vision and textual information is multimodal sentiment analysis [Morency et al., 2011, Mihalcea, 2012, Maynard et al., 2013]. Given a multimodal post (image and text, or video and text) the task consists in predicting the sentiment of the post (in general positive or negative). The multimodal systems have to model both textual and visual contents to extract the necessary knowledge to make the predictions. Recent systems use deep learning state of the art algorithms (like CNNs and LSTMs) to tackle this task [Poria et al., 2015, Poria et al., 2016b]. In Section 7.5 we show experiments on a similar line, but instead of learning sentiments we learn to predict emojis, that can be considered a more fine grained representation of sentiments and emotions.

2.2.2 Emojis

Emojis are ideograms used in online messages that were initially used by Japanese mobile operators, NTT DoCoMo and SoftBank Mobile that defined their own emojis. The first emoji was created in 1999 by Shigetaka Kurita[8] a Japanese

[8] http://time.com/4114886/oxford-word-of-the-year-2015-emoji/

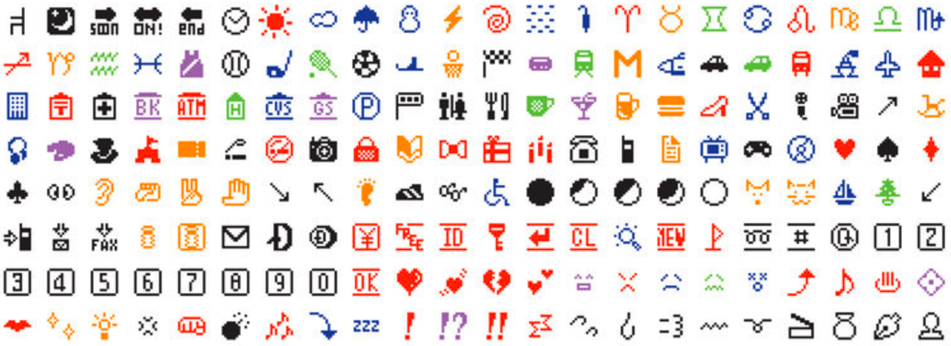


Figure 2.2: Set of the first 176 emojis, exposed at the MoMa museum in New York.

employee of NTT DoCoMo's i-mode mobile Internet platform. At the moment there are 2,623 emoji variants, designed by the Unicode consortium[9]. The first set of emojis was composed by 176 elements (Figure 2.3), and they are now exposed at the MoMa museum in New York. Currently, emojis represent a widespread and pervasive global communication device largely adopted by almost any Social Media service and instant messaging platform [Jibril and Abdullah, 2013, Park et al., 2013, Park et al., 2014]. Emojis, like the older emoticons, support the possibility to express diverse types of contents in a visual, concise and appealing way that is perfectly suited to the informal style of Social Media communication.

2.2.2.1 Emoticons

The meaning expressed by emoticons has been exploited to enable or improve several tasks related to the automated analysis of Social Media contents, like sentiment analysis [Hogenboom et al., 2015, Hogenboom et al., 2013]. In this context, emoticons have also been often exploited to label and thus characterize the textual excerpts where they occur. As a consequence, by analyzing all the textual contents where a specific emoticon appears several sentiment and emotional lexicons have been build [Yang et al., 2007, Tang et al., 2014, Boia et al., 2013]. [Go et al., 2009] and [Castellucci et al., 2015] use distant supervision over emotion-labelled textual contents in order to respectively train a sentiment classifier and build a polarity lexicon. [Aoki and Uchida, 2011] described a methodology to represent each emoticon as a vector of emotions and [Jiang et al., 2015] proposed a sentiment and emotion classifier based on semantic spaces of emojis in the Chinese Website Sina Weibo. [Pavalanathan and Eisenstein, 2015] used a matching

[9] Full list can be found at http://www.unicode.org/emoji/charts/full-emoji-list.html

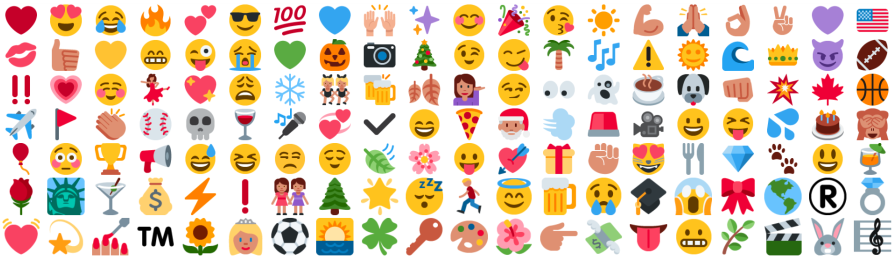


Figure 2.3: The 140 most frequent new emojis in Twitter in 2016 in United States (in the dataset that we retrieved, described in Chapter 6).

approach from causal inference to test whether the adoption of emojis causes individual users to employ fewer emoticons in their text on Twitter. They show that happy and playful emoticons, such as :-) and :P, have higher rate of decrease of use than sad emoticons such as :(.

2.2.2.2 Emoji Sentiments

[Novak et al., 2015] built a manually annotated lexicon and drew a sentiment map of the 751 most frequently used emojis. 83 human annotators labeled over 1.6 million tweets in 13 European languages by the sentiment polarity (negative, neutral, or positive). About 4% of the annotated tweets included emojis. Analyzing the sentiment of the emojis, they found out that most of the emojis are positive, especially the most popular ones, and that the sentiment distribution of the tweets with and without emojis is significantly different.

[Wood and Ruder, 2016] collect 588,607 of multilingual tweets containing emotion-specific emoji and assess selected emoji as emotion labels, utilizing human annotators as the ground truth. They found high correspondence between emoji and emotion annotations, indicating the presence of emotion indicators in tweet texts alongside the emoji. They suggest that emojis may be useful as distant emotion labels for statistical models of emotion in text instead of hashtags. They state so as emojis are more popular than hashtags, and emojis present a more faithful representation of a user's emotional state.

2.2.2.3 Emoji Semantics

[Miller et al., 2016] explored whether emoji renderings or differences across platforms (e.g. Apple's iPhone vs. Google's Nexus phone) give rise to diverse interpretations. Moreover, they perform a human evaluation on the meaning and use of emojis and find out that emojis are not interpreted in the same way. They also extended this study [Miller et al., 2017] and add that when emojis are interpreted in textual contexts, the potential for miscommunication appears to be roughly the same than without text.

[Eisner et al., 2016] learn the meaning of emojis by their description, improving our work on emojis semantics [Barbieri et al., 2016b], where we explored meaning of Twitter emojis in American English with Distributional Semantics (Section 6).

[Ai et al., 2017] study the Emoji Popularity through semantic embeddings [Mikolov et al., 2013c]. They find that (1) emojis with clear semantic meanings are more likely to be popular; (2) entity-related emojis are more likely to be used as alternatives to words; and (3) sentiment-related emojis play a complementary role in a message.

Chapter 3

#IRONY AND #SARCASM DETECTION IN SOCIAL MEDIA

Automatic detection of figurative language is a challenging task in computational linguistics. Recognizing both literal and figurative meaning is not trivial for a machine and, in some cases, it is hard even for humans. In this chapter we present our approach to detect irony and sarcasm in Twitter. The content of this chapter was published in the following papers [Barbieri and Saggion, 2014a, Barbieri and Saggion, 2014b, Barbieri et al., 2014b].

3.1 Introduction

The main idea behind our irony detection model is unexpectedness [Lucariello, 1994], a key factor for irony. We described this idea in the introduction of this dissertation, but to recall it we can say that ironic authors use unexpected terms in an unexpected context to surprise the listener and make him think that the sentence is not serious. We build a rich set of features to model unexpectedness and surprise in social media texts. We run experiments to detect irony and sarcasm in Twitter and show that our approach is superior to state of the art. In order to evaluate the robustness of our model to topic changes, we carry on cross-domain experiments. Our system shows acceptable performances also when the training and testing domains are different, while word-based approaches (Bag of Words) tend to model a specific topic instead of capturing a linguistic phenomenon such as irony. The domain adaptation problem is also addressed in detail in Chapter 4 in the context of irony detection and sentiment analysis in Italian.

3.2 Dataset and Text Processing

Dataset

We use a corpus of 60,000 tweets equally divided into six different topics: *Irony*, *Sarcasm*, *Education*, *Humour*, *Politics* and *Newspaper*. The Newspaper set includes 10,000 tweets from three popular newspapers (The Economist, New York Times and The Guardian). The rest of the tweets (50,000) were automatically selected by looking at Twitter hashtags #education, #humour, #irony, #politics and #sarcasm) added by users in order to link their contribution to a particular subject and community. These hashtags are removed from the tweets for the experiments. According to [Reyes et al., 2013], selecting tweets using the distant supervision of hashtags, is convenient for three reasons: (i) to avoid manual selection of tweets, (ii) to allow irony analysis beyond literary uses, and because (iii) irony hashtag may "reflect a tacit belief about what constitutes irony" (and sarcasm in the case of the hashtag #sarcasm). *Irony*, *Education*, *Humour* and *Politics* tweets were prepared by [Reyes et al., 2013], we added *Newspaper* and *Sarcasm* tweets obtaining them trough the Twitter APIs[1].

In Table 3.1 are reported two examples for each category of our dataset.

Another dataset employed was the American National Corpus dataset[2]. We adopted the second release of the American National Corpus Frequency Data [Ide and Suderman, 2004], which provides the number of occurrences of a word in the written and spoken American National Corpus.

Text Processing

In order to pre-process the tweets and extract syntactic features we decided to use a toolkit specifically designed for Twitter language. The toolkit we employed was the GATE Plugin TwitIE [Bontcheva et al., 2013]. TwitIE is a pipeline of three text processes: (i) a tokenizer for Twitter language (words with hashtag and links are correctly tokenized) (ii) a name entity recognition system that uses gazetteers vocabularies (iii) a Part of Speech tagger, an adapted version of the Stanford tagger [Toutanova et al., 2003] trained on Twitter data that achieves 90.54% token accuracy on the POS tagging task of short text.

In our experiments we modify TwitIE extending the gazetteers vocabulary and including new empirical rules for the text normalization (for instance, *looool* is normalized to *lol*).

[1] https://dev.twitter.com

[2] http://www.anc.org/

I'm so consistent when it comes to misspelling the word inconsistency. **#irony**
You're right. Passive aggressiveness is probably the best way to handle any situation. **#irony**
First run in almost two months. I think I did really well. **#sarcasm**
Jeez I just love when I'm trying to eat lunch and someone's blowing smoke in my face. Yum. I love ingesting cigarette smoke. **#sarcasm**
A skeleton walks into a bar and says: give me a beer....and a mop :D #jokes **#humor**
If love is blind, why is lingerie so popular? **#humor**
Can India ascend into a leader in the open knowledge economy? **#education**
Do violent games turn us into killers? **#education**
What political figures in recent history do you admire most? **#politics**
Solving The Unemployment Crisis, Republican Style **#politics**
No-makeup selfies raise 8m for Cancer Research UK in six days **(from @guardian)**
Millennials at work are lazy and callow. But then again, so were their parents, and their parents, and their parents **(from CNN)**

Table 3.1: Examples from the irony and sarcasm detection dataset.

At the moment of publications also another system oriented to Twitter language was available, a POS tagger and parser trained on tweets that obtain good results on the POS and parsing tasks in the context of short text messages [Gimpel et al., 2011, Owoputi et al., 2013].

3.3 Model

We approach the detection of irony as a classification problem applying supervised machine learning methods to the Twitter corpus. Each tweet is represented as a vector of feature/values obtained from our analysis of the text of the tweet. The vectors are used to train a machine learning algorithm. In our case, to make the work comparable with the state of the art, we carried out the experiments with the classification algorithm Decision Tree [Reyes et al., 2013].

The features we extract from each tweet can be categorized in seven groups. Some of them are designed to detect imbalance and unexpectedness, others to detect common patterns in the structure of the ironic tweets (like type of punctuation, length, emoticons), and some others to recognize sentiments and intensity of the terms used. Below is an overview of the group of features in our model:

- Frequency *(gap between rare and common words)*
- Written-Spoken *(written-spoken style uses)*
- Intensity *(intensity of adverbs and adjectives)*
- Sentiments *(gap between positive and negative terms)*
- Synonyms *(common vs. rare synonyms use)*
- Ambiguity *(measure of possible ambiguities)*
- Structure *(length, punctuation, emoticons)*

To the best of our knowledge Frequency, Written Spoken, Intensity, Synonyms and Ambiguity groups were novel when we published this approach [Barbieri and Saggion, 2014a]. The other groups (Sentiments and Structure) were employed by other computational models (e.g. [Carvalho et al., 2009, Reyes et al., 2013]).

In the following sections we describe in detail all the features the model is composed of. Note that when describing the features we use the term irony, but the same model is also used for sarcasm detection.

3.3.1 Frequency

In this first group of features we try to model whether a tweet contains surprising elements in terms of register used by the author of the tweet. We do this by exploring the frequency imbalance between the words employed, i.e. register inconsistencies between terms of the same tweet. The idea is to detect the presence of uncommon words (i.e. low frequency in ANC) in a text which contains mainly a common vocabulary (i.e. high frequency in ANC). Three features belong to this group: **frequency mean**, **rarest word**, **frequency gap**. The first one is the arithmetic average of all the frequencies of the words in a tweet, and it is used to detect the *frequency style* of a tweet. The second one, **rarest word**, is the frequency value of the rarest word, designed to capture the word that may create imbalance. The assumption is that very rare words may be a sign of irony. The third one is the absolute difference between the first two and it is used to measure the imbalance between them, and capture a possible intention of surprise.

3.3.2 Written-Spoken

These features are designed to model the register inconsistency, introduced in the previous section, in a spoken and written context. One tweet includes only one style, either written style or spoken style, but mixing together these two styles might be signal of irony.

We explore the unexpectedness created by using spoken style words in a mainly written style tweet, and vice-versa (formal words, usually adopted in written text, employed in a spoken style context). We can model this idea thanks to the ANC written and spoken corpora, that provides usage frequency of a word in written and spoken English.

There are three features in this group: **written mean**, **spoken mean**, **written spoken gap**. The first and second ones are the means of the frequency values, respectively, in written and spoken ANC corpora of all the words in the tweet. The third one, **written spoken gap**, is the absolute value of the difference between the first two, designed to see if ironic tweets include both styles (creating imbalance) or only one of them. A low difference between written and spoken styles means that both styles are used.

3.3.3 Intensity

In order to produce a ironic effect some authors might use an expression which is antonym to what they are trying to describe (*saying the opposite of what they mean*). In the case the word being an adjective or adverb its intensity, more or less exaggerated, may well play a role in producing the intended effect [Riloff et al., 2013]. We adopted the intensity scores of [Potts, 2011] who uses naturally occurring metadata (star ratings on service and product reviews) to construct adjectives and adverbs intensity scores. An example of adjective scale (and relative scores in brackets) could be the following: horrible (-1.9) → bad (-1.1) → good (0.2) → nice (0.3) → great (0.8).

With these scores we evaluate four features for adjective intensity and four for adverb intensity (implemented in the same way) The sum of the AdjScale scores of all the adjectives in the tweet (**adj tot**), the same sum divided by the number of adjectives in the tweet (**adj mean**), the maximum AdjScale score within a single tweet (**adj max**), and finally, **adj gap** that is the difference between **adj max** and **adj mean**, designed to see "how much" the most intense adjective is out of context.

3.3.4 Synonyms

As previously said, irony conveys two messages to the audience at the same time. It follows that the choice of a term (rather than one of its synonyms) is very important in order to send the second, not obvious, message.

For each word of a tweet we get its synonyms with WordNet [Miller, 1995], then we calculate their ANC frequencies and sort them into a decreasing ranked list (the actual word is part of this ranking as well). We use these rankings to define the four features which belong to this group. Given the word w_i, **syno lower** is defined as follow:

$$sl_{w_i} = |syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)| \tag{3.1}$$

where $syn_{i,k}$ is the synonym of w_i with rank k, and $f(x)$ the ANC frequency of x (so this feature is the number of synonyms of the word w_i with frequency lower than the frequency of w_i). Then we also defined **syno lower mean** as mean of sl_{w_i} (i.e. the arithmetic average of sl_{w_i} over all the words of a tweet).

We also designed two more features: **syno lower gap** and **syno greater gap**, but to define them we need two more parameters. The first one is *word lowest syno* that is the maximum sl_{w_i} in a tweet. It is formally defined as:

$$wls_t = \max_{w_i}\{syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)\} \tag{3.2}$$

The second one is *word greatest syno* defined as:

$$wgs_t = \max_{w_i}\{syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)\} \tag{3.3}$$

We are now able to describe **syno lower gap** which detects the imbalance that creates a common synonym in a context of rare synonyms. It is the difference between *word lowest syno* and **syno lower mean**. Finally, we detect the gap of very rare synonyms in a context of common ones with **syno greater gap**. It is the difference between *word greatest syno* and *syno greater mean*, where *syno greater mean* is the following:

$$sgm_t = \frac{|syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)|}{n.\, words\, of\, t} \tag{3.4}$$

The arithmetic averages of **syno greater gap** and of **syno lower gap** in the Sarcasm corpus are higher than in the other topics, suggesting that a very common (or very rare) synonym is often used out of context i.e. a very rare synonym when most of the words are common (have a high rank in our model) and vice versa.

3.3.5 Ambiguity

Another interesting aspect of irony is ambiguity. We noticed that ironic tweets presents words with more meanings (average number of WordNet synsets is higher in ironic tweets than in the other topics). Our assumption is that if a word has many meanings it is more likely to be used in an ambiguous way.

There are three features that aim to capture these aspects: **synset mean**, **synset max**, and **synset gap**. The first one is the mean of the number of synsets of each word of the tweet, to see if words with many meanings are often used in the tweet. The second one is the greatest number of synsets that a single word has; we consider this word the one with the highest possibility of being used ironically (as multiple meanings are available to say different things). In addition, we calculate **synset gap** as the difference between the number of synsets of the word with more synonyms in the tweet (**synset max**) and the average number of synsets (**synset mean**), assuming that if this gap is high the author may have used that inconsistent word intentionally.

3.3.6 Sentiments

We also evaluate the sentiment of the ironic tweets. The SentiWordNet sentiment lexicon [Baccianella et al., 2010] assigns to each synset of WordNet sentiment scores of positivity and negativity. We use these scores to examine what kind of sentiments characterize irony. We explore the sentiment of the tweet according to information from a sentiment lexicon with two different views: the first one being the simple analysis of sentiments (to identify the main sentiment of a tweet) and the second one concerns sentiment imbalances between words.

There are six features in the **Sentiments** group. The first one, **positive sum**, is the sum of all the positive scores in a tweet, the second one is **negative sum**, defined as sum of all the negative scores. The arithmetic average of the previous ones is another feature, named **positive negative mean**, designed to reveal the sentiment that better describe the whole tweet. Moreover, there is **positive-negative gap** that is the difference between the first two features, as we wanted also to detect the positive/negative imbalance within the same tweet.

The imbalance may be created using only one single very positive (or negative) word in the tweet, and the previous features will not be able to detect it, thus we needed to add two more. For this purpose the model includes **positive single gap** defined as the difference between most positive word and the mean of all the sentiment scores of all the words of the tweet and **negative single gap** defined in the same way, but with the most negative one.

3.3.7 Structure

With this group of features we want to study the structure of the tweet. These group is composed of three subgroup of features: Characters, Name Entities, and Part of Speech.

The Character subgroup consists in simple shallow features related to the characters included in the tweets. The **length** feature consists of the number of characters that compose the tweet, **n. words** is the number of words, and **words length mean** is the mean of the words length. The **punctuation** feature is the sum of the number of **commas**, **full stops**, **ellipsis**, **exclamation** and **quotation** marks that a tweet contain. The **emoticon** feature is the sum of the emoticons *:)*, *:D*, *:(* and *;)* in a tweet.

We include name entity features provided by TwitIE [Bontcheva et al., 2013]. These features are the count of the following entities in the tweet: *n. organization*, *n. location*, *n. person*, *n. first person*, *n. title*, *n job title*, *n. date*. Some of this features work very well when distinguishing sarcasm from newspaper tweets. We call these subgrup Name Entity (NE)[3].

Finally, we add the subgroup of Part of Speech (POS) features, that consist in the number of verbs, nouns, adjectives and adverbs as features (**n. verbs**, **n. nouns**, **n. adjectives** and **n. adverbs**). With these last four features we also computed the ratio of each part of speech to the number of words in the tweet; we called them **verb ratio**, **noun ratio**, **adjective ratio**, and **adverb ratio**. All these features have the purpose of capturing the style the structure of the tweet.

3.3.8 Bag of Words Baseline

We designed a Bag-of-Words (BoW) classifier as such model has been successfully employed in several irony detection task tasks (see Section 2.1.3). We represent each message with a vector of the 2,000 most informative tokens (punctuation marks are included as well). Words are selected using term frequency-inverse document frequency (TF-IDF), which is intended to reflect how important a word is to a document (tweet) in the corpus. After obtaining a vector for each message we classify with the Decision Tree algorithm.

3.4 Experiments and Results

We perform two types of experiments to assess the model in the task of discriminating ironic and non-ironic messages. For the first experiment we followed the same experimental settings of [Reyes et al., 2013], so to be able to compare with current state of the art. Our second experiments seeks to understand if the model is robust to topic changes.

3.4.1 First Experiment

In the first experiment we compare our approach with state of the art approach from [Reyes et al., 2013] at the time our research was carried out. In order to make a valid comparison, we use the same dataset and the same classification algorithm employed in that paper. The experiment consist in creating three datasets,

[3] These last seven features were not available in [Barbieri and Saggion, 2014a]

each containing 10,000 ironic tweets and 10,000 non-ironic tweets. The negative classes are Education, Politics and Humor:

- Irony vs Education
- Irony vs Politics
- Irony vs Humor

To train and test the systems, we run in each of the three datasets a 10-fold cross-validation classification experiments. The results of this experiment are reported in Table 3.2. We can see that our model outperforms in all the tree datasets the Reyes' model [Reyes et al., 2013].

3.4.2 Second Experiment

The second experiment is designed to test the robustness to topic variations of our approach. As baseline we use the Bag of Words described in Section3.3.8. We were also interested in verifying if the model could be used to identify sarcastic tweets, so we also report here experiments using a corpus of sarcastic tweets (retrieved with the #sarcasm). We also add Newspapers to the non-ironic domains. The datasets employed in this experiments are composed as before by 10,000 ironic (or sarcastic) tweets, and 10,000 non-ironic (or non-sarcastic) tweets. Ten datasets are obtained, five for irony and five for sarcasm:

- Irony vs Education | Politics | Humor | News | Sarcasm
- Sarcasm vs Education | Politics | Humor | News | Irony

Each dataset is divided in training set (80%) and testing set (20%). The systems are trained on one dataset and tested in all the other ones (keeping the positive label the same). For example, a system is trained on the training set of Irony vs Education and tested on the test set of Irony vs Education, Irony vs Politics, Irony vs Humor, Irony vs News, and Irony vs Sarcasm. The same setting is applied when sarcasm is the positive label. This experimental setup was followed to assess the classification accuracy of the models when the testing topics where not the same as the test topics. The results of irony detection are shown in Table 3.3 and the results for sarcasm detection are reported in Table 3.4. These tables include results for all the combinations of training/testing. We can see that in both cases our system outperform the bag of words baseline, especially when the training and testing topics are different. Precision, recall and F-measure of the majority

	Education			**Humour**			**Politics**		
Model	P	R	F1	P	R	F1	P	R	F1
Reyes et. al	.76	.66	.70	.78	.74	.76	.75	.71	.73
Our model	**.86**	**.86**	**.86**	**.88**	**.88**	**.88**	**.85**	**.85**	**.85**

Table 3.2: Precision, Recall, and F-Measure over the three corpora Education, Humour, and Politics. Reyes et al. and our results are shown; the classifier used is Decision Tree for both models, and also the dataset is the same. We marked in **bold** the results that are better compared to the other model.

	Train/Test	**Edu**	**Hum**	**Pol**	**New**	**Sar**
BOW model	**Edu**	0.88	0.82	0.80	0.57	0.42
	Hum	0.78	0.91	0.77	0.49	0.38
	Pol	0.79	0.78	0.91	0.59	0.37
	New	0.56	0.41	0.56	0.82	0.39
	Sar	0.53	0.56	0.51	0.45	0.69
Our model	**Edu**	0.92	0.85	0.87	0.65	0.49
	Hum	0.86	0.93	0.82	0.6	0.49
	Pol	0.89	0.84	0.91	0.67	0.49
	New	0.75	0.59	0.74	0.91	0.44
	Sar	0.63	0.67	0.63	0.54	0.76

Table 3.3: F-Measure over the various corpora train/test corpora combinations for irony detection. In the rows we list the corpora used for training and in the columns the corpora used for testing. The classifier used is Decision Tree for both models (BOW and OUR).

baseline (predicting always as one of the two classes) in these experiments are respectively 0.25, 0.50, and 0.33. Hence, neither of the two models performs worse than the majority baseline.

In order to have a clear understanding about the contribution of each set of features in our model, we also studied the of information gain in each training dataset. We compute information gain experiments over the ten balanced corpora and present the results in Table 3.5 (irony detection) and Table 3.6 (sarcasm detection). The information gain on the task irony vs sarcasm is treated separately in Table 3.6, where we also report the mean values of each feature, and the difference between them (normalized by the sum).

	Train/Test	**Edu**	**Hum**	**Pol**	**New**	**Iro**
BOW model	**Edu**	0.88	0.84	0.82	0.63	0.45
	Hum	0.78	0.91	0.79	0.54	0.44
	Pol	0.83	0.84	0.94	0.64	0.44
	New	0.7	0.53	0.69	0.88	0.54
	Iro	0.65	0.62	0.67	0.72	0.69
Our model	**Edu**	0.96	0.86	0.89	0.74	0.51
	Hum	0.89	0.96	0.86	0.69	0.50
	Pol	0.89	0.84	0.97	0.78	0.52
	New	0.78	0.65	0.78	0.97	0.53
	Iro	0.75	0.73	0.78	0.81	0.76

Table 3.4: F-Measure over the various corpora train/test corpora combinations for sarcasm detection. In the rows we list the corpora used for training and in the columns the corpora used for testing. The classifier used is Decision Tree for both models (Bow and Our).

Education	**Humor**	**Politics**	**Newspaper**
[C]question,0.3	**[C]**question,0.17	**[C]**question,0.21	**[P]**fullstop,0.17
[P]noun-ratio,0.09	**[C]**emoticons,0.08	**[P]**noun-ratio,0.12	**[C]**punctuation,0.12
[SY]syno-low,0.08	**[C]**smile,0.07	**[SY]**syno-low,0.06	**[C]**w-avg-length,0.11
[P]fullstop,0.06	**[F]**rarest,0.07	**[F]**freq-avg,0.05	**[C]**length,0.11
[P]adj,0.05	**[SY]**syno-gr-gap,0.07	**[E]**organization,0.05	**[P]**noun,0.07
[C]length,0.05	**[I]**adv-max,0.05	**[P]**verbs,0.05	**[NE]**locations,0.07
[P]punctuation,0.05	**[SE]**pos-sum,0.05	**[WS]**written-avg,0.05	**[C]**number-words,0.06
[I]adj-max,0.05	**[WS]**wr-sp-gap,0.04	**[WS]**wr-sp-gap,0.04	**[I]**adj-max,0.06
[P]adj,0.04	**[I]**adv-max,0.04	**[WS]**spoken-avg,0.04	**[P]**adj,0.05
[I]adj-avg,0.04	**[WS]**written-avg,0.04	**[C]**w-avg-length,0.04	**[WS]**spoken-avg,0.05
[C]w-avg-length,0.04	**[I]**adv-avg,0.04	**[C]**length,0.04	**[C]**exlamation,0.05
[P]adj-ratio,0.04	**[F]**freq-avg,0.04	**[P]**adj,0.04	**[P]**adj-ratio,0.05
[C]number-words,0.04	**[P]**adj,0.03	**[P]**verb-ratio,0.04	**[P]**verbs,0.04
[E]organization,0.04	**[I]**adj-max,0.03	**[I]**adj-max,0.04	**[P]**adj,0.04
[P]verbs,0.04	**[SY]**syno-low,0.03	**[P]**fullstop,0.04	**[SY]**syno-low,0.04

Table 3.5: Best 15 features of our model ranked considering the information gain scores in the **irony** vs all other datasets. In **[bold]** are reported the group of each feature. A=Ambiguity, C=Characters, F=Frequency, I=Intensity P=POS, SE=Sentiments, SY=Synonyms, WS=written spoken.

Education	Humor	Politics	Newspaper
[C]question,0.27	**[C]**question,0.15	**[P]**noun-ratio,0.2	**[P]**noun-ratio,0.24
[P]noun-ratio,0.17	**[F]**rarest,0.12	**[C]**question,0.19	**[P]**noun,0.19
[P]noun,0.1	**[I]**adv-max,0.08	**[P]**noun,0.11	**[C]**w-avg-length,0.19
[SY]syno-low,0.09	**[I]**adv-avg,0.08	**[I]**adv-max,0.09	**[P]**fullstop,0.17
[C]w-avg-length,0.08	**[SY]**syno-gr-gap,0.08	**[F]**rarest,0.09	**[P]**punctuation,0.15
[I]adj-max,0.07	**[I]**adv-max,0.07	**[C]**w-avg-length,0.09	**[C]**punctuation,0.15
[E]organization,0.07	**[C]**emoticons,0.07	**[E]**organization,0.08	**[I]**adv-max,0.13
[P]adj-ratio,0.07	**[P]**noun-ratio,0.06	**[SY]**syno-low,0.08	**[F]**rarest,0.11
[P]adj,0.07	**[C]**smile,0.06	**[I]**adv-max,0.07	**[C]**exlamation,0.1
[I]adv-max,0.06	**[P]**adj-ratio,0.05	**[SE]**pos-sum,0.07	**[I]**adj-max,0.1
[P]punctuation,0.06	**[SE]**pos-sum,0.05	**[P]**adj,0.06	**[NE]**locations,0.1
[I]adj-avg,0.06	**[WS]**written-avg,0.05	**[I]**adj-max,0.06	**[C]**length,0.1
[P]adj,0.06	**[WS]**wr-sp-gap,0.05	**[SE]**pos-gap,0.05	**[SY]**syno-low,0.09
[P]fullstop,0.05	**[F]**freq-avg,0.05	**[C]**length,0.05	**[P]**adj,0.09
[SY]syno-gr-gap,0.05	**[P]**noun,0.04	**[P]**adj-ratio,0.05	**[SE]**pos-sum,0.08

Table 3.6: Best 15 features of our model ranked considering the information gain scores in the **sarcasm** vs all the other datasets. In **[bold]** are reported the group of each feature. A=Ambiguity, C=Characters, F=Frequency, I=Intensity P=POS, SE=Sentiments, SY=Synonyms, WS=written spoken.

3.5 Discussion

In Table 3.2 we can see that our model outperforms with an important margin the [Reyes et al., 2013] model, suggesting that the features we propose are able to model irony with good accuracy. Moreover, our features are supposed to not depend on specific topics as we avoid word based features like in Bag of Words approaches. Indeed, in the second experiment we can see that our model is able to recognize irony with good accuracy also when the topic changes. The bag of words presents problems in detecting several topic when training and testing topics are different. For instance, in the Irony detection task (Table3.3) BOW performs poorly when trained on Humour and tested in Newspapers (0.49) and vice-versa (0.41), while when trained on Education and tested on politics (and vice-versa) the results are better (0.8 and 0.79). The drop in performances is similar in our model, suggesting that Education and Politics are the most close topics, and Humor and Newspaper the most different ones. The most problematic topic in irony detection is sarcasm. Indeed when Bow and our models are trained or tested on sarcasm, they obtain the worst results. Also when the topic does not change (training and testing on irony vs sarcasm), the results are worse than in all the other topics (0.76). This is probably due to the similarity between these two phenomena.

Feature	**IG**	**Irony**	**Sarcasm**	**Diff**
[I]adv-max	0.05	0.39	0.55	-0.17
[P]noun	0.03	3.38	2.53	0.14
[SE]senti-total	0.03	0.01	0.15	-0.88
[C]length	0.03	72.7	62.4	0.08
[F]rarest	0.03	0.85	0.13	0.74
[SE]pos-sum	0.03	0.24	0.35	-0.19
[I]adv-max	0.03	0.47	0.66	-0.17
[SE]pos-gap	0.02	0.16	0.22	-0.16
[P]noun-ratio	0.02	0.34	0.29	0.08
[I]adj-max	0.02	0.26	0.28	-0.04
[I]adv-avg	0.02	0.32	0.46	-0.18
[C]number-words	0.01	10.2	8.8	0.07
[C]exlamation	0.01	0.17	0.38	-0.38
[I]adj-avg	0.01	0.23	0.22	0.02
[SY]syno-low	0.01	78.1	68.7	0.06

Table 3.7: Best 15 features of our model ranked considering the information gain scores in the **irony** vs **sarcasm** task. In addition to the information gain, we report the mean of each feature over the irony and sarcasm datasets, and also the difference (normalized by the sum) between these values.

In the sarcasm detection (Table3.4) the results are higher than in the irony detection, suggesting that sarcasm might be slightly simpler to recognize. However, when training or testing on irony the results are poor.

These results suggest that our system is able to recognize irony and sarcasm, but struggles to distinguish these two, suggesting that these two linguistic devices are very similar.

Analyzing the data on Table 3.5, we observe that features which are more discriminative of ironic style over the different negative topics are the structural features (like Characters and POS). In particular, **questions** works quite well, as ironic topic include fewer questions than the rest of topics. Also the the **noun-ratio** play an important rule in irony detection, especially Education and Politics (that include more nouns than irony). The choice of the synonym is important too, **synonym lower** has a high information gain in Humor and Politics. Also the intensity of the adjective used is an informative feature, as in Ironic tweets the intensity of the adjective used is higher than in the other topics. Note, that there is a topic or theme effect since features behave differently depending on the dataset used: the Newspaper corpus seems to be the least consistent. For instance **punctuation** and features related to the length seem to be important only in this domain. It is also interesting to see that the Written-Spoken features are relevant only in irony vs Politics and Newspaper, as these last two domains do not include slang or words used often in spoken language.

The relevant features for the sarcasm detection experiments are similar than the ones of irony detection. Important features are **questions** and **noun-ratio** since question marks and nouns are fewer in sarcastic tweets. Intensity play an important role too, as sarcasm present more intense adverbs and adjectives than the other topics. Also in this case, the character features and length related features are very important in Newspapers.

Finally, we focus on the most problematic task, that is irony vs sarcasm. We have previously seen that the our system accuracy drops when trying to distinguishing these two hashtags. This might be due to the fact that people use it interchangeably, or that our model lacks important features for this task. In Table 3.7 we report information gain scores of the best 15 features on this task, and also the mean values of these features in the ironic and sarcastic corpus. The best feature to say if a tweet is ironic or sarcastic is the intensity of the adverbs: sarcastic tweets includes more intense adverbs, but less nouns than ironic tweets. Moreover, as expected, the sentiment of the tweet helps, as sarcastic tweets are more positive than ironic tweets.

3.6 Conclusions

In this chapter, we describe the problem of irony detection in social media and introduce our approach to automatically recognize irony. Irony detection was tackled as a binary classification problem, where, given a tweet, the task is to recognize if the tweet is ironic or not. To solve this task, we propose a machine learning approach where a tweet is represented with several features calculated using shallow characters (e.g. length of the tweet and number of words), using lexicons (frequency lexicons and sentiment lexicons), and also knowledge-based systems (WordNet for the synonyms, and features related to synsets). Our approach outperformed state of the art when we published the results. Moreover, we avoided the use of words as features (like Bag of Words approaches), in order to be as much topic-independent as possible. Initial results on cross-domain experiments (where training and testing domains are different) shown in this chapter support this idea. In the next chapter we will study this problem of topic bias in greater depth.

Chapter 4

TOPIC BIAS IN IRONY DETECTION AND SENTIMENT ANALYSIS

In the previous chapter we introduced cross-domain experiments for irony detection and showed that systems based on words are not robust to topic changes. On the other side, the system that we proposed seems to better model irony when the testing domain is not seen in the training phase. In this chapter, we study in deep this problem, exploring the issue of topic bias in sentiment analysis and irony detection in Italian. Our model shows to be at state of the art level, and that it adapts for different domains since it uses features designed to be topic-independent. The experiments described in this chapter were presented in [Barbieri et al., 2015b].

4.1 Introduction

The automatic identification of sentiments and opinions expressed by users online is a significant and challenging research trend. The task becomes even more difficult when dealing with short and informal texts like tweets and other microblog texts. Sentiment analysis of tweets has been already investigated by several research studies [Jansen et al., 2009, Barbosa and Feng, 2010]. Moreover, during the last few years, many evaluation campaigns have been organized to discuss and compare sentiment analysis systems tailored to tweets. Among these campaigns, since 2013, in the context of SemEval [Nakov et al., 2013], several tasks targeting sentiment analysis of English Short Texts took place. In 2014, SENTIPOLC [Basile et al., 2014], the SENTIment POLarity Classification Task of Italian tweets, was organized in the context of EVALITA 2014, the fourth evaluation campaign of Natural Language Processing and Speech tools for Italian. SEN-

TIPOLC distributed a dataset of Italian tweets annotated with respect to subjectivity, polarity and irony. This dataset enabled training, evaluation and comparison of the systems that participated to the three tasks of SENTIPOLC, respectively dealing with Subjectivity, Polarity and Irony detection. In the Subjectivity task participants were asked to recognize whether a tweet was objective or subjective, in the Polarity Task they were asked to classify tweets as positive or negative, and finally, in the Irony task, they were asked to detect whether the content of a tweet was ironic or not. The study presented in this chapter was carried after SENTIPOLC 2014 [Basile et al., 2014], since we noticed that Bag of Words features were extremely important to score good results in the three tasks of the shared challenge. We show in this chapter that using Bag of Words features is convenient as the same topics are present in the training and testing tweets, and modelling these topics with a Bag of Words leads to good results. As we will see, words related to specific topics or persons, like a specific politician, are very informative for irony detection and negative tweets, but this is not a good feature to model irony in general. Imagine if the training tweets are about politics and the testing ones are not: this kind of features will not work.

4.2 Dataset and Text Processing

We used the dataset employed in SENTIPOLC – the combination SENTI-TUT [Bosco et al., 2013a] and TWITA [Basile and Nissim, 2013]. Each tweet was annotated over four dimensions: subjectivity/objectivity, positivity/negativity, irony / non-irony, and political/non-political topic. SENTIPOLC dataset is made of a collection of tweet IDs, since the privacy policy of Twitter does not allow to share the text of tweets. As a consequence we were able to retrieve by the Twitter API the text of only a subset of the tweets included in the original SENTIPOLC dataset. In particular, our training set included 3,998 tweets (while the original dataset included 4,513). The following tweets include an example of each SENTIPOLC class:

- **Objective tweet:**
  RT @user: Fine primo tempo: #FiorentinaJuve 0-2 (Tevez, Pogba). Quali sono i vostri commenti sui primi 45 minuti?#ForzaJuve
  (RT @user: First half: #FiorentinaJuve 0-2 (Tevez, Pogba). What are your comments on the first 45 minutes? #GOJUVE)

- **Subjective / Positive / Non-Ironic tweet:**
  io vorrei andare a votare, ma non penso sia il momento di perder altro tempo e soprattutto denaro.Un governo Monti potrebbe andare. E x voi?

(I would like to vote, but I do not think it is the moment to waste time and money. Monti's government might work. What do you think?)

- **Subjective / Negative / Ironic tweet:**
  Brunetta sostiene di tornare a fare l'economista, Mario Monti terrorizzato progetta di mollare tutto ed aprire un negozio di pescheria
  (Brunetta states he will work as an economist again, a terrified Mario Monti plans to leave everything and open a fish shop)

The first example is an objective tweet as the user only asks what are the opinions on the football match Fiorentina against Juventus. The second tweet is subjective, positive and non-ironic as the user is giving his positive opinion on the new government ("Monti's government might work"). The last tweet is subjective, negative and ironic since the user is making fun of the politician Brunetta (who stated he would work as an economist again), saying that the prime minister Monti is so worried that he is considering to open a fish shop instead of working with Brunetta as an economist.
Regarding text preprocessing, we follow the same approach of Section 3.2.

4.3 Model

Our model is the same of Section 3.3 while the bag of words used in this chapter is an extended version of the one presented in Section 3.3.8. We extend it by adding also n-grams and Wordnet synsets IDs. The baseline used in this chapter includes the two sets of features described in the following two sections.

4.3.1 Word-Based

We designed this group of features to detect common word-patterns. With these features we are able to capture common phrases used in certain type of tweet and grasp the common topics that are more frequent in certain type of tweet (positive/negative/ironic). We computed three word-based features: *lemma* (lemmas of the tweet), *bigrams* (combination of two lemmas in a sequence) and *skip one gram* (combination of three lemmas in a row, excluding the one in the middle).

4.3.2 Synsets

This group includes features related to WordNet Synsets. After removing stop words, we disambiguate each word against Wordnet (UKB) [Agirre and Soroa, 2009], thus obtaining the most likely sense (Synset) associated to each word, and use the ID of this WordNet synset as feature.

		Our system	**SENTIPOLC**
Subjectivity	subjective	0.866	0.828
	objective	0.564	0.601
	avg	**0.715**	0.714
Polarity (POS)	positive	0.554	0.823
	other	0.839	0.527
	avg	**0.697**	0.675
Polarity (NEG)	negative	0.619	0.717
	other	0.741	0.641
	avg	**0.680**	0.679
Irony	ironic	0.260	0.355
	non-ironic	0.916	0.796
	avg	**0.588**	0.576

Table 4.1: Results of our system and best system of SENTIPOLC in the three Tasks subjectivity, polarity, and irony. We show F-Measures scores for each class and the arithmetic average too.

4.4 Experiments and Results

In this Section we show the performance of our system in the three Tasks of SENTIPOLC 2014 (see Table 4.1). In order to compare our system with the best ones of SENTIPOLC, beside using the same dataset, we adopted the same experimental framework. Since each task was a binary decision (e.g. subjective vs objective), SENTIPOLC organizers computed the arithmetic average of the F-measures of the two classes (e.g. mean of F-Measures of subjective and objective).

We carried out a study of the features contribution to the classification process performing six classification experiments. In each experiment we added one of the feature groups described in the previous Section. Thus we were able to measure the effect that the addition of the features has on the F-measure.

In Section 4.4.4 we present an experiment useful to check if our classification features are effective across different domains.

4.4.1 Task 1: Subjectivity Classification

SENTIPOL 2014 Task 1 was as follows: *given a message, decide whether the message is subjective or objective.*

As we can see in Table 4.1, in the subjectivity Task our system scores a slightly higher F-Measure of the best system of SENTIPOLC (0.715 vs 0.714). The two systems behave in different ways: our system scored less in the detection of the

		Subj	**Pol (pos)**	**Pol (neg)**	**Irony**
BL	class 1	0.842	0.507	0.509	0.2
	class 2	0.335	0.829	0.720	0.913
	avg	0.589	0.668	0.6145	0.5565
BL + Ambig.	class 1	0.843	0.515	0.529	0.196
	class 2	0.327	0.833	0.716	0.914
	avg	0.585	0.674	0.623	0.555
	improvement	-0.004	0.006	0.008	-0.002
BL + Synset	class 1	0.835	0.514	0.520	0.239
	class 2	0.542	0.82	0.716	0.903
	avg	0.689	0.667	0.618	0.571
	improvement	**0.1**	-0.001	0.004	**0.015**
BL + Senti.	class 1	0.847	0.522	0.578	0.192
	class 2	0.520	0.833	0.731	0.911
	avg	0.684	0.678	0.655	0.552
	improvement	**0.095**	0.010	**0.040**	-0.005
BL + POS	class 1	0.847	0.513	0.542	0.192
	class 2	0.447	0.831	0.717	0.911
	avg	0.647	0.672	0.630	0.552
	improvement	**0.059**	0.004	**0.015**	-0.005
BL + Syno.	class 1	0.843	0.506	0.515	0.195
	class 2	0.322	0.828	0.718	0.913
	avg	0.583	0.667	0.617	0.554
	improvement	-0.006	-0.001	0.002	-0.0025
BL + Char.	class 1	0.832	0.532	0.559	0.212
	class 2	0.463	0.834	0.722	0.914
	avg	0.648	0.683	0.641	0.563
	improvement	**0.059**	0.015	**0.026**	0.007

Table 4.2: Features Analysis of our baseline (BL) and combinations with our features. In each task, class 1 and 2 are respectively: subjective/objective, positive/non-positive, negative/non-negative and ironic/non-ironic

objective class (0.564 vs 0.601), but it is more accurate in subjective detection (0.866 vs 0.828). The best sentipolc system relied on both word based features like bag of words and also sentiment lexicons features (e.g. average of the sentiment scores of the words present in the tweet).

In Table 4.2 we can examine the F-Measure improvement of each feature group. We can note that the greatest improvement is given by Synset and Sentiment features (adding respectively 0.1 and 0.95 points to the baseline); POS and Characters produce an increasing of 0.059, hence can be considered rich features as well. The groups Ambiguity and Synonym do not increase the accuracy of the classification.

Subjectivity	Polarity	Irony
monti	**syn (no, non, neanche)**	governo
syn (no, non, neanche)	**grazie**	passera
governo	monti	politico
syn (avere, costituire)	grillo	bersani_non
syn (essere, fare, mettere)	governo	monti
mi	**piacere**	se_governo
paese	**syn (avere, costituire)**	grillo
prince	**syn (essere, fare, mettere)**	bersani
essere_dire	paese	capello
of_Persia	**syn (migliaio, mille)**	cavallo

Table 4.3: For each test set topic the ten Word-based and Synset features with higher information gain are shown. The domain independent words are in bold. “Syn(word1, word2)" is the synset associated to word1 and word2.

4.4.2 Task 2: Polarity Classification

SENTIPOL 2014 task 2 required *given a message, to decide whether the message is of positive, negative, neutral or mixed sentiment (i.e. conveying both a positive and negative sentiment).*

SENTIPOLC annotators tagged each tweet with four tags related to polarity: positive, negative, mixed polarity, unspecified. As in SENTIPOLC we split up the Polarity classification in two sub-classifications. The first one is the binary classification of positive and mixed-polarity tweets versus negative and unspecified ones. The second one is focused on the recognition of negative tweets being the binary decision between negative and mixed polarity versus positive and unspecified tags.

In the positive classification, our system reached a F-Measure of 0.697, while the F-Measure of the best SENTIPOLC system was 0.675 (see Table 4.1). As previously, the systems behaved differently: ours lacked in detection of the Positive + Mixed-polarity class but it was able to achieve a good F1 in the negative + unspecified class. In the negative classification we outperformed the SENTIPOLC system with a score of 0.680 (versus a 0.675). Again, the best SENTIPOLC system got a better score in negative + mixed-polarity and ours reached a better F1 in positive + unspecified.

In the feature analysis (Table 4.2) we can see that the most important groups of features for the negative classification were Sentiments (giving an improvement of 0.040 points), Characters (0.026) and POS (0.015). On the other hand, in the Positive classification, the word-base features seem to be the most important suggesting that word-patterns were very relevant for this task.

		pol / non-pol	**non-pol / pol**
Subjectivity	dom. dependent	0.734	0.672
	dom. indepentent	**0.767**	**0.746**
	all	0.747	0.689
Polarity (POS)	dom. dependent	0.555	0.631
	dom. indepentent	0.443	**0.736**
	all	**0.583**	0.728
Polarity (NEG)	dom. dependent	0.614	0.554
	dom. indepentent	**0.671**	**0.624**
	all	0.663	0.567

Table 4.4: Cross-domain experiments, where “political / non-political" means training in politics dataset and testing in non-political dataset, “non-political / political" vice-versa. For these two domain combinations we report the results of three models: “domain dependent" (word-based + synset), “domain independent" (Sentiment, Synonyms, Character, Ambiguity), and the model “all" with all the features of our model.

4.4.3 Task 3: Irony Detection

SENTIPOLC 2014 Task 1 asked *given a message, to decide whether the message was ironic or not.* Our system scored a F1 of 0.588 (0.26 in the irony class, and 0.916 in non-irony) while best SENTIPOLC system a F1 of 0.5759 (0.3554 in the irony class and 0.7963 in non-irony). In this task, the use of the words and domain dependent features is very relevant. None of the other domain independent features increase the F1. The only feature that gives a F1 increase is Synset, which can be considered domain dependent. With the help of Table 4.3 we can see that the ten most important textual features in the irony task are related to a specific topic, since 4 out of 10 words are names of politicians (Passera, Bersani, Monti, Grillo) and other 4 are related to politics (with words like “politics" or “government"). Of course a name of a Politician can not be a good feature for irony detection in general.

4.4.4 Cross-Domain Experiments

In this section we show the results of the cross-domain experiments. We trained our classifier with the tweets of one topic (politics related tweets) and tested the same classifier with the tweets related to the other topic (non-politics related tweets). In this way, we can examine whether the model is robust with respect to domain-switches. We were able to run these experiments as SENTIPOLC tweets provided a topic flag that points out if a tweet is political or not. We obtained two different systems dividing our features in two groups: domain dependent (word-based and synset group) and domain independent (Sentiment, Synonyms, Character, Ambiguity). We run the cross-domain experiments over the Subjectivity and Polarity datasets with these two systems, and also with our model (“all"). Unfortunately, we were not able to run cross-domain experiments on irony as there were not enough data to effectively train a classifier (e.g. non-political ironic tweets were only 39 in the test set).

We can see in Table 4.4 that in the cross-domain experiments domain independent features are five out of six times outperforming the domain dependent system. Moreover an interesting result is that in five out of six combinations the domain independent system outperforms the respective “all" features system, suggesting that when the domain changes, domain dependent features introduce noise.

4.5 Discussion

In this section we will discute all the three task of SENTIPOLC, with special focus on the cross domain problem of Bag of Words systems.
The system that we propose outperformed the best SENTIPOLC systems in all the tasks. However, as showed in the previous section, not all of our features are effective for the SENTIPOLC Tasks. Specifically, in Polarity and Irony Tasks the features with biggest impact on the classification accuracy resulted to be the domain dependent ones. We can identify two possible explanations. The first one is that for these Tasks is very important to model pattern that are representative of the different classes (for example common phrases used in negative tweets to detect this class). The second hypothesis is that word-based features, that are often used to model a domain, worked well because training and test set of the dataset shared the same topics. Hence, word-based features worked well because there was a topic bias. For example, in the case of the Polarity Task, a word-based system could detect that often the name of a certain politician is present in the negative tweets, then using this name as feature to model negative tweets. With cross-domain experiments we confirmed the second hypothesis, showing that word-based features are not robust when the topic of training and test set are different. On the other hand domain independent features do not decrease their performance when training and test do not share the same topics.

However, in the SENTIPOLC task domain dependent features were relevant, and detecting the topic of a specific class was important. We show (Table 4.3) that the ten best word-based features are often related to a specific topic (politics in this particular case, see Table 3) rather than to typical expression (e.g. “worst", “don’t like" to mean something negative), meaning that our word-based features modelled a specific domain. For example, using words like “Monti" and “Grillo" who are two Italian politicians is important to detect negative tweets. These features may be in some cases important but they narrow the use of the system to the domain of the training set (and eventually to tweets generated in the same time-frame).

In the light of these results, we suggest that if a Sentiment Analysis system has to recognize polarity cross-domain should avoid word-based features and focus more on features that are not influenced by the content. On the other hand, if the a Sentiment Analysis system is used in a specific domain, words may have an important role to play.

4.6 Conclusions

In this chapter we show that the topic of a dataset can bias classification systems of irony and sentiments. Our model includes two types of features: domain-dependent and domain-independent features. We showed with cross-domain experiments that the use of domain-dependent features may constrain a system to work only on a specific domain, while using domain-independent features achieved domain independence and a greater robustness when the topic of the tweet changes.

Chapter 5

SATIRE DETECTION

Satire is a powerful linguistic device used in several situations and contexts. In this chapter we study the use of satire and show that our irony detection system is able to model Twitter satirical news. The idea is to recognize whether a news post on Twitter is real or satirical, carrying out experiments in three different languages. For this task we employ our English irony detection system (Chapter 3) that we extended to Italian (like in Chapter 4) and Spanish. This chapter describes the experiments carried on in [Barbieri et al., 2015a, Barbieri et al., 2015c, Barbieri et al., 2014a].

5.1 Introduction

Satire is a form of language where irony is employed to criticize and ridicule someone or something. Even if often misunderstood, "in itself, satire is not a comic device —it is a critique — but it uses comedic devices such as parody, exaggeration, slapstick, etc. to get its laughs." [Colletta, 2009]. Satire is distinguished by figurative language and creative analogies, where the fiction pretends to be real. Satire is also characterized by emotions (like anger and disappointment) that are hard to detect due to their ironic dimension.

The ability to properly detect and deal with satire is strongly beneficial to several fields where a deep understanding of the metaphorical traits of language is essential, including Affective Computing [Picard, 1997] and Sentiment Analysis [Turney, 2002, Pang and Lee, 2008].

Looking at the big picture, computational approaches to satire are fundamental to build a smooth human-computer interaction, improving the way computers interpret and respond to peculiar human emotional states.

[Burfoot and Baldwin, 2009] proposed one of the few attempts to computationally model satire in English. They retrieved news-wires documents and satire

Language	Non-Satirical	Satirical
English	The Daily Mail (UK) The Times (UK)	NewsBiscuit The Daily Mash
Spanish	El Pais El Mundo	El Mundo Today El Jueves
Italian	Repubblica Corriere della Sera	Spinoza Lercio

Table 5.1: The accounts are in British English, Iberian Spanish, and Italian.

news articles from the web and built a model able to recognize satirical articles. Their approach included standard text classification, lexical features (including profanity and slang) and semantic validity where they identify the named entities in a given document and query the web for the conjunction of those entities. The differences with our approach are various: we use different features to model satire, and we only work on short text (tweets) while their work is focused on whole news articles.

In this chapter we study the characterization of satire in social networks, experimenting new approaches to detect satirical tweets inside and across languages. In order to carry out our research we needed a dataset of satirical and real news, since we wanted to study satire in Twitter. Therefore, we retrieve satirical tweets from popular satirical news Twitter accounts, in English, Spanish and Italian. We approach the problem as a binary classification task, where a news post can be satirical or real. Our machine learning approach relies on the features discussed in Section 5.3.4 (word usage frequency in a reference corpus, number of associated meanings, etc.) and on *word-based features* (lemmas, bigram, skip-gram). As a classifier we employ the supervised algorithm Support Vector Machine[1] [Platt et al., 1999] because it has proven to be effective in text classification tasks.

5.2 Dataset and Text Processing

In order to train and test our system we retrieved tweet posted from June 2014 to January 2015 from twelve pupular twitter accounts. We considered four Twitter accounts for each language that we study: English, Spanish and Italian. Within each language two accounts tweet satirical news and two tweet real news accounts (that we consider the negative class non-satirical). The accounts we use are shown in Table 5.1.

We rely on these accounts since their content is a contribution of several people

[1] LibLINEAR: http://www.csie.ntu.edu.tw/cjlin/liblinear

and their popularity reflects the interest and appreciation for this type of content. A few examples from the dataset are the following ones:

- **Satirical News**
  English: Police Creative Writing Awards praise ‘most imaginative witness statements ever’.
  Spanish: Artur Mas sigue esperando el doble “check" de Mariano Rajoy tras la votación del 9-N.
  (Artur Mas is still waiting for Mariano Rajoy’s double check after 9-N consultation).
  Italian: “Potrei non opporre veti a un presidente del Pd", ha detto Berlusconi iscrivendosi al Pd.
  (“I might not limit powers of Democratic Party president", said Berlusconi enrolling in the Democratic Party).

- **Non-Satirical News**
  English: Major honours for The Times in the 2014 British Journalism Awards at a ceremony in London last night.
  Spanish: Rajoy admite que no ha hablado con Mas desde que se convocó el 9-N y que no sabe quién manda ahora en Cataluña.
  (Rajoy admits that he hasn’t talked to Mas since the convocation of 9-N consultation and that he doesn’t know who’s governing in Catalonia).
  Italian: Berlusconi e il Colle: “Non metterò veti a un candidato Pd".
  (Berlusconi states: “I will not limit powers of a Democratic Party candidate").

In these examples we can see that satire is used to criticize and convey a peculiar hidden meaning to the reader. The satirical English example is a critic against police and its dishonest way of solving issues by “inventing” witnesses. The satirical Spanish tweet is a critic against Rajoy (Prime Minister of Spain at the time of writing), as he did not want to discuss with Mas (Prime Minister of Catalonia at the time of writing) the decision of doing a consultation on November 9th 2014 (on the Catalonia independence). For this reason “Mas is still waiting for him to consider it". The satirical tweet in Italian criticizes the power Berlusconi had in Italy even though he was not Italian prime minister any more, and the message is ironic as it means the opposite, indeed Berlusconi is going to limit the powers of the democratic president.

Text Processing

After downloading the tweet we filtered them removing the tweet that were not relevant to our study (for instance: “Buy our t-shirt" or “Watch the video"). We left only tweet that were actual news (satirical or non-satirical). We normalize the text of each tweet by expanding abbreviations and slang expressions (with manual rules for each language), properly converting hashtags into words whether they have a syntactic role (i.e. they are part of the sentence), and removing links and mentions (“@user”).

In order to have a balanced dataset, with the same contribution from each Twitter account, we selected 2,766 tweet randomly from each account, obtaining a total of 33,192 tweet, where half (16,596) were satirical and half were non-satirical news (2,766 was the least number of tweet that a single account included, which was the Italian satirical account “Lercio").

5.3 Model

We use the same model of Section 3.3, however, since we need to deal with different languages in these experiments, we use different resources, as much as possible consistent among languages. In the next three subsections we report the detail of the tools used for each language, and in Section 5.3.4 we summarize the model used here.

5.3.1 English Resources

We made use of the GATE application TwitIE [Bontcheva et al., 2013] where we enriched the normalizer, adding new abbreviations, new slang words, and improving the normalization rules. We also employed TwitIE for tokenization, Part Of Speech (POS) tagging and lemmatization. We used WordNet [Miller, 1995] to extract synonyms and synsets of a word. We employed the sentiment lexicon SentiWordNet3.0 [Baccianella et al., 2010]. Finally, the American National Corpus [2] has been employed as frequency corpus to obtain the usage frequency of words in English.

5.3.2 Spanish Resources

We relied on the tool Freeling [Carreras et al., 2004] to perform sentence splitting, tokenization, stop words removal, POS tagging, and Word Sense Disambiguation

[2] http://www.anc.org/

(UKB [Agirre and Soroa, 2009]). We use the Spanish Wordnet of the TALP Research Centre, mapped by means of the Inter-Lingual-Index to the English WordNet 3.0 whose synset IDs are in turn characterized by sentiment scores by means of SentiWordNet. Frequency usage of Spanish words were calculated using a dump of the Spanish Wikipedia as of May 2014.

5.3.3 Italian Resources

We tokenized, POS tagged, applied Word Sense Disambiguation (UKB [Agirre and Soroa, 2009]) relying on Freeling. We used the Italian WordNet1.6[3] to get synsets and synonyms of each word of a tweet. The setiment lexicon used was Sentix [Basile and Nissim, 2013], that is derived from the English SentiWordNet. We relied on the CoLFIS Corpus frequency of Written Italian[4] for the frequency usage of Italian words.

5.3.4 Model Summary

The model used for the satirical news experiments is slightly different than the one proposed in Chapter 3. This is because we have to deal with more languages hence different (less) resources. We were not able to implement the feature groups "written/spoken" and "intensity" as we could not find the resources to calculate these features in Spanish and Italian. The rest of the features are the same, apart from one difference: we added for the various group of features additional features measured only within one of four Part of Speech (adjectives, verbs, nouns and adverbs). For instance, the feature *rarest word* is calculated for each POS (e.g. rarest noun copared to the other nouns in the tweet, rarest verb compared to the other verbs in the tweets, and so on).
What follow is a summary of the model employed in these experiments.

Frequency

We derive three types of Frequency features: *rarest word frequency* (frequency of the most rare word included in the tweet), *frequency mean* (the arithmetic average of all the frequency of the words in the tweet) and *frequency gap* (the difference between the two previous features). These features are computed including all the words of each tweet. We also determined these features by considering only Nouns, Verbs, Adjectives, and Adverbs. Moreover, we count the number of bad/slang words in the tweet (using three lists we compiled for each language). The final number of Frequency features is 16.

[3] http://multiwordnet.fbk.eu/english/home.php

[4] http://linguistica.sns.it/CoLFIS/Home_eng.htm

Ambiguity

To model the ambiguity of the words in the tweet, we use the WordNet synsets associated to each word. For each tweet we calculate the *maximum number of synsets* associated to a single word, the *mean synset number* of all the words, and the *synset gap* that is the difference between the two previous features. We determine the value of these features by including all the words of a tweet as well as by considering only Nouns, Verbs, Adjectives or Adverbs separately. The Ambiguity features are 15.

Synonyms

We consider the frequencies of the synonyms of each word in the tweet, as retrieved from WordNet. Then we computed, across all the words of the tweet: the *greatest* and the *lowest number of synonyms* with frequency higher than the one present in the tweet, the *mean number of synonyms* with frequency greater/lower than the frequency of the word. We determine also the greatest/lowest number of synonyms and the mean number of synonyms of the words with frequency greater/lower than the one present in the tweet (*gap* feature). We computed the set of Synonyms features by considering both all words of the tweet together and only words belonging to each one of the four POS: Nouns, Verbs, Adjectives, and Adverbs.

Sentiments

Relying on the three Sentiment lexicons described in previous three sections, we computed the *number of positive / negative words*, the *sum of the intensities of the positive / negative scores of words*, the *mean of positive / negative score of words*, the *greatest positive / negative score*, the *gap between the greatest positive / negative score and the positive / negative mean*. Moreover, we simply count the *words with polarity* not equal to zero, to detect subjectivity in the tweet. For the same reason we also measure the ratio of words with polarity in the tweet. As previously done, we computed these features by considering only Nouns, Verbs, Adjectives, and Adverbs.

Structure

There are two types of features in the structure groups, Characters and Part of Speech.
Characters: The charater based features were designed to capture the punctuation style of the satirical tweet. Each feature that is part of this set is the number of a specific punctuation mark, including: “.”, “!”, “?”, “$”, “%”, “&”, “+”, “-”, “=”.

We also compute numbers of Uppercase and Lowercase characters, and length of the tweet.

Part Of Speech: The Part of Speech features were designed to capture the syntactic structure of the tweet. The features of this group are eight and each one of them counts the number of occurrences of words characterized by a certain POS. The eight POS considered are *Verbs*, *Nouns*, *Adjectives*, *Adverbs*, *Interjections*, *Determiners*, *Pronouns*, and *Apposition*.

5.4 Experiments

In order to test the effectiveness of our approach we carried out monolingual (Section 5.4.1) and cross-lingual experiments (Section 5.4.2).

5.4.1 Monolingual Experiments

We run two kind of balanced binary classification experiments, where the two classes are "satire" and "non-satire". We gathered three datasets of English, Spanish and Italian tweet; each dataset includes two newspaper accounts, N1 and N2, and two satirical news accounts, S1 and S2.

In the **first binary balanced classification experiment**, we train the system on a dataset composed of 80% of tweet from one of the newspaper accounts and 80% of tweet from one of the satirical accounts (5,444 tweet in total). Then we test the system on a dataset that includes 20% of the tweet of a newspaper account that is different from the one used for training and 20% of the tweet of a satirical account that has not been used for training. The final size of our testing set is 1,089 tweet. For example, we train on The Daily Mail vs NewsBiscuit, and test on The Times vs The Daily Mash.

We run the following configurations:

- Train: 80% N1 and 80% S1 / Test: 20% N2 and 20% S2
- Train: 80% N1 and 80% S2 / Test: 20% N2 and 20% S1
- Train: 80% N2 and 80% S1 / Test: 20% N1 and 20% S2
- Train: 80% N2 and 80% S2 / Test: 20% N1 and 20% S1

It is relevant to remark that thanks to these training and test set configurations, we never use tweet from the same account in any of the training and testing datasets, thus we were able to evaluate the ability of our system to detect satire independently from the linguistic and stylistic features of a specific Twitter account. As a consequence we avoid the *account modelling / recognition* effect, as the system is never trained on the same accounts where it is tested.

In the **second binary balanced classification experiment**, the training set is composed of all the tweet of each account. The dataset include 33,192 tweet, and we evaluate the performance of our SVM classifier by a 5-folds cross validation.

For each experiment we evaluate a word-based model (W-B, word-based features from Section 3.3.8) that we consider our baseline, a model that relies on our features (described in Section 5.3.4), and a third model that includes both models (our features together with the word-based features).

5.4.1.1 English Experiments

In Table 5.2 are reported the results of the English classification: the first four lines are the results of the first experiment, while the last line include the results of the second experiment (we report in bold the best results between our model and the word based baseline, confirmed by a two-matched-samples t-test with unknown variances). Except in the case of training on The Times and The Daily Mash, and testing on Daily Mail and NewsBiscuits, the word-based features obtained worst results than our model. In all the other cases, including the classification of all satirical tweet versus all non satirical ones (N1+N2 vs S1+S2), our model outperforms the word-based one. We can note that the results of the second experiment are higher with respect to any feature set, especially if we consider word-based features. We have to highlight that unlike the first experiment, in the second experiment tweet from the same accounts are used both for training and testing, thus the system can possible learn to recognize the language and writing style features of each account.

When we extended our feature set by adding also word based features (column "all" of Table 5.2), we can observe that the performance of the classifier improves (up to 0.678 in one combination, and up to 0.801 in the union of the accounts). We have also computed the information scores to assess the strength of each features of our model. The information score results suggest that the best features in the N1+N2 vs S1+S2 dataset (see Table 5.6) belongs to the groups Character (length of the tweet, the number of First uppercase words), POS (number of nouns) and Sentiment groups (ratio of words with polarity), Ambiguity (synset gap of nouns) and Frequency (rarest word frequency).

5.4.1.2 Spanish Experiments

The Spanish model performances are reported in Table 5.3. F-measures are promising, with the best score (0.805) when training on the accounts El Mundo and El Mundo Today using only our features. Our model outperformed the word-based

Train	Test	W-B	Our	All
N1S1	N2S2	0.646	**0.736**	0.683
N1S2	N2S1	0.610	**0.621**	0.660
N2S1	N1S2	0.641	**0.659**	0.678
N2S2	N1S1	0.632	0.635	0.639
N1N2S1S2	*5-fold*	0.752	**0.763**	0.801

Table 5.2: The table shows the F1 of each model, where N1=The Daily Mail, N2=The Times, S1=NewsBiscuit and S2=The Daily Mash. In **bold** the best results (not by chance confirmed by two-matched-samples t-test with unknown variances) between word-based and our model.

baseline in all the classifications. When adding the word-based features to our features the results decrease in three cases out of four. Moreover word-based model obtained worse results also in the N1+N2 vs S1+S2 classification, even with the chance of modelling specific accounts. We can see in Table 5.6 that best features for Spanish were the Character (length, uppercase character ratio), POS (number of noun and appositions) and Frequency group (frequency gap of nouns, rarest noun and rarest adjective) and Ambiguity (mean of the number of synsets).

Train	Test	W-B	Our	All
N1S1	N2S2	0.622	**0.754**	0.727
N1S2	N2S1	0.563	**0.712**	0.723
N2S1	N1S2	0.592	**0.805**	0.709
N2S2	N1S1	0.570	**0.778**	0.737
N1N2S1S2	*5-fold*	0.738	**0.816**	0.852

Table 5.3: The table shows the F1 of each model, where N1=El Pais, N2=El Mundo, S1=El Mundo Today, and S2=El Jueves. In **bold** the best results between word-based and our model (same statistical test than English).

5.4.1.3 Italian Experiments

In the Italian experiments (Table 5.4) our model outperformed the word-based model in all the combinations obtaining the best result when training on Repubblica and Lercio and testing on the other accounts (F1 are respectively 0.746 and 0.541). Incorporating word-based features to our features model increased the F1 in two cases and decrease in the other two. However in the second type of experiment adding word-features helps. In Table 5.6 we can see that the best groups of features to detect satire were Characters (uppercase and lowercase ratio, length)

POS (number of verbs), Ambiguity (verb synset mean, gap and max number of synset), Frequency (verb mean, gap, and rarest). In general, verbs seems play an important role in satire detection in Italian.

Train	**Test**	**W-B**	**Our**	**All**
N1S1	N2S2	0.518	**0.725**	0.672
N1S2	N2S1	0.541	**0.746**	0.674
N2S1	N1S2	0.527	**0.618**	0.640
N2S2	N1S1	0.578	**0.612**	0.625
N1N2S1S2	*5-fold*	0.739	**0.800**	0.842

Table 5.4: The table shows the F1 of each model, where N1=Repubblica, N2=Corriere della Sera, S1=Spinoza, and S2=Lercio. In **bold** the best results between word-based and our models (same statistical test than English).

5.4.2 Cross-Lingual Experiments

In addition to these experiments focused on a single language, we also analyzed the performances in a multi-lingual context. We were able to employ our system since, unlike word-based system, our system can be used across languages. We run two types of experiments. In the **first cross-language experiment** we train our model on the tweet in a language and test the model over the tweet of a different language; this way, we can see if the satirical accounts of different languages cross-reinforce the ability to model satire. By considering each language pair, we trained our satirical tweet classifier on a language and tested it on another one.

We carry out these experiments to gain a deeper understanding of our model assessing whether a model induced from one language can be used to detect the satire phenomena in a different language.

The **second cross-language experiment** was a 5-folds cross validation over all the dataset, including the tweet from all the accounts of all the languages (total of 22,228 tweet, where 16,596 were satirical and 16,596 non-satirical news).

Table 5.5 shows the results of the cross-lingual experiments (F1 of Non-Satirical and Satirical classes and the mean). A model trained in one language is not always capable of recognizing satire in a different language. For example, a model trained in Italian is not able to recognize English and Spanish satire (F1 of 0.05 and 0.156). However, when testing in Italian and training in English and Spanish the system obtains the highest F1 scores of this type of experiment (respectively 0.632 and 0.695). When testing in English the system recognizes satire (0.669) but not newspapers (0.031) when trained in Spanish, and vice versa when trained in Italian (good F1 for non-satirical newspaper, but low for satire). When testing

Spanish (while training in an other language) the system seems better recognizing newspapers rather than satire.

One of the most interesting result is the 5-fold cross validation over the whole dataset, including all the accounts of all the languages (last raw of Table 5.5). The F1 score of this experiment is 0.767 and it can be considered a high score considering the noise that could derive when we generate the same features in different languages. Indeed, the word-based model scores 8 point less.

Train	Test	Non-sat.	Satire	Mean
Majority Baseline	-	0	0.666	0.333
English	Spanish	0.676	0.475	0.575
English	Italian	0.710	0.555	0.632
Spanish	English	0.031	0.669	0.350
Spanish	Italian	0.657	0.733	0.695
Italian	Spanish	0.664	0.050	0.357
Italian	English	0.665	0.156	0.410
All word-based	*(5-folds)*	0.659	0.713	0.686
All our	*(5-folds)*	**0.765**	**0.769**	**0.767**

Table 5.5: Train in one language and testing in a different one, and in the last two rows a 5-folds cross validation on the whole dataset (all accounts of all languages) using the word-based and our model.

5.5 Discussion

Across the three languages, we need to consider the different quality of the linguistic resources adopted and the different accuracy on the NLP tools exploited to analyze tweet, as they can introduce some biases. Hence we need to take in account these issue in the interpretation of the results of our cross-lingual experiments. For instance, English Wordnet is considerably richer and more structured than the Italian and Spanish ones.

Our model outperforms the word-based baseline in each single language experiments, showing that the use of our features represent a good approach for satire detection across the three languages we considered. The best performance of our model occurs in the Italian dataset, where our model obtains an F-measure of 0.746 in one combination, while the word-based model scores only 0.541. Adding word-based features to our model seems to increase the performance only in the second type of experiment, where tweets from all accounts (news or satirical) are included in the training set. Yet, the word-based features are strictly

English	**Spanish**	**Italian**	**Eng+Spa+Ita**
[C]lenght,0.19	**[C]**lenght,0.28	**[C]**upcase-ratio,0.2	**[C]**lenght,0.08
[P]noun,0.11	**[C]**upcase-ratio,0.11	**[C]**lowcase-ratio,0.1	**[C]**upcase-ratio,0.07
[C]first-upcase,0.07	**[C]**lowcase-ratio,0.08	**[P]**verb,0.08	**[C]**lowcase-ratio,0.07
[SE]w-with-pol,0.06	**[C]**first-up,0.08	**[C]**lenght,0.07	**[P]**noun,0.06
[SE]pos-ratio,0.05	**[P]**noun,0.07	**[A]**syns-avg-verb,0.07	**[C]**first-up,0.03
[C]avg-w-lenght,0.04	**[C]**longst-word,0.04	**[A]**syns-gab-verb,0.06	**[SE]**w-with-pol,0.02
[C]upcase-ratio,0.04	**[C]**Exclamation,0.04	**[A]**syns-max-verb,0.05	**[C]**long-short-gap,0.01
[P]Determiner,0.04	**[C]**long-short-gap,0.03	**[F]**freq-avg-verb,0.05	**[A]**syns-max-verb,0.01
[A]syns-gap,0.04	**[C]**avg-w-lenght,0.03	**[F]**freq-gap-verb,0.04	**[SE]**pos-ratio,0.01
[SE]noun-with-pol,0.04	**[P]**adposition,0.03	**[F]**rarest-verb,0.04	**[A]**syns-gap-noun,0.01
[P]verb,0.04	**[F]**freq-noun-gap,0.03	**[P]**noun,0.03	**[C]**longst-word,0.01
[C]shortest-w,0.04	**[F]**rarest-adj,0.03	**[C]**longst-word,0.03	**[A]**syns-max-noun,0.01
[F]freq-avg,0.03	**[F]**rarest-noun,0.03	**[C]**long-short-gap,0.03	**[C]**shortest-w,0.01
[F]freq-gap,0.03	**[P]**number,0.02	**[A]**syns-max,0.03	**[F]**freq-avg-verb,0.01
[F]freq-gap-verb,0.03	**[A]**syns-avg,0.02	**[P]**pronoun,0.025101	**[F]**rarest-verb,0.01

Table 5.6: Best 15 features of our model ranked considering the information gain scores in the N1+N2 vs S1+S2 dataset. In the last column are reported the best features considering the arithmetic average of the information gain of each language. In **[bold]** are reported the group of each feature. A=Ambiguity, C=Characters, F=Frequency, P=POS, S=Sentiments, S=Synonyms

related to the words used by specific accounts. The use of word-based features is not domain and language independent because it is strictly related to specific words rather than inner “cross-account" and “cross-language" linguistic traits of satire.

The best features (see Table 5.6) across the languages were Characters, Part Of Speech and Ambiguity. In English we note that beside the Characters features (relevant in all the languages), the number of words with polarity (positive or negative) is important (but not that important for Spanish and Italian). Additionally, the use of rare, infrequent words, is a characteristic of English satire. What distinguishes Spanish satire is the number of nouns and appositions, and the use of long words. In this language also the detection of rare nouns and rare adjectives is a distinctive feature of satire. In Italian, the Characters feature are also important, especially the uppercase and lowercase ratio. Moreover, in Italian satire verbs play a key role. Indeed the number of verbs, the number of synsets associated to a verb and the frequency usage of a verb (whether it is rare or not) are strongly indicative for Italian satirical news. Furthermore, as in Spanish, using long words may be sign of Italian satire.

One last curious result is that the use of slang and bad words does not appear to be a relevant feature if compared to the satire detection contributions of structural features (Characters and Part of Speech) and semantic features (like ambiguity). This fact suggests that the satirical news of the accounts we selected mimic appropriately non-satirical news.

Based on our cross-lingual experiments, we argue that it is not always possible to train in one language and test in another one with the proposed model (Table 5.5). Yet, there are interesting results. For instance, when training in Italian the system is not able to detect English and Spanish satire, but when testing on Italian and training in the other languages results are better. The interpretation may be that Italian satire is less intricate, easy to detect but not able to recognize other kind of satire. Our model when trained in Spanish is able to detect Italian satire with a precision of 0.695 (with satire F1 of 0.733), which is a very interesting result considering the complexity of the task. We need to consider that the two datasets are written in different languages, and the satirical topics are different (as they are related to politics and culture). On the other hand English can not be detected by Spanish nor Italian systems, but they both can recognize an aspect of the English dataset (Spanish recognizes English satire, and Italian recognizes with good accuracy, F1 of 0.71, English newspapers). Finally, the last results that deserve further analysis is the 5-fold cross validation over the all dataset, where all the accounts of all the languages were included. The accuracy of our model is promising (F1 of 0.767) as in this dataset the noise is very high: 22,228 tweet on three different languages and different topics.

5.6 Conclusions

In this chapter we employ our irony detection approach to detect satirical news in Twitter in different languages. Our approach avoids the use of word-based features (Bag of Words), by relying only on language-independent features that aim to detect inner characteristics of the satirical tweets. We tested the approach on English, Spanish and Italian tweets and obtained significant results. Our system was able to recognize if a tweet advertises a non-satirical or satirical piece of news, outperforming a word-based baseline. Moreover, we tested the system with cross-language experiments, obtaining interesting results that deserve a deeper investigation.

Chapter 6

EMOJI SEMANTICS

In this chapter we study how the meaning and usage of emojis varies across languages and locations (by analyzing tweets posted from United States of America, United Kingdom, Spain and Italy) as well as across seasons (by analyzing tweets posted in spring, summer, autumn, and winter). We use distributional semantic models to represent the meaning of the emojis in each language and season respectively. We compare the semantics of emojis across languages and seasons by means of two experimental approaches relying respectively on the analysis of Nearest-Neighbor emojis and on the comparison of similarity matrices of emojis. The overall semantics of the emojis is preserved across languages and seasons, but we spotted some emojis that are used differently depending on the language or the season.

6.1 Introduction

During the last few years, Twitter users have started to extensively use emojis in their posts. Emojis, as we have seen in Section 2.2.2, are pictures that can be naturally combined with plain text to create a new form of language. Such a practice has also been widely adopted in other networking platforms such as Facebook, Whatsapp and Instagram. Emojis pose important challenges for researchers in multimedia information systems, since their meaning is scarcely explored. In spite of their assumed universality, the sense of an emoji may change from language to language, from location to location and from time to time. For instance, an emoji may undergo substantial changes of its semantics and usage patterns in a specific season because of the adoption of that emoji to refer to a specific event occurring at that time. Understanding the meaning of emojis with respect to their context of use is important for multimedia information indexing, retrieval, or content extraction systems.

In this chapter, we investigate how distinct languages, locations and different periods of the year affect the way we use emojis. We adopt an empirical research methodology by relying on vector space representations [Turney et al., 2010, Mikolov et al., 2013b] to model and thus understand the "semantics" of these important elements of multimedia communication. More specifically, we collected a corpus of more than 80 million tweets in four languages, American English (USA), British English (UK), Peninsular Spanish (ESP), and Italian (ITA), and carried out various experiments to compare emojis. In this chapter we propose a method to compare emojis over different languages and locations, avoiding the need to rely on language-specific information. We adopted the same approach to compare the semantics of emojis in different periods of the year by considering tweets posted from the USA in different seasons: spring, summer, autumn and winter.

Through our methodology, we were able to observe interesting emojis usage patterns that are either specific of a particular language or a season of the year. For instance, the emojis 🔥, 👋🏽, and 🙌🏼 seem to be used in different contexts across distinct languages, while there is a relative agreement on the cross-language use and meaning of 🎶 and 🌳. Other examples are the emojis 🎤, 🍰 and 🐟 that keep their meaning equal across different seasons, while the semantics of 🌲 and 🎁 considerably varies with respect to the season of the year considered.

In Section 6.2, we describe the dataset we collected from Twitter and used to carry out the experiments presented in this work: we quantify the presence and usage of emojis in our dataset by putting special focus on the use of two or more emojis inside a single tweet. Section 6.3 introduces the text processing tools we exploited to parse the textual content of tweets and the approach we adopted to shape and compare the meaning of emojis across languages, locations and seasons, based on vector semantic models. In Section 6.4, we describe the experimental framework we employed to investigate the semantics of emojis.

In particular, we investigate how the context of use of emojis varies in different locations and through different seasons of the year.

We show and discuss the results of the experiments in Section 6.5 and conclude the chapter with Section 6.6, a summary of our findings and avenues for further research.

6.2 The Twitter Dataset

In order to support the creation of the semantic vector models presented in this chapter, we gathered a dataset composed of more than 80 million tweets retrieved by relying on the Twitter Streaming APIs[1]. From October 2015 to November 2016 we collected all the geo-located tweets posted from the following four countries: United States of America (USA), United Kingdom (UK), Spain, and Italy. We decided to include in our dataset only geo-located tweets in order to retrieve posts from real user, filtering out spam and bot-generated messages as much as possible. Moreover, using the automatic language identification provided by Twitter, for each one of the four countries considered we collected only geo-located tweets in a specific target language: English for United States of America and United Kingdom, Spanish for Spain and Italian for Italy.

The text of each Tweet was preprocessed by a modified version of the CMU Tweet Twokenizer [Gimpel et al., 2011], where we changed several regular expressions and integrated a Twitter emojis vocabulary to consistently deal with the presence of emojis during the tokenization process.

In order to study the variation of the meaning of emojis across different seasons, we considered the English tweets posted from USA, and divided them into four subsets to compare: tweets posted in spring, summer, autumn and winter. We decided to analyze English tweets from USA since in our dataset they represent the biggest collection of tweets from a specific country and language.

6.2.1 Size of the dataset and usage of emojis

Our dataset of about 80 million tweets is distributed across countries and languages as shown in Table 6.1. We rely on the collections of tweets from different country-language pairs to compare the semantics of emojis across locations and languages (see Section 6.5.1). English tweets posted from USA represent the largest part of our dataset (more than 66 million), followed by English tweets from UK (8 million), Spanish tweets from Spain (3.3 million) and Italian tweet from Italy (2.5 million). Spanish is the language where emojis are used the most since almost half of the tweets retrieved from Spain (47%) includes at least one emoji. One in four Italian tweets (25%) contains one or more emojis, whereas emojis are used only in 19% and 13% of tweets from UK and USA respectively.

Table 6.1 shows also the distribution of English tweets posted from USA across seasons: such distribution is exploited to investigate how the semantics of emojis varies across different seasons (see Section 6.5.2). The number of tweets is comparable from season to season, even if in winter we can notice a slight decrease of Twitter activity. Also the number of tweets including at least one emoji is very similar across seasons. We expected such patterns since we collected all the geo-located English tweets posted from USA during 12 consecutive months, three for each season.

[1] https://dev.twitter.com/streaming/overview

	Tweets	Tweets W/ E	Emojis
USA	66.4	8.8 (13.3%)	11.5 (1.30)
UK	8	1.5 (19.3%)	1.9 (1.24)
ESP	3.3	1.5 (47.0%)	1.9 (1.26)
ITA	2.5	0.6 (25.2%)	0.8 (1.33)
TOT	80.2	12.5 (15.6%)	16.2 (1.29)
SPR	15.2	1.4 (9.2%)	1.7 (1.19)
SUM	14.9	1.5 (10.2%)	1.8 (1.18)
AUT	15.9	1.5 (9.5%)	1.7 (1.14)
WIN	13.6	1.3 (9.4%)	1.4 (1.16)
TOT	59.6	5.7 (9.6%)	6.7 (1.17)

Table 6.1: Number of tweets and use of emojis in our dataset. The first five rows of the table describe the distribution of tweets across country-language, while the last five rows show the distribution of English tweets from USA across seasons (spring, summer, autumn, winter). First column: million tweets. Tweets W/E is the percentage of tweets including at least one emoji. Third column (Emojis) indicates the million of emojis employed in the tweets (average number of emojis per tweet considering only tweets that include at least one emoji).

In Table 6.2, the 15 most frequent emojis of each country-language pair and each season are shown. We can see that 😂, ❤ and 😍 are the most common emojis in all the considered country-language pairs. ITA and UK are country-language pairs that often use their respective national flag emojis. One of the difference we can see is that the use of 🔥 is very common in USA but not in the other country-language pairs. Also, many of the most used emojis in SPA (that are not frequently adopted in the other country-language pairs) are related to parties (like 🎉, 🍻, 💃).

Regarding the frequency of usage of emojis across different seasons, 😂, ❤ and 😍 are still the most common ones. As expected, we can notice that some emojis are season-specific, like 🎃 in autumn, ❄ and 🎄 in winter, and ☀ in spring and summer.

6.2.2 Repeated and combined use of emojis in a tweet

As we can notice from the last column of Table 6.2, if we consider only tweets with at last one emoji, the average number of emojis per tweet is equal to 1.29: as a consequence a considerable number of tweets (3.7 million) employs more than one emoji. In particular, by manually exploring tweets with more than one emoji, we noticed that if a single emoji is repeated within a single tweet, it is usually

Rank	USA	UK	ESP	ITA	SPR	SUM	AUT	WIN
1	❤️	❤️	❤️	❤️	❤️	❤️	❤️	❤️
2	😍	😍	😍	😍	😍	😍	😂	😂
3	😂	😂	💕	😂	😂	😂	😍	😍
4	🔥	😊	🎉	🇮🇹	🔥	🔥	🔥	🔥
5	💕	👌🏽	👌🏽	😎	💕	☀️	💕	🎄
6	😎	🇬🇧	😂	😊	☀️	😎	💯	❄️
7	💯	👍🏽	💪🏽	☀️	😎	💕	✨	💯
8	💙	✨	😊	🔝	🙌🏼	💙	🙌🏼	💕
9	🙌🏼	💕	😘	😬	💯	✨	💙	😊
10	✨	🎉	🍻	😘	💙	🙌🏼	🎄	🙌🏼
11	😊	🙌🏼	👭🏽	😉	✨	🇺🇸	😊	😘
12	🎉	😘	💃🏽	🎉	😊	💯	🎉	😎
13	😘	☀️	🔝	💙	🎉	🎉	🎃	🎉
14	☀️	💙	😎	✌🏽	😘	💪🏽	😎	✨
15	💪🏽	😎	👏🏽	💪🏽	💪🏽	😊	😘	💪🏽

Table 6.2: The 15 most frequent emojis of the four country-language pairs and the four seasons (by considering only tweets from USA).

employed to stress or empathize the meaning conveyed by that specific emoji (as repeating many times the emoji 👏🏽); on the other hand, two or more different emojis are used jointly in order to convey a sort of "compositional" meaning, for instance, we can use ❤️ 🎄 to communicate the idea of enjoying Christmas time.

We analyzed our Twitter datasets in order to confirm and better ground such intuitions. In particular, we investigated the repeated use of the same emoji by computing the distribution of repetitions of the 15 most frequent emojis inside a single tweet as shown in Figure 6.1. Even if most of the times (between 70% and 80% of the occurrences of an emoji in a tweet) the emoji is not repeated, it is interesting to notice that emojis are repeated twice or three times inside the same tweet between 5% and 25% of their occurrences. In most of the cases the number of tweets in which the same emoji is repeated decreases with the increase of the number of repetitions considered. This behaviour is not followed by the emoji 😂 that has a slightly different repetition pattern (consistent across the four country-language pairs): almost half of the times this emoji is used it is repeated twice or more. In the country-language pairs ITA and USA, the number of times 😂 is employed with three or more repetitions inside a tweet is greater than the number of times the emoji appears twice. Moreover, the majority of the 15 most frequent emojis considered in Figure 6.1 are repeated no more than 4 times. This fact may

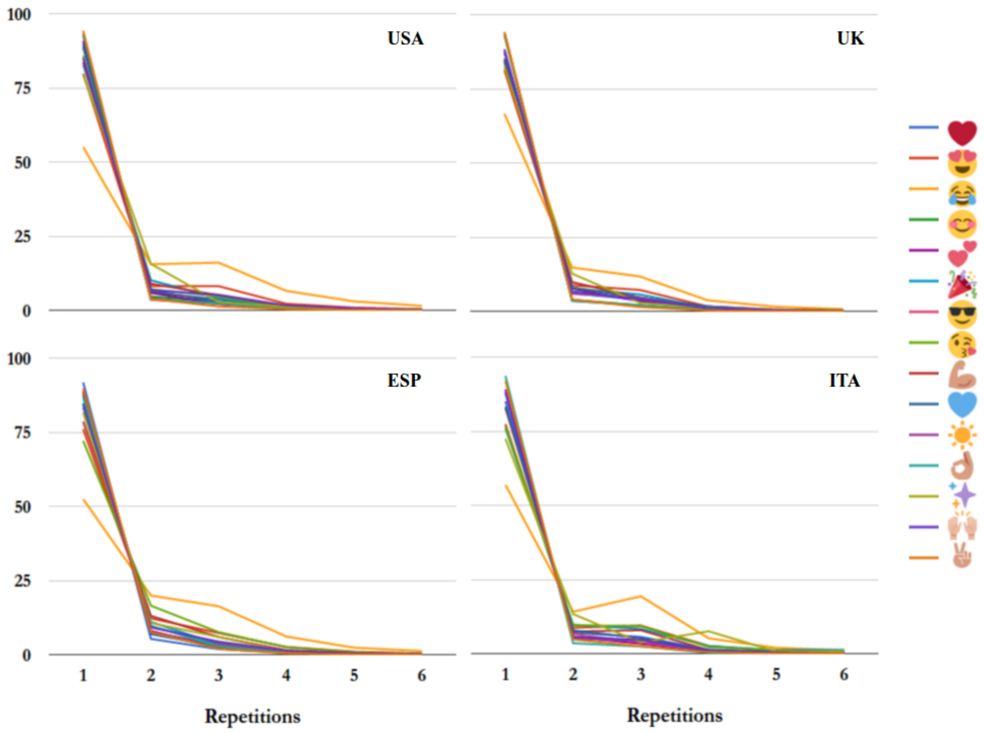


Figure 6.1: Repetition of the 15 most frequent emojis inside the same tweet by considering the four country-language pairs of our dataset. Each line represents the percentage (y-axis) of times a specific emoji is used alone in a tweet (1 repetition, x-axis) or repeated two or more times.

point out that, in order to stress the meaning conveyed by an emoji, two or three repetitions are considered effective. Among the 200 most frequent emojis in our dataset, the most repeated ones are 😂, 😭, 👏🏽, 🔥, and 😱.

Even if not directly addressed by the experiments presented in the rest of this chapter, the phenomenon of repeated use of the same emoji inside a tweet as well as the ’compositional’ meaning conveyed by two or more different emojis employed together deserve a deeper investigation and constitute part of our future venues of research.

6.3 Comparing the semantics of words and emojis: the skip-gram vector model

In order to model the use and the meaning of the emojis we employ the skip-gram vector model introduced by [Mikolov et al., 2013a]. This algorithm allows to convert emojis into vectors, thus representing emojis in a continuous vector space where semantically similar emojis are mapped to nearby points. The skip-gram model is based on the Distributional Hypothesis, which states that words that appear in the same contexts share the same meaning [Harris, 1954, Miller and Charles, 1991]. This approach enables us to compare the semantics of emojis across languages, locations and seasons as described in Section 6.4. We computed embeddings with 300 dimensions by considering a window size of 6 tokens (we previously found out that this configuration is optimal when we need to jointly model words and emojis by means of embeddings [Barbieri et al., 2016b]). Before the computation of the embedding of words and emojis, we pre processed the text of each tweet removing stopwords, punctuation marks (but leaving emoticons like :) or :P), Twitter hashtags and user mentions as they are not directly relevant to determine the semantics of emojis. We also lower-cased each tweet to reduce noise.

We built 8 skip-gram vector models, one for each country-language pair and one for each season. In Figure 6.2 we plot the embeddings of the 200 most popular emojis of the English tweets of our dataset posted from USA, by reducing the embedding dimensions from 300 to two thanks to t-SNE [Maaten and Hinton, 2008]. We can see that similar emojis are plotted one close to the others, like the music-related emojis, food and drinks and nature-related emojis.

6.4 Experimental Framework

We run two types of experiments to compare how the meaning of emojis varies across distinct locations (i.e. country-language pairs) and seasons of the year. In particular, in a first experiment (Nearest-Neighbour experiment, described in Section 6.4.1) we investigate whether the meaning of single emojis is preserved across country-language pairs and seasons. In a second experiment (Similarity-matrix experiment, described in Section 6.4.2), we compare the overall semantic models of the 200 most frequent emojis across different country-language pairs. For the sake of simplicity, we will describe the framework only for the language-location case, but the season related experiments adopt exactly the same scheme.

6.4.1 Nearest-Neighbour experiment

In our first experiment, we quantify to which extent the meaning of an emoji A is preserved across different locations by measuring the overlap across locations of the set of emojis that are most similar to A. We exploit the vector representation of each emoji in a specific location to select the other emojis with similar vectors and thus presumably closest in meaning. We define the Nearest-Neighbours $NN_l(e)$

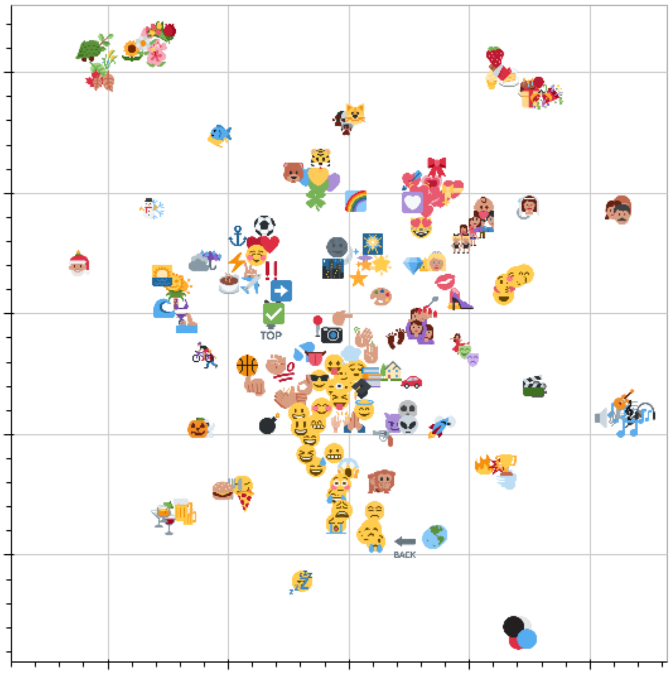


Figure 6.2: Skip-gram embeddings trained on the USA corpus, reduced to two dimensions with t-SNE.

of the emoji e in the location l, as the set of the 10 nearest emojis to the emoji e in the semantic space of location l. We determine the Nearest-Neighbours of each emoji by considering its cosine similarity with other emojis[2].

Note that the semantic vectors (embeddings) are derived from co-occurrence statistics extracted from both emojis and words of the tweets posted in a specific location. However, since in each location an emoji is described by other similar emojis, we are able to compare these representations across different locations.

In order to see if an emoji is semantically similar across a pair of locations, we look at the common elements in the Nearest-Neighbours representation of that emoji in both locations. If the representations of the emoji in different locations share many elements (Nearest-Neighbours) it would mean that the emoji is defined and thus used in a similar way. If there isn't any or there are few common emojis among the two set of Nearest-Neighbours of a specific emoji, such emoji is more likely to mean something different in the two locations. More precisely, to determine if the emoji e has a similar semantics in both locations l_1 and l_2 we

[2] We noticed that most of the times the cosine similarity drops after the 10^{th} closest emoji; for each emoji the cosine similarity of the ten most similar emojis is always greater than 0.4.

measure the size of the intersection of the Nearest-Neighbours sets:

$$sim_{l_1 l_2}(e) = |NN_{l_1}(e) \bigcap NN_{l_2}(e)|$$

We assume that if $sim_{l_1 l_2}$ is equal to 10, the emoji e has the same meaning in both locations l_1 and l_2. On the other hand, if $sim_{l_1 l_2}$ is equal to 0 the emoji means something different in the two locations.

Moreover, we also measure whether an emoji preserves the same meaning across all the locations considered by looking at the overlap of all the sets of emojis that are most similar to the emoji e in each location:

$$sim_{all}(e) = |NN_{l_1}(e) \bigcap NN_{l_2}(e) \bigcap ... \bigcap NN_{l_n}(e)|$$

where n is the number of locations.

6.4.2 Similarity-matrix experiment

In order to globally evaluate if the semantics of pairs of emojis is preserved across different locations, we compute for each location the similarity matrix of the 200 emojis that occur most frequently in our Twitter dataset. The value of each cell of the similarity matrix is equal to the cosine similarity of the corresponding pair of emojis. Once computed the similarity matrices, we can compare a pair of matrices (thus a pair of locations) by evaluating their Pearson's correlation: in this way we can globally quantify to what extent emojis are used with similar semantics across distinct locations. Moreover, we can explore specific differences among pairs of similarity matrices, to see which pairs of emojis have a different semantics across distinct locations.

6.5 Results and discussion

In this Section we present and discuss the results of our experiments aiming at evaluating how the semantics of emojis varies across languages and locations (Section 6.5.1) as well as across seasons (Section 6.5.2). In particular, we describe the outcome of the two experiments described in Section 6.4 by considering the different locations (i.e. country-language pairs) and seasons (i.e. seasons) identified in our Twitter dataset.

6.5.1 Language-and-location analysis of emojis

In this Section we analyze the semantics of emojis across different locations identified by distinct country-language pairs: to this purpose we consider and compare the four vector models built by processing the USA, UK, ESP, and ITA collections of tweets from our Twitter dataset (see Section 6.2 and Table 6.1 for the descriptions of both the dataset and the collections). In particular, we compare the semantics of emojis across locations in Section 6.5.1.1 by relying on the Nearest-Neighbour approach (introduced in Section 6.4.1) and in Section 6.5.1.2 thanks to the Similarity-matrix experimental framework (presented in Section 6.4.2).

6.5.1.1 Nearest-Neighbour experiment for language-and-location analysis

Table 6.3 presents the results of the Nearest-Neighbour experiment carried out by computing for each emoji both the sim_{all} score and the six $sim_{l_1 l_2}$ scores, each one related to one of the six possible combinations of pairs of locations (i.e. country-language pairs). The emojis that seem to have the same meaning independently from the language include those which explicitly refer to music, nature, food or facial expressions. In the bottom half of Table 6.3 we can see the emojis with lowest sim_{all} (all emojis have a sim_{all} score equal to 0): these emojis have meanings that mainly depends on the specific country-language pair where they are used. Looking at the bottom of the table we can see that the emojis 👣 and 🐻 are used in a different way across all the country-language pairs: each country-language pair seems to have its own way to define them. Also the emojis 💯 and 🔝 do not seem to keep their meaning across different country-language pairs even if these two emojis are often used together in Italian Tweets posted from Italy (ITA).

Regarding the clover emoji 🍀, we can see from Table 6.4 that the 10 Nearest-Neighbours are different across the four country-language pairs: USA relates the clover to vegetation emojis, rugby and Ireland (the letter “I”, ℹ️). In UK it is related to Ireland as well, while in Italy the 10 Nearest-Neighbours are flowers, and the closest emoji is 🐞 which means luck (as the clover) to Italian people. Finally the country-language pair that uses the 🍀 emoji in the most peculiar way is represented by the Spanish tweets posted from Spain (SPA), where the 10 Nearest-Neighbours show that this emojis is used in a friendship and love contex. Also the case of 🎀 and 🏆 is interesting, as they are concrete objects but they are exploited to convey different meanings across different country-language pairs.

6.5.1.2 Similarity-matrix experiment for language-and-location analysis

In this section we report the results of the similarity-matrix experiments in which the four similarity matrices of the 200 most frequent emojis of our Twitter dataset (one matrix for each country-language pair) are compared. First of all we analyze the Pearson correlation between the four similarity matrices. Then we evaluate the differences of single elements of these matrices.

	Rank	**USA UK**	**USA ESP**	**USA ITA**	**UK ESP**	**UK ITA**	**ESP ITA**	s_{all}
🎶	25	10	9	9	9	9	9	9
🎤	68	10	9	9	9	9	9	9
🎵	127	10	9	9	9	9	9	9
🎼	131	9	9	9	9	9	9	9
🎸	180	9	9	9	9	9	9	9
👩🏽	147	8	8	8	9	9	8	8
👧🏽	166	8	8	8	8	8	8	8
🐟	175	9	8	8	8	8	9	8
👨🏽	191	9	8	9	8	9	8	8
🍫	192	9	8	9	9	10	9	8
🌸	53	8	8	7	9	7	7	7
🍃	59	8	9	8	9	8	8	7
🍁	66	9	7	7	7	8	8	7
🍂	67	9	7	7	7	8	8	7
🎈	70	8	9	8	8	7	8	7
💯	71	4	0	1	1	2	2	0
👀	77	4	1	1	1	1	1	0
⚡	81	2	2	2	0	2	1	0
🍀	84	6	0	1	0	3	0	0
👋🏼	95	0	3	2	1	1	3	0
🎀	99	6	1	2	3	3	2	0
🏆	111	3	2	3	6	5	5	0
🔝	116	4	2	0	2	2	3	0
🌈	118	2	0	1	2	3	3	0
😬	145	6	1	1	1	0	3	0
💭	168	1	2	2	0	0	1	0
🚀	174	3	3	1	1	1	2	0
📍	176	0	2	0	0	1	3	0
👣	177	3	0	3	0	3	1	0
🐻	185	3	2	0	7	6	6	0

Table 6.3: Nearest-Neighbour experiment with language-and-location datasets: once ordered the 200 most used emojis of our Twitter dataset with respect to their sim_{all} score (indicated as s_{all} in the table)

Correlation between country-language pairs. We take advantage of the similarity matrices to analyze whether the semantics of emojis is defined similarly across two country-language pairs. In particular, we report in Table 6.6 the Pearson's correlations of the similarity matrices of the four country-language pairs we included in our Twitter dataset (USA, UK, SPA and ITA). We can notice that most of the country-language pairs are strongly correlated to each other. This is an interesting finding: the semantics of the emojis analyzed is in some way preserved across the four country-language pairs even if the vocabularies of the languages and the words that occur next to each emoji (that are exploited to build the embedding of the same emoji) are different. The average correlation of the four country-language pairs is height and each correlation value is always equal or greater than

	1	2	3	4	5	6	7	8	9	10
USA	💚	💛	🏈	🌿	🌱	ℹ️	🌾	🌻	🍃	🌰
UK	ℹ️	🌿	💚	🌱	🌵	🎍	🍃	🌳	🌾	🌷
ESP	🔐	👭	👫	🔒	💘	💑	💞	💏	💕	💫
ITA	🐞	🌹	🌺	🌻	🌼	💐	🌷	🎍	🚀	🌳

Table 6.4: The 10 Nearest-Neighbour emojis of the clover symbol in the semantic spaces of the four country-language pairs. Unlike the other country-language pairs, in Spanish tweets posted from Spain the clover is used in a love/friendship context.

	USA	**UK**		**USA**	**ESP**		**USA**	**ITA**		**UK**	**ESP**		**UK**	**ITA**		**ESP**	**ITA**
☕🍰	0.35	0.67	👭🍀	0.33	0.76	👍🔝	0.32	0.58	👭🍀	0.21	0.76	😎💫	0.69	0.13	👭🍀	0.76	0.31
🐾💝	0.21	0.55	🍀👫	0.32	0.75	⚽🔵	0.41	0.63	💘🍀	0.16	0.65	⚡🐯	0.18	0.56	🍀👫	0.75	0.34
😎💫	0.42	0.69	💘🍀	0.27	0.65	⚡🐯	0.3	0.56	🍀👫	0.29	0.75	😎🌟	0.57	0.08	💭👣	0.68	0.28
⚽🔴	0.41	0.66	🍀💑	0.29	0.64	🍴🍕	0.43	0.63	🍀💎	0.08	0.56	😉⚡	0.11	0.46	😳😚	0.44	0.05
⚽🔵	0.41	0.66	💥👅	0.34	0.66	💯🔝	0.43	0.63	🌊👙	0.42	0.81	💜😇	0.57	0.12	💃🍻	0.68	0.3
💐🚀	0.09	0.4	🍀💝	0.22	0.57	💣🔊	0.38	0.59	👑🍀	0.17	0.58	🌴🏊	0.31	0.6	🎉🍻	0.7	0.32
💙🐻	0.76	0.27	⚽🔵	0.41	0.69	💣🚀	0.36	0.58	💞🍀	0.21	0.61	💙😀	0.48	0.07	😳🙆	0.62	0.24
🌺🌻	0.76	0.89	🍻🔝	0.21	0.55	🌊⚓	0.5	0.66	💥✋	0.15	0.57	🎀🏡	0.16	0.46	🍻💘	0.49	0.12
⚽🐻	0.56	0.13	💨🚗	0.51	0.75	🎤🎸	0.81	0.85	🍀💑	0.25	0.64	🌴⚓	0.27	0.55	⚡🐯	0.21	0.56
🌟☁️	0.54	0.11	💥✋	0.24	0.57	⚽⚪	0.52	0.66	💕🍀	0.19	0.59	🌅⚓	0.23	0.51	☁️🚀	0.5	0.14
💥💨	0.7	0.24	💨✊	0.32	0.68	😍🔝	0.13	0.41	🌊🌴	0.46	0.81	☀️🌊	0.44	0.68	😃😝	0.15	0.49
⚫🐻	0.47	0.06	👌🔝	0.38	0.67	🍻🍕	0.3	0.51	💫👣	0.24	0.61	🌊🌴	0.46	0.69	🎉🍹	0.67	0.31
🍁🌿	0.61	0.77	💭👣	0.42	0.68	💛🐯	0.68	0.12	🌟😴	0.02	0.44	🏊⚓	0.28	0.54	🍻👯	0.63	0.27
🎉💨	0.56	0.14	🎉🍸	0.44	0.69	⚫🔵	0.82	0.84	💫💭	0.35	0.7	🌊🏊	0.43	0.66	💫💭	0.7	0.34
😍🌟	0.26	0.49	📷🏀	0.25	0.57	🎉🎓	0.39	0.56	🍀👸	0.1	0.48	☀️⚓	0.46	0.68	🍻👅	0.63	0.28

Table 6.5: Similarity-matrix experiment with language-and-location datasets

	USA	UK	ESP	ITA	AVG
USA	1	**0.802**	0.694	0.66	0.719
UK	**0.802**	1	0.741	0.734	0.723
ESP	0.694	0.741	1	0.737	0.713
ITA	0.66	0.734	0.737	1	0.71

Table 6.6: Similarity-matrix experiment with language-and-location datasets: pairwise Pearson's correlation between the similarity matrices of the four country-language pairs.

0.71. However, the correlation values present sensible variations across different country-language pairs: the strongest correlation is between USA and UK (0.802), probably because both country-language pairs share similar vocabularies (American and British English respectively), and the weakest is between USA and ITA (0.66). On the other hand, Italian tweets posted from Italy have an high correlation with Spanish tweets posted from Spain (0.737) and English tweets posted from United Kingdom (0.734). Spanish tweets from Spain highly correlate with British English tweets (UK, 0.741) and Italian tweets (IT, 0.737).

Differences in the use of emojis across country-language pairs. We observed that in most of the cases the semantics of emojis is somehow preserved across the four country-language pairs we considered (USA, UK, SPA and ITA). Nevertheless we can spot some interesting difference in the language-specific use of these small images. In particular, in this Section we explore the disagreement in the similarity matrices of the four country-language pairs using a method similar to [Kriegeskorte et al., 2008]: we analyze couples of emojis that have different similarities across two country-language pairs (for instance, two emojis can be strongly similar in a country-language pair, but can convey different meanings in another country-language pair). Table 6.5 shows the similarity matrix scores of couples of emojis for all the possible combinations of country-language pairs. We report all the couples of emojis which are semantically related in one country-language pair (e.g. USA) but unrelated in the other (e.g. UK).

In the first place, from the first column (USA - UK) of Table 6.5 we can notice the different similarity scores of the cup ☕ with the cake 🍰. These two emojis convey similar meanings in UK (similarity score equal to 0.67) but have a distinct semantics in USA (similarity score equal to 0.35): this behaviour can be related to the fact that the cake emoji in USA is used as a birthday cake while in UK as a cake to eat with tea.

The ⚽ in USA is not used together with emojis representing colors (like red 🔴, blue 🔵 and white ⚪) while in UK and ITA it is. In Italy the football emoji is associated with both blue and white, and looking at the tweets that include these

	Rank	**SPR**	**SPR**	**SPR**	**SUM**	**SUM**	**AUT**	s_{all}
		SUM	**AUT**	**WIN**	**AUT**	**WIN**	**WIN**	
🎶	32	9	10	9	9	10	9	9
🎤	56	9	10	9	9	9	9	9
🎼	127	9	10	9	9	10	9	9
🍦	133	9	9	10	10	9	9	9
🎸	142	9	9	9	9	9	9	9
🌙	150	9	9	9	10	10	10	9
🎵	155	10	10	9	10	9	9	9
🐠	167	10	9	9	9	9	9	9
🐟	189	9	9	9	9	9	9	9
🍭	195	10	10	9	10	9	9	9
😂	2	8	8	8	9	8	8	8
💖	39	8	8	9	8	8	9	8
😏	47	8	8	8	8	8	9	8
💓	112	8	8	9	9	8	9	8
🍰	184	8	10	10	8	8	10	8
🚀	160	5	3	4	3	4	5	2
💣	179	5	4	3	5	4	3	2
🍀	180	3	4	3	5	8	5	2
🏀	53	6	5	4	4	2	5	1
🚨	65	2	3	4	3	4	3	1
🏆	70	6	6	4	5	4	3	1
⚡	84	6	4	3	6	4	2	1
❗	95	2	5	3	1	2	4	1
⭐	132	7	4	5	3	5	4	1
🎓	136	2	3	1	2	5	2	1
➡️	198	6	5	5	6	5	3	1
🌲	100	8	3	1	3	1	7	0
🎨	116	3	3	1	1	2	1	0
🌈	135	2	1	2	5	2	3	0
📍	163	3	1	2	2	0	0	0

Table 6.7: Nearest-Neighbour experiment with season-based datasets: once ordered the 200 most used emojis of our Twitter dataset with respect to their sim_{all} score (indicated as s_{all} in the table).

three emojis, we find out that are about Naples and Lazio football teams since these are the main colors of their uniforms. Another difference between USA and ITA is the pizza emoji 🍕, as in Italy, differently from USA, this emoji is associated to 🍴 and 🍺 as it is typical to eat pizza with fork and knife while drinking beer.

Regarding USA and ESP the main difference is the clover emoji 🍀. As we have observed in the previous section, in Spanish, differently from USA, this emoji is used in a context of love and friendship. Another difference between SPA and the other three country-language pairs concerns the alcoholic drinks emojis that in Spanish are often associated to the emojis 🔝, 🎉 (USA-SPA), 💃 and 👯 (SPA-ITA). On the semantics of the drink-related emojis SPA agrees more with UK than with USA and ITA.

	1	2	3	4	5	6	7	8	9	10
SPR	🌳	🍃	🍂	🌾	🌿	🍁	🌱	🗻	🌄	⛺
SUM	🌳	🌾	🌿	🍃	⛺	🌄	🍄	🌱	🌻	🗻
AUT	🎄	🎅🏽	🎁	⛄	🌳	❄️	🍃	❤️	🌿	💚
WIN	🎄	⛄	🎅🏽	🎁	❄️	❤️	💚	⛺	✨	☺️

Table 6.8: The 10 Nearest-Neighbour emojis of the pine emoji computed with respect to the semantic spaces of the four seasons.

One last interesting difference can be spotted between UK and ITA. The swimmer emoji 🏊🏽 and the anchor emoji ⚓ are associated to beach-related emojis in ITA (like 🌴, 🌊, 🌅, ☀️) while in UK they are not. This suggests that in UK swimming is not an activity related to beach and sun (considering the weather of the country), but it is probably mainly practised in swimming pools. Similarly, also sailing does not remind of nice weather and palms in UK.

6.5.2 Season-based analysis of emojis

In this Section we study the semantics of emojis across different time-frames identified by distinct season of the year: to this purpose we consider and compare the four vector models built by processing English tweets posted from USA in spring, summer, autumn and winter (see Section 6.2 and Table 6.1 for the description of these collections of tweets). In particular, we analyze if and how the semantics of emoji changes across seasons in Section 6.5.2.1 by relying on the Nearest-Neighbour approach (introduced in Section 6.4.1) and in Section 6.5.2.2 thanks to the Similarity-matrix experimental framework (presented in Section 6.4.2).

6.5.2.1 Nearest-Neighbours experiment for season-based analysis

In this Section we present and discuss the outcome of the Nearest-Neighbours experiment (see Section 6.4.1): by considering the season-specific vector models, for each emoji we computed the 10 Nearest-Neighbours emojis in each season. In this way we can investigate if a specific emoji shares the same set of Nearest-Neighbours across distinct seasons and thus if that emoji preserves its meaning across seasons. The results of the Nearest-Neighbour experiment are shown in Table 6.7.

As it happens when we compare distinct country-language pairs (see Table 6.3), the meaning of the emojis related to the music domain remains the same across different seasons of the year. Also, the laughing 😂, the blink 😏, and love-related

	SPR	SUM	AUT	WIN	AVG
SPR	1	**0.871**	0.839	0.837	0.849
SUM	0.871	1	0.86	0.84	0.857
AUT	0.839	0.86	1	0.849	0.849
WIN	0.837	0.84	0.849	1	0.842

Table 6.9: Similarity-matrix experiment with season-based datasets: pairwise Pearson's correlation between the similarity matrices of the four seasons.

emojis seem to preserve their meaning when posted in different seasons of the year. Also sweets-related emojis like 🍦, 🍭, and 🍰 are characterized by the same set of Nearest-Neighbours in the four season-specific vector models.

Accordingly to the Nearest-Neighbours experiment, the emojis that convey a meaning that varies the most with the season are the ones listed in the bottom half of Table 6.7. Among these emojis there is some related to sport (like 🏀 and 🏆) as there are seasons of the year that are richer of sportive events that other periods.

Also the emoji 🎓, used in a University context, has a meaning that is not preserved across seasons. The Nearest-Neighbours of this emojis are party and heart-related emojis in spring, while books-related emojis and also a gun emoji in autumn.

Another interesting example concerning the semantics of the 🌲 across seasons is described in Table 6.8 where the 10 Nearest-Neighbours of 🌲 for each season are shown. The pine is used as vegetation, camping and sunrise-related emoji in spring and summer, while in autumn and winter it is used as a Christmas-related emoji. The same behaviour happens if we consider the emoji 🎁: in spring and summer the three emojis closest in meaning and usageare 🎈, 🎊 and 🎂, while in winter the three closet emojis are 🎅, 🎄, 🌲.

6.5.2.2 Similarity-matrix experiment for season-based analysis

In this section we show the results for the Similarity-matrix experiment (see Section 6.4.2). Some interesting difference in the use of emojis can be spotted by analyzing and comparing the four season-specific similarity matrices.

Correlation between seasons. As we can notice from Table 6.9, all the season-specific similarity matrices are strongly correlated presenting a Pearson Correlation coefficient always equal or greater 0.83 for every pair of seasons. spring and summer show the highest correlation (0.87). The semantics of emojis in summer is also highly similar to the semantics in autumn (0.860) (a bit less if we consider the meaning of emojis in winter, 0.840). winter show the lowest correlations with other seasons, suggesting that winter is the season where the emojis undergo

		SP	SU			SP	AU			SP	WI			SU	AU			SU	WI			AU	WI
❗	💤	0.03	0.41	🎄	🎁	0.11	0.67	🎄	🎁	0.11	0.75	🎄	🎁	0.2	0.67	🎄	🎁	0.2	0.75	😄	🔴	0.45	0.03
👉🏽	💤	0.47	0.13	🏆	💉	0.31	0.7	❄️	🎁	0.12	0.56	😄	🔴	0.1	0.45	💨	🏆	0.73	0.32	🎬	☝🏽	0.66	0.3
🌟	🍑	0.23	0.55	❄️	🎁	0.12	0.5	🎉	🏆	0.7	0.27	🎁	🎊	0.89	0.55	🇺🇸	💣	0.12	0.51	❤️	💀	0.09	0.42
👋🏽	✅	0.5	0.18	😄	🔴	0.07	0.45	💨	🏆	0.73	0.32	🎬	☝🏽	0.37	0.66	🎉	💨	0.56	0.16	⚾	💀	0.05	0.37
🎈	❗	0.2	0.51	🎬	☝🏽	0.3	0.66	🎄	🍃	0.85	0.45	🎊	☝🏽	0.21	0.51	🏆	💉	0.23	0.61	💎	❗	0.16	0.48
🚨	🎵	0.14	0.45	🍴	➡️	0.46	0.08	🔥	💨	0.67	0.27	🔫	🎓	0.17	0.46	🎉	🏆	0.66	0.27	🌃	🍎	0.21	0.52
🍴	➡️	0.46	0.15	🔊	🚗	0.23	0.58	✨	🎁	0.3	0.66	🎧	😑	0.09	0.37	🎈	❗	0.51	0.14	💀	❗	0.09	0.4
😁	🍷	0.29	0.59	❗	💤	0.03	0.37	🎉	💨	0.53	0.16	🎤	💍	0.15	0.43	❄️	🎄	0.34	0.69	😂	👏🏽	0.58	0.28
😃	⚡	0.17	0.46	🎧	🚗	0.27	0.59	👑	🐝	0.66	0.3	🎧	🚗	0.23	0.59	💚	🎁	0.26	0.59	☺️	💀	0.17	0.47
❗	😴	0.07	0.36	💄	👙	0.28	0.57	🎄	🎀	0.21	0.53	🌟	🍑	0.55	0.22	💎	❗	0.15	0.48	✨	🗽	0.18	0.48
💛	🏈	0.48	0.76	👑	🐝	0.66	0.34	❄️	🎄	0.37	0.69	😂	💔	0.49	0.2	😛	😃	0.37	0.69	🚨	🎵	0.4	0.11
😏	📚	0.35	0.38	🎄	🍃	0.85	0.53	💚	🎁	0.27	0.59	🔊	🚗	0.29	0.58	🔑	📍	0.67	0.32	☕	💀	0.06	0.35
🎉	💎	0.29	0.31	🎊	☝🏽	0.23	0.51	💋	🎄	0.16	0.48	🏆	💉	0.23	0.7	✨	🎁	0.34	0.66	🔥	💨	0.55	0.27
🍸	🏃	0.21	0.23	💰	🎊	0.15	0.43	💥	💨	0.69	0.34	💅	😚	0.42	0.38	🚨	🎵	0.45	0.11	🇺🇸	💣	0.22	0.51
🌞	🎸	0.27	0.28	🍴	❗	0.46	0.16	🇺🇸	🗽	0.44	0.75	🔥	🌻	0.14	0.14	☺️	💀	0.16	0.47	💕	😫	0.48	0.21

Table 6.10: Similarity-matrix experiment with season-based datasets: for every combination of two seasons, the couples of emojis with biggest difference among their season specific similarity values are shown. SP: spring, SU: summer, AU: autumn, WI: winter.

substantial changes of their semantics.

Differences in the use of emojis across seasons. We can spot several examples of emojis that change their meaning across different seasons including for instance the pine tree as described in the previous Section (6.5.2.1).

Table 6.10 shows the differences in the meaning of emojis across seasons as spotted by comparing the season-specific similarity matrices. In particular, for each pair of seasons this Table shows the couples of emojis that are characterized by the highest difference in similarity across those seasons. The first column on the left shows the differences between spring and summer, that seem difficult to be interpreted, also because the differences are not as high as in the other cases. For example is not easy to explain why the emoji 💤 is similar to ❗ in summer but not in spring, while 💤 is close to 👉 in spring but not in summer. The cup and syringe emojis are closer in autumn than in spring, probably for a case of doping in sport discovered in that season of our Twitter dataset. The most different emojis in spring and autumn are the one related to birthday and Christmas, as the gift 🎁 points out mainly a Christmas gift in autumn and winter (close to ❄ and 🌲 for instance). The case of the couple of emojis 🎓 and 🔫 is also interesting as these two emojis are close in autumn than in summer, suggesting that in autumn students have harder times at school. One of the emojis that seems to be used differently in autumn and winter is the Skull 💀, probably because this emoji is employed to point out Halloween-related stuff in autumn.

6.6 Conclusions

In this chapter we have explored if and how the meaning and usage of emojis varies across languages and locations (by analyzing English tweets posted from United States of America, English tweets from United Kingdom, Spanish tweets from Spain, and Italian tweets from Italy) as well as across seasons (by analyzing tweets posted in spring, summer, autumn and winter). We used distributional semantic models to represent the meaning of the emojis in each language, location and season respectively. Then we compared the semantics of emojis across languages, locations and seasons by means of two experimental approaches relying respectively on the analysis of Nearest-Neighbor emojis and on the comparison of similarity matrices of emojis.

Our results suggest that even if the overall emoji semantics of the languages we studied is similar, we can identify some emojis that are not used in the same way from language to language: this fact may be related to the cultural differences that exist between countries. For instance, the clover emoji 🍀 is used in a friendship and love context in Spain, while in the other countries is mainly used in relation to luck and the symbol of Ireland.

Regarding the variations of the meaning of emojis across seasons (studied by considering only English tweets posted from the United States of America), we figured out that even if most of the emojis preserve their semantics, specific differences can be identified. Two examples are the gift 🎁 and the pine 🌲 emojis that in winter are used as Christmas-related emojis, but in spring and summer are used to respectively point out a birthday present and a tree.

Knowing how emojis are used in different countries and time of the year is valuable for various NLP applications. We can think at search query expansions, for example, if a user uses a gift emoji in winter, we know that he may not be expressing the same things than in summer. These results can also be a starting point for more structured social studies on emoji usage.

Chapter 7

EMOJI DETECTION

In the previous chapter we explored the semantic of emojis with unsupervised systems, and shown that emojis are very subjective. In this chapter we explore if, from a text, we can identify the emoji which is most appropriate for it. This seems a difficult task since the emojis can be ambiguous, but we will show that it is possible to predict them with reasonable accuracy. Moreover, we will also put forward multimodal experiments, taking into account text and images which accompany social media posts. Part of the experiments presented in this chapter have been published in [Barbieri et al., 2017].

7.1 Introduction

Despite its status as a language form, emojis have been so far scarcely studied from a Natural Language Processing (NLP) standpoint, apart from a few notable exceptions described in Section 2.2.2. However, the interplay between text-based messages and emojis was virtually unexplored. In this chapter we aim to fill this gap by investigating the relation between words and emojis, studying the problem of predicting which emojis are evoked by text-based tweet messages.

[Miller et al., 2016] performed an evaluation asking human annotators the meaning of emojis, and the sentiment they evoke. People do not always have the same understanding of emojis, indeed, there seems to exist multiple interpretations of their meaning beyond their designer's intent or the physical object they evoke[1]. Their main conclusion was that emojis can lead to misunderstandings. The ambiguity of emojis raises an interesting question in human-computer interaction: how can we teach an artificial agent to correctly interpret and recognize

[1] https://www.washingtonpost.com/news/the-intersect/wp/2016/02/19/the-secret-meanings-of-emoji/

emojis' use in spontaneous conversations?[2] The main motivation of our research is that an artificial intelligence system that is able to predict emojis could contribute to a better natural language understanding [Novak et al., 2015] and thus to different natural language processing tasks such as generating emoji-enriched social media content, enhance emotion/sentiment analysis systems, and improve retrieval of social network material. In this chapter, we employ a state of the art classification framework to automatically predict the most likely emoji associated to a Twitter text message. We also extend the task to Instagram multimodal posts composed by an image and a textual caption. We show that emojis can be predicted by using the text, but also using a vector representation of the picture. Our main findings are two. The first one is that given a text, we can predict with good accuracy the emoji, even if emojis are very subjective and they are not used in the same way by different people (as we have seen in the previous chapter). The second finding is that incorporating two modalities (text and images) in a combined model improves the emoji prediction accuracy. This result demonstrates that text and images encode different information on the use of emojis and they should be used in a complementary way. We also discuss some initial experiments useful to explore the part of images that most influence the automated prediction of the most likely emoji to associate to that picture.

The chapter is organized as follows, in Section 7.2 we describe the two datasets (Twitter and Instagram posts) we use to experiment the emoji prediction task. Section 7.3 presents into details the approaches we exploit to model the textual and visual information of Twitter and Instagram posts. In Section7.4, we discuss the results of the textual emoji prediction and also perform an human evaluation, to see how humans perform on the same emoji prediction task. In Section7.5 we explore how emoji prediction accuracy improves when we rely on both textual and visual inputs (multimodal prediction) as well as the presence of parts of Instagram pictures that are particularly relevant to predict specific emojis. Section 7.6 concludes this chapter with a brief summary and some remarks on future research.

7.2 Dataset and Task

In this section we present first the two dataset we employ, and later we describe the tasks we propose.

7.2.1 Dataset

As said in the introduction, we experiment emoji prediction on two different dataset. One dataset is composed of Twitter textual data, and the other one in-

[2] http://www.dailydot.com/debug/emoji-miscommunicate/

clude visual and textual content of Instagram posts.

Twitter: We retrieved 40 million tweets with the Twitter APIs[3]. Tweets were posted between October 2015 and May 2016 geo-localized in the United States of America. Tweet texts were preprocessed with a modified version of the CMU Tweet Twokenizer [Gimpel et al., 2011], where we changed several regular expressions and added a Twitter emojis vocabulary to better tokenize the tweets and extract the emojis. We also remove the five modifiers of the skin color[4]. We removed all hyperlinks from each tweet, and lowercased all textual content in order to reduce noise and sparsity. From the dataset, we selected tweets which include *one and only one* of the 20 most frequent emojis, resulting in a final dataset composed of 584,600 tweets. In the experiments we also consider the subsets of the 10 (502,700 tweets) and 5 most frequent emojis (341,500 tweets). The 20 most frequent emoji of the Twitter dataset can be found in Table 7.2.

Instagram: We gathered Instagram posts published between July 2016 and October 2016, and geo-localized in the United States of America. Each Instagram post is made of a picture together with the textual comment of the user who published that picture. In our experiments we considered only posts that contained, beside the photo, a user comment (which is like a caption of the photo) that include a minimum of 4 words (exluding emojis). We preprocess the Instagram texts in the same way as the tweets. As in the Twitter dataset we considered only the posts which include *one and only one* of the 20 most frequent emojis (the most frequent emojis of the Instagram dataset are shown in Table 7.5). As a consequence, our dataset is composed of 299,809 posts, each containing a picture, the user comment and only one emoji. Like in the Twitter data, we will consider the subsets of the 10 (238,646 posts) and 5 most frequent emojis (184,044 posts).

7.2.2 Task

We cast the emoji prediction problem as a classification task: given an image or a text (or both inputs in the multimodal scenario) we select the most likely emoji that could be added to (thus used to label) such contents.

Twitter: In the unimodal experiments, we remove the emoji from the sequence of tokens of the tweet, and use it as a label both for training and testing. The task for our machine learning models is to predict the single emoji that appears in the input tweet.

[3] https://dev.twitter.com

[4] These Unicode charaters indicate the skin tone of an emoji, like in ✌🏻 and ✌🏿 are used different skin modifiers but the base emoji is the same. The Unicode definition can be found at http://unicode.org/reports/tr51/#Diversity

Instagram: We extend the Twitter experimental scheme, by considering also visual information when modeling posts. In particular, for each Instagram post of our dataset, we remove the emoji from the textual comment and use it as a label both for training and testing. The task for our machine learning models is, given the visual and textual content of a post, to predict the single emoji that appears in the input comment.

In the emoji prediction experiments that will be discussed in this chapter we do not consider the position of the predicted emoji inside the textual content of the posts, leaving the exploration of this aspect as future work.

7.3 Models

We present and motivate the models that we use to predict emojis given Twitter or Instagram posts. To model the textual content we use different models, including Bidirectional LSTMs, FastText classifier [Joulin et al., 2017], and also simpler model as Bag of Words and another baseline based on word embeddings of the words.

For image classification we train a state of the art system for image recognition: Deep Residual Neural Networks [He et al., 2016] (Section 7.3.2). We fine-tune a ResNet pre-trained on the Imagnet dataset [Deng et al., 2009b].

7.3.1 Textual Models

7.3.1.1 Bi-Directional LSTMs

Given the proven effectiveness and the impact of recurrent neural networks in different tasks [Chung et al., 2014, Vinyals et al., 2015a, Dzmitry et al., 2014, Dyer et al., 2015, Lample et al., 2016, Wang et al., 2016, inter-alia], which also includes modeling of tweets [Dhingra et al., 2016], our emoji prediction model is based on bi-directional Long Short-term Memory Networks [Hochreiter and Schmidhuber, 1997, Graves and Schmidhuber, 2005]. The input of an RNN is a sequence of vectors. (x_1, x_2, ..., x_n) and it returns another sequence (h_1, h_2, ..., h_n) which is the learned encoded vector. Simple RNNs have a bias, called the vanishing gradient problem, towards learning the most recent input, a solution to avoid this issue was presented in [Hochreiter and Schmidhuber, 1997]: Long Short-term Memory Networks (LSTMs). LSTMs incorporate a memory cell to avoid the vanishing gradient problem. The model [5] is based on bidirectional LSTMs[Graves and

[5] Implemented using Dynet https://github.com/clab/dynet

Schmidhuber, 2005].[6]. The forward LSTM reads the tweet from left to right and the backward one reads it in the reverse direction.[7] The learned vector of each LSTM, is passed through a component-wise rectified linear unit (ReLU) non-linearity [Glorot et al., 2011]; finally, an affine transformation of these learned vectors is passed to a softmax layer to give a distribution over the list of emojis that may be predicted given the tweet. More formally, the message representation, which we write $\mathbf{s}$, is defined as follows: The B-LSTM can be formalized as follows:

$$\mathbf{s} = \max \{\mathbf{0}, \mathbf{W}[\mathbf{fw}; \mathbf{bw}] + \mathbf{d}\}$$

where $\mathbf{W}$ is a learned parameter matrix, $\mathbf{fw}$ is the forward LSTM encoding of the message, $\mathbf{bw}$ is the backward LSTM encoding of the message, and $\mathbf{d}$ is a bias term, then passed through a component-wise ReLU. The vector $\mathbf{s}$ is then used to compute the probability distribution of the emojis given the message as:

$$p(e \mid \mathbf{s}) = \frac{\exp\left(\mathbf{g}_e^\top \mathbf{s} + q_e\right)}{\sum_{e' \in \mathcal{E}} \exp\left(\mathbf{g}_{e'}^\top \mathbf{s} + q_{e'}\right)}$$

where $\mathbf{g}_{e'}$ is a column vector representing the (output) embedding[8] of the emoji e, and q_e is a bias term for the emoji e. The set $\mathcal{E}$ represents the list of emojis. The loss/objective function the network aims to minimize is the following:

$$Loss = -log(p(e_m \mid \mathbf{s}))$$

where m is a tweet of the training set $\mathcal{T}$, $\mathbf{s}$ is the encoded vector representation of the tweet and e_m is the emoji contained in the tweet m. The inputs of the LSTMs are word embeddings[9]. We use two alternative representations of the words included in the tweet.

Word Representations: We generate word embeddings which are learned together with the updates to the model. We stochastically replace (with p = 0.5) each word that occurs only once in the training data with a fixed represenation (out-of-vocabulary words vector). When we use pre-trained word embeddings, these are concatenated with the learned vector representations obtaining a final representation for each word type. This is similar to the treatment of word embeddings by [Dyer et al., 2015].

Character-based Representations: We compute character-based continuous-space vector embeddings [Ling et al., 2015, Ballesteros et al., 2015] of the tokens in each

[6] Bidirectional LSTMs are (recently) found to be useful in different tasks [Dyer et al., 2016, Lample et al., 2016, Wang et al., 2016, Plank et al., 2016]

[7] LSTM hidden states are of size 100, and each LSTM has two layers.

[8] The output embeddings of the emojis have 100 dimensions.

[9] 100 dimensions.

tweet using, again, bidirectional LSTMs. The character-based approach learns representations for words that are orthographically similar, thus, they should be able to handle different alternatives of the same word type occurring in social media.

7.3.1.2 FastText

Fastext [Joulin et al., 2017] is a linear model for text classification. We decided to employ FastText as it has been shown that on specific classification tasks, it can achieve competitive results, comparable to complex neural classifiers (RNNs and CNNs). The best feature of FastText is the speed as it can be much faster than complex neural models. Thus, FastText represents a valid approach when dealing with Social Media content classification, where big amounts of data need to be processed and new, relevant information is continuously generated. The FastText algorithm is similar to the CBOW algorithm [Mikolov et al., 2013a], where the middle word is replaced by the label, in our case the emoji. Given a set of N documents, the loss that the model attempts to minimize is the negative log-likelihood over the labels (in our case, the emojis):

$$loss = -\frac{1}{N}\sum_{n=1}^{N} e_n \log(softmax(BA_{x_n}))$$

where e_n is the emoji included in the n-th Instagram post, represented as hot vector, and used as label. A and B are affine transformations (weight matrices), and x_n is the unit vector of the bag of features of the n-th document (comment). The bag of features is the average of the input words, represented as vectors with a look-up table.

7.3.1.3 Skip-Gram Vector Average

We train a Skip-gram model [Mikolov et al., 2013b] learned from 65M Tweets (where testing instances have been removed) to learn Twitter semantic vectors. Then, we build a model which represents each message as the average of the vectors corresponding to each token of the tweet. Formally, each message m is represented with the vector V_m:

$$Vm = \frac{\sum_{t \in T_m} S_t}{|T_m|}$$

Where T_m are the set of tokens included in the message m, S_t is the vector of token t in the Skip-gram model, and $|T_m|$ is the number of tokens in m. After obtaining a representation of each message, we train a L2-regularized logistic regression, (with ε equal to 0.001).

7.3.1.4 Bag of Words

We applied a bag of words classifier as baseline, since it has been successfully employed in several classification tasks, like sentiment analysis and topic modeling [Wallach, 2006, Blei, 2012, Titov and McDonald, 2008, Maas et al., 2011, Davidov et al., 2010]. We represent each message with a vector of the most informative tokens (punctuation marks included) selected using term frequency−inverse document frequency (TF-IDF). We employ a L2-regularized logistic regression classifier to make the predictions.

7.3.2 Visual Models

Deep Residual Networks (ResNets) [He et al., 2016] are Convolutional Neural Networks that outperformed state of the art systems in several image classification tasks ([Russakovsky et al., 2015a, Lin et al., 2014]) and showed to be one of the best CNN architectures for image recognition. ResNet is a feed-forward CNN that exploits "residual learning", by bypassing two or more convolution layers (like similar previous approaches [Sermanet and LeCun, 2011]). ResNet architecture allows to create very deep networks since they help to alleviate the underfitting problem of traditional networks, caused by optimization difficulties when adding a certain number of layers to the network. We use an implementation[10] of the original ResNet where the scale and aspect ratio augmentation are from [Szegedy et al., 2015], the photometric distortions from [Howard, 2013] and weight decay is applied to all weights and biases (instead of only weights of the convolution layers). The network we used is composed of 101 layers (ResNet-101), initialized with pre-trained parameters learned on ImageNet[Deng et al., 2009b]. We use this model as a starting point to finetune the ResNet on our emoji classification task.

7.4 Unimodal Twitter Experiments

In order to study the relation between words and emojis, we performed two different experiments. In the first experiment, we compare our machine learning models, and in the second experiment, we pick the best performing system and compare it against humans.

7.4.1 First Textual Experiment

This experiment is a classification task, where in each tweet the unique emoji is removed and used as a label for the entire tweet. We use three datasets, each con-

[10] https://github.com/facebook/fb.resnet.torch/

	5			**10**			**20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BOW	.59	.60	.58	.43	.46	.41	.32	.34	.29
AVG	.60	.60	.57	.44	.47	.40	.34	.36	.29
FT	.61	.62	.61	.47	.49	.46	.38	**.39**	**.36**
W	.59	.59	.59	.46	.46	.46	.35	.36	.33
C	.61	.61	.61	.44	.44	.44	.36	.37	.32
W+P	.61	.61	.61	.45	.45	.45	.34	.36	.32
C+P	**.63**	**.63**	**.63**	**.48**	**.47**	**.47**	**.42**	**.39**	.34

Table 7.1: Results of 5, 10 and 20 emojis. Precision, Recall, F-measure. BOW is bag of words, AVG is the Skipgram Average model, FT is FastText, C refers to char-BLSTM and W refers to word-BLSTM. +P refers to pre-trained embeddings.

taining the 5, 10 and 20 most frequent emojis (see Section 7.2.1). We analyze the performance of the five models described in Section 7.3: a bag of words model, a Bidirectional LSTM model with character-based representations (char-BLSTM), a Bidirectional LSTM model with standard lookup word representations (word-BLSTM). The latter two were trained with/without pre-trained word vectors. To pre-train the word vectors, we use a modified skip-gram model [Ling et al., 2015] trained on the English Gigaword corpus[11] version 5.

We divide each dataset in three parts, training (80%), development (10%) and testing (10%). The three subsets are selected in sequence starting from the oldest tweets and from the training set since automatic systems are usually trained on past tweets, and need to be robust to future topic variations.

Table 7.1 reports the results of the five models and the baseline. All neural models outperform the baselines in all the experimental setups. However, the BOW and AVG are quite competitive, suggesting that most emojis come along with specific words (like the word *love* and the emoji ❤️). Also the FastText classifier seems quite accurate comparing to the rest of the the systems, scoring the best F1 in the 20 emojis task. However, considering sequences of words in the models seems important for encoding the meaning of the tweet and therefore contextualize the emojis used. Indeed, the B-LSTMs models always outperform BOW and AVG, and in 2 task out of 3 also FastText. The character-based model with pre-trained vectors is the most accurate at predicting emojis. The character-based model seems to capture orthographic variants of the same word in social media. Similarly, pre-trained vectors allow to initialize the system with unsupervised pre-trained semantic knowledge [Ling et al., 2015], which helps to achieve better results.

[11] https://catalog.ldc.upenn.edu/LDC2003T05

Emoji	P	R	F1	Rank	Num
😂	0.48	**0.74**	**0.58**	2.12	783
❤️	0.32	**0.74**	0.45	**1.59**	757
😍	0.35	0.22	0.27	3.58	470
😊	0.31	0.15	0.21	4.2	260
😎	0.24	0.1	0.14	4.39	212
🔥	0.46	0.49	0.47	3.76	207
💕	1	0	0.01	4.69	206
💯	0.44	0.19	0.27	5.15	200
💪🏼	0.44	0.54	0.48	4.71	165
🙌🏻	0.33	0.11	0.17	5.79	150
😘	0.3	0.12	0.17	5.78	148
💙	0.54	0.11	0.18	6.73	131
✨	0.45	0.19	0.27	6.43	120
💋	**0.56**	0.09	0.15	7.58	112
👌🏼	0.2	0.01	0.02	9.01	110
🙏🏻	0.46	0.33	0.39	5.83	108
😭	0.5	0.08	0.13	4.9	105
🎉	0.32	0.25	0.28	6.13	89
❄️	0.44	0.53	0.48	5.35	34
🎄	0.22	0.67	0.33	1.67	3

Table 7.2: Precision, Recall, F-measure, Ranking and occurrences in the test set of the 20 most frequent emojis using char-BLSTM + Pre.

7.4.1.0.1 Qualitative Analysis of Best System: We analyze the performances of the char-BLSTM with pre-trained vectors on the 20-emojis dataset, as it resulted to be the best system in the experiment presented above. In Table 7.2 we report Precision, Recall, F-measure and Ranking[12] of each emoji. We also added in the last column the occurrences of each emoji in the test set.

The frequency seems to be very relevant. The Ranking of the most frequent emojis is lower than the Ranking of the rare emojis. This means that if an emoji is frequent, it is more likely to be on top of the possible choices even if incorrect. On the other hand, the F-measure does not seem to depend on frequency, as the highest F-measures are scored by a mix of common and uncommon emojis (😂, ❤️, 🔥, and ❄️) which are respectively the first, second, the sixth and the second last emoji in terms of frequencies.

[12] The Ranking is a number between 1 and 20 that represents the average number of emojis with higher probability than the gold emoji in the probability distribution of the classifier.

	Humans			B-LSTM		
Emo	**P**	**R**	**F1**	**P**	**R**	**F1**
😂	0.73	0.56	0.63	0.7	0.84	**0.77**
❤	0.53	0.51	0.52	0.61	0.78	**0.69**
😍	0.43	0.38	**0.4**	0.52	0.3	0.38
💯	0.19	0.4	0.26	0.62	0.26	**0.37**
🔥	0.24	0.26	0.25	0.66	0.51	**0.58**
Avg	0.53	0.48	0.50	0.65	0.65	**0.65**

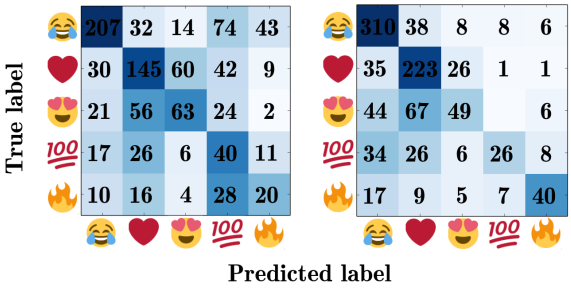
Table 7.3: Precision, Recall and F-Measure of human evaluation and the character-based B-LSTM for the 5 most frequent emojis and 1,000 tweets.

The frequency of an emoji is not the only important variable to detect the emojis properly; it is also important whether in the set of emojis there are emojis with similar semantics[13]. If this is the case the model prefers to predict the most frequent emojis. This is the case of the 💕 emoji that is almost never predicted, even if the Ranking is not too high (4.69). The model prefers similar but most frequent emojis, like ❤ (instead of 💕). The same behavior is observed for the 💙 emoji, but in this case the performance is a bit better due to some specific words used along with the blue heart: “blue”, “sea” and words related to childhood (e.g. “little” or “Disney”).

Another interesting case is the Christmas tree emoji 🎄, that is present only three times in the test set (as the test set includes most recent tweets and Christmas was already over; this emoji is commonly used in tweets about Christmas). The model is able to recognize it twice, but missing it once. The correctly predicted cases include the word “Christmas”; and it fails to predict: *“getting into the holiday spirit with this gorgeous pair of leggings today ! #festiveleggings”*, since there are no obvious clues (the model chooses ❤ instead probably because of the intended meaning of “holiday” and “gorgeous”.).

In general the model tends to confuse similar emojis to ❤ and 😂, probably for their higher frequency and also because they are used in multiple contexts. An interesting phenomenon is that 😭 is often confused with 😂. The first one represent a small face crying, and the second one a small face laughing, but the results suggest that they appear in similar tweets. The punctuation and tone used is often similar (many exclamation marks and words like *“omg”* and *“hahaha”*). Irony may also play a role to explain the confusion, e.g. *“I studied journalism and communications , I’ll be an awesome speller! Wrong. 😭 haha so much fun”*.

[13] As we have seen in the previous chapter some emojis share a very similar semantics, and it is difficult to distinguish them.



	😂	❤️	😍	💯	🔥
😂	207	32	14	74	43
❤️	30	145	60	42	9
😍	21	56	63	24	2
💯	17	26	6	40	11
🔥	10	16	4	28	20

	😂	❤️	😍	💯	🔥
😂	310	38	8	8	6
❤️	35	223	26	1	1
😍	44	67	49		6
💯	34	26	6	26	8
🔥	17	9	5	7	40

Figure 7.1: Confusion matrix of the second experiment. On the left the human evaluation and on the right the char-BLSTM model.

7.4.2 Second Textual Experiment

Given that [Miller et al., 2016] pointed out that people tend to give multiple interpretations to emojis, we carried out an experiment in which we evaluated human and machine performances on the same task. We randomly selected 1,000 tweets from our test set of the 5 most frequent emojis used in the previous experiment, and asked humans to predict, after reading a tweet (with the emoji removed), the emoji the text evoked. We opted for the 5 emojis task to reduce annotation efforts. After displaying the text of the tweet, we asked the human annotators "What is the emoji you would include in the tweet?", and gave the possibility to pick one of 5 possible emojis 😂, ❤️, 😍, 💯, and 🔥. Using the crowdsourcing platform ''Crowd-Flower", we designed an experiment where the same tweet was presented to four annotators (selecting the final label by majority agreement). Each annotator assessed a maximum of 200 tweets. The annotators were selected from the United States of America and of high quality (level 3 of CrowdFlower). One in every ten tweets, was an obvious test question, and annotations from subjects who missed more than 20% of the test questions were discarded. The overall inter-annotator agreement was 73% (in line with previous findings [Miller et al., 2016]). After creating the manually annotated dataset, we compared the human annotation and the char-BLSTM model with the gold standard (i.e. the emoji used in the tweet).

We can see in Table 7.3, where the results of the comparison are presented, that the char-BLSTM performs better than humans, with a F1 of 0.65 versus 0.50. The emojis that the char-BLSTM struggle to predict are 😍 and 💯 , while the human annotators mispredict 💯 and 🔥 mostly. We can see in the confusion matrix of Figure 7.1 that 😍 is misclassified as ❤️ by both human and LSTM, and the 💯 emoji is mispredicted as 😂 and ❤️. An interesting result is the number of times 💯 was chosen by human annotators; this emoji occurred 100 times (by chance) in the test set, but it was chosen 208 times, mostly when the correct label was the laughing emoji 😂. We do not observe the same behavior in the char-BLSTMs, perhaps because they encoded information about the probability of these two emojis and when in doubt, the laughing emoji was chosen as more probable.

7.5 Multimodal Instagram Experiments

In order to study the relation between Instagram posts and emojis, we performed two different experiments. In the first experiment (Section 7.4) we compare the FastText model with the state of the art on emoji classification (B-LSTM) by [Barbieri et al., 2017]. Our second experiment (Section 7.5.2) evaluates the visual (ResNet) and textual (FastText) models on the emoji prediction task. Moreover, we evaluate a multimodal combination of both models respectively based on visual and textual inputs. Finally we discuss the contribution of each modality to the prediction task.

We use 80% of our dataset (introduced in Section 7.2.1) for training, 10% to tune our models, and 10% for testing. We select these three sets randomly to avoid biases and overfitting. When performing our emoji prediction experiments we consider three different settings : 5 most frequent (top-5), 10 most frequent (top-10) and 20 most frequent (top-20) emojis.

In the next Section 7.5.1 we describe visual and textual feature extraction processes. We also provide an overview of the training procedures.

7.5.1 Feature Extraction and Classifier

To extract the features and classify, we follow the procedure described below for each one of the two emojis prediction experiments. To model visual features we first finetune the ResNet presented in Section 7.3.2 on the emoji prediction task, then extract the vectors from the input of the last fully connected layer (that leads to the softmax). The textual embeddings are the bag of features shown in Section 7.3.1.2 (the x_n vectors), extracted after training the FastText model on the emoji prediction task. We decided to use FastText instead of the B-LSTMs as we have seen in the previous experiments the results are similar, but FastText is way faster to train (takes minutes instead of days), and seems to be more suitable in a big data context. With respect to the combination of textual and visual modalities [Bruni et al., 2014] investigated the contribution of linguistic and visual inputs to similarity estimation and outlined that different approaches can be adopted to jointly consider (and thus fuse) inputs coming from different modalities. In general, it is possible to (1) jointly learn the representation of each modality (early fusion), (2) independently learn unimodal representations and combine them to obtain a multimodal one (middle fusion) or (3) exploit independently each unimodal representation to support a similarity prediction (or classification) task and then combine the results [Kiela and Clark, 2015]. In our experiments we adopt the middle fusion approach: we associate to each Instagram post a multimodal embedding obtained by concatenating the unimodal representations of the same post (i.e. the visual and textual embeddings), previously learned. Then, we feed a classifier[14] with visual (ResNet), textual (FastText), pre-traintimodal feature embeddings (fusion of the two previous representation, explained in the next section), and test the accuracy of the three systems.

	top-5			**top-10**			**top-20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Majority	0.079	0.200	0.113	0.027	0.100	0.042	0.009	0.050	0.015
Weight. Rand.	0.201	0.200	0.201	0.098	0.098	0.098	0.046	0.048	0.047
Visual	0.386	0.311	0.310	0.263	0.209	0.205	0.203	0.175	0.161
Textual	0.561	0.544	0.549	0.416	0.375	0.383	0.367	0.299	0.313
Multimodal	0.574	0.563	**0.567**	0.423	0.405	**0.411**	0.366	0.352	**0.355**
Impr. %	2.32	3.49	3.28	1.68	8	7.31	-0.27	17.73	13.42

Table 7.4: Prediction results of top-5, top-10 and top-20 most frequent emojis in the Instagram dataset: Precision (P), Recall (R), F-measure (F1). Experimental settings: majority baseline, weighted random, visual, textual and multimodal systems. In the last line we report the percentage improvement of the multimodal over the textual system.

7.5.2 Multimodal Emoji Prediction

We present the results of the three emoji classification tasks, using as features: the visual, textual and multimodal (see Table 7.4).

The emoji prediction task seems very difficult by just using the image of the Instagram post (**Visual**), even if it largely outperforms the majority baseline[15] and weighted random [16]. On the other side we have better performances when we use feature embeddings extracted from the text. This makes sense as the emoji used as label is strictly related to the text, since it is part of it. The most interesting finding is that when we use a multimodal combination of visual and textual features, we get a non-negligible improvement. This suggests that these two modalities embed different representation of the posts, and when used together they are complementary. It is also interesting to note that the more emojis to predict, the more improvement the multimodal system provides over the textual system (3.28% for top-5 emojis, 7.31% for top-10 emojis, and 13.42 for the top-20 emojis task).

[14] L2 regularized logistic regression

[15] Always predict ❤ since it is the most frequent emoji.

[16] Random keeping labels distribution of the training set

7.5.3 What Does Each Modality Learn?

We explore the errors of the classifier. In Table 7.5 we show the results for each class in the top-20 emojis task. The Table includes results for the Textual, Visual and Multimodal systems, and the distribution of the emojis (percentage of each emoji over the entire test set).

The emoji better predicted by the textual features is the most frequent one ❤️ (0.62) and the flag 🇺🇸 (0.52). The latter is easy to predict because it appears in specific contexts: when the word USA/America is used (or when American cities are referred, like #NYC).

The hardest emojis to predict by the text only system are the two gestures 👌🏽 (0.12) and 🙌🏼 (0.13). The first one is often selected when the gold standard emoji is the second one or instead of 😍. 🙌🏼 is often mispredicted by wrongly selecting 😍 or 😎. The confusion matrix resulting from the prediction of the top-20 most frequent emojis from Instagram posts is shown in Figure 7.2.

When we rely on visual contents (Instagram picture), the emojis which are easily predicted are the ones in which the associated photos are similar. For instance, most of the pictures associated to 🐶 are dog/pet pictures. The same happens for ☀️, as most pictures are taken outside in general (e.g. having breakfast in an outdoor table, or at the beach). It is also important that the majority of posts with the emoji ☀️ are very bright (and thus, sunny). If we analyze pictures associated to the ❤️ emoji we can surprisingly note that most of them are not related to love. Instead, they are related to friendship and family, and in most of the pictures there is more than one person. As a consequence, by relying on visual information, the ❤️ emoji is difficult to predict (0.35 F1, Table 7.5). The accuracy of 💪🏽 is high since most posts including this emoji are related to fitness (and the pictures are simply either selfies at the gym, weight lifting images, and about protein food).

Employing a multimodal approach improves performance. This means that the two modalities are somehow complementary, and adding visual information helps to solve potential ambiguities that arise when relying only on textual content. For instance, the two following Instagram posts are mispredicted by the text model, but correctly predicted by the vision (and more importantly, the multimodal) system:

1. "Love my new home ☀️"
   associated to a picture of a bright garden, outside;

2. "I can't believe it's the first day of school!!! I love being these boys' mommy!!!! *#myboys #mommy* 💙"
   associated to picture of two boys wearing two blue shirts.

In (1) and (2), the textual system selects the ❤️ instead. The blue color in the picture associated to (2) helps to change the color of the heart, and the sunny/bright picture of the garden in (1) helps to correctly predict ☀.

Emoji	Text	Visual	MM	%
❤	0.62	0.35	**0.69**	17.46
😂	0.45	0.3	**0.47**	9.1
😍	0.32	0.15	**0.34**	8.41
💕	0.23	0.08	**0.26**	5.91
✨	0.35	0.17	**0.36**	5.73
🔥	0.45	0.24	**0.46**	4.58
🇺🇸	0.52	0.23	**0.53**	4.31
☀	0.38	0.26	**0.49**	4.15
😎	0.19	0.1	**0.22**	3.84
🙌	0.13	0.03	**0.16**	3.73
💙	0.22	0.15	**0.29**	3.68
✌	0.2	0.02	**0.26**	3.55
😘	0.13	0.02	**0.2**	3.54
💯	0.26	0.17	**0.31**	3.51
💪	0.43	0.25	**0.45**	3.31
😊	0.12	0.01	**0.16**	3.25
👌	0.12	0.02	**0.15**	3.14
🙏	0.34	0.11	**0.36**	3.11
🎉	0.36	0.04	**0.37**	2.91
🐶	0.45	0.54	**0.59**	2.82

Table 7.5: F-measure and percentage of occurrences in the test set of the 20 most frequent emojis using the three different models.

7.5.4 Which portions of an image are used to predict an emoji?

Recently [Zhou et al., 2016] proposed an approach useful to visualize the areas of an image where a CNN, trained to label pictures, focuses its attention to drive the label-prediction process[17]. By performing global average pooling on the convolutional feature maps obtained after the chain of layers of a CNN, they are able to build a heatmap, referred to as Class Activation Mapping: this heatmap highlights the portions of the input image that have mostly influenced the image classification process.

We use this technique in order to explore if there is any relevant pattern in the portions of an Instagram image exploited by our CNN to predict the most

[17] http://cnnlocalization.csail.mit.edu/

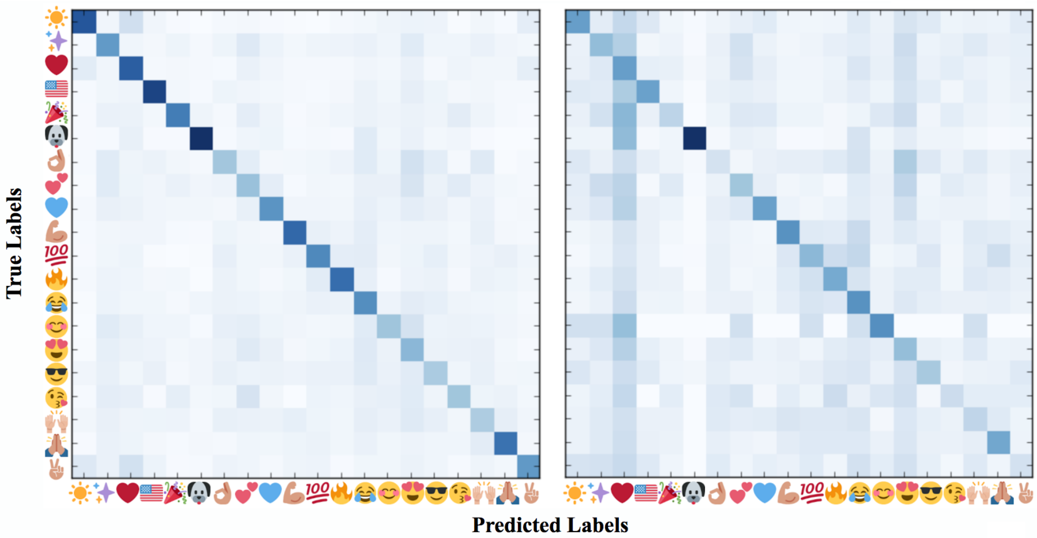


Figure 7.2: Confusion matrix of the two modalities in the top-20 task. On the left the textual confusion matrix and on the right the confusion matrix of the visual embeddings. Textual and visual embeddings make mispredictions on different classes.

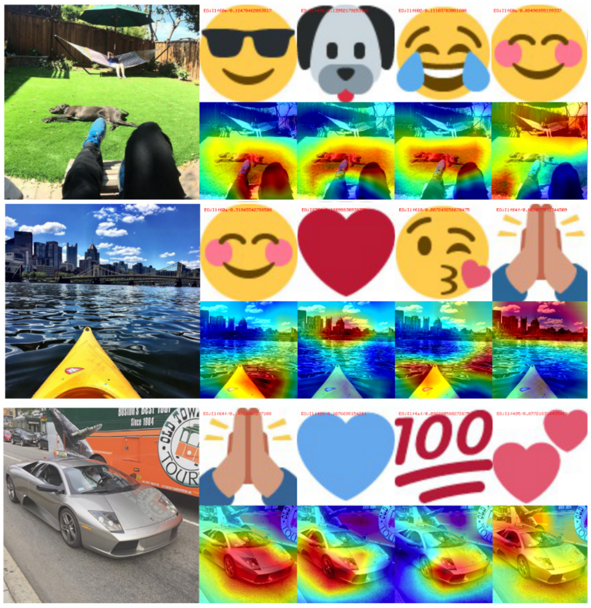


Figure 7.3: Examples of Class Activation Mapping heatmap generated from images of three Instagram posts. From left to right, we show the first four most likely predictions of the network.

likely emoji to associate to the same image. To this purpose, we consider a CNN trained to predict the most likely emoji to associate to the images of the posts of our Instagram dataset, limiting our analysis to posts incluiding one of the 20 most frequent emojis (top-20). Then, for each one of these 20 emojis, we randomly selected 50 Instagram posts in which the considered emoji has been included in the related comment by the author. For each post picture we highlighted, by means of the Class Activation Mapping heatmap, the parts of the picture where the CNN focuses its attention when evaluating each one of the top-20 emojis to predict the most likely one.

By visually exploring the image heatmaps of the set of Instagram post pictures we can note that in most cases it is quite difficult to determine a clear association between the emoji used by the user and some particular portion of the image. For this reason detecting the correct emoji given an image is harder than a simple object recognition task, as the emoji choice depends on personal emotions of the user who posted the image. However, we can note that some emojis are associated to areas of pictures that share specific semantics, as in the examples shown in Figure7.3. For each one of the three pictures we show the first four predictions of the CNN (four most likely emoji to be associated to each picture), and where the network focus its attention (part of the picture highlighted in red). We can see that in the first example the network selects the smile with sunglasses emoji 😎 because of the legs in the bottom of the image, the dog emoji 🐶 is selected while focusing on the dog in the image, and the smiling emoji 😊 while focusing on the person in the back, who is lying on an hammock. In the second example the network selects again the smiling emoji 😊 because of the water and part of the kayak, the heart emoji ❤️ focusing on the city landscape, and the praying emoji 🙏🏽 focusing on the sky. The same "praying" emoji is also selected when focusing on the luxury car in the third example, probably because the same emoji is used with another meaning, similar to "please, I want one of these cars".

7.6 Conclusions

In this chapter we explore whether it is possible to predict the most likely emoji used in a text message. This is a difficult problem given that, as we have seen in the previous chapter, emojis are used in a very subjective way, and we all use them differently. We propose a new task, the emoji prediction task, that consists in predicting the emoji present in a text message using only the text. This is an interesting task for different reasons. First, because predicting the emoji is like predicting the emotional context of the message, indeed it has been recently shown [Felbo et al., 2017] that learning to predict emojis helps in several subjective-related tasks, like sentiment analysis and emotion prediction. Moreover, learning to understand and use emojis in the correct context is extremely valuable in human computer interaction systems, as this kind of new visual language is a form of communication that we use everyday.

We show in this chapter that this emoji prediction task can be performed by deep-learning systems with good accuracy. Moreover, we also explore whether the images included in social media posts (like Instagram) are important to recognize the emoji included in the text content of the same post. We show that using a deep multimodal system, that learns both textual and visual representations of social media posts, it is possible to outperform systems based only on the text content. This suggests that both textual and visual content of social media posts encode important information for the emoji prediction task, and that these two modalities are complementary.

Chapter 8

CONCLUSIONS

In this dissertation we studied the language of social media, focusing on two particular aspects: irony and emojis. We proposed novel automatic systems, based on machine learning algorithms, able to recognize and interpret these two phenomena.

Irony detection was tackled as a binary classification problem, where, given a tweet, the task is to recognize if the tweet is ironic or not. To solve this task, we proposed a machine learning approach where a tweet is represented with several features calculated using shallow characters (e.g. length of the tweet and number of words), lexicons (frequency lexicons and sentiment lexicons), and also knowledge repositories (WordNet for the synonyms, and features related to synsets). Our approach outperformed the state of the art. Moreover, we avoided the use of words as features (like Bag of Words approaches), in order to be as much topic-independent as possible.

We have also explored the problem of topic bias in sentiment analysis and irony detection, showing that traditional word-based systems are not robust when they have to recognize irony on a new domain. On the other side, we show that our approach was better, and less biased by the topic when predicting irony.

We also tested our approach for irony detection to recognize whether a news post on Twitter is satirical or real, in English, Spanish and Italian, and obtained significant results. We were able to automatically recognize if a tweet belonged to a satirical or a non-satirical Twitter account.

Future research in the irony detection could be exploring our approach with new languages and seeking methods to combine languages to obtain better accuracy in cross-lingual irony detection. Additionally, the procedure of cross-language feature generation could be investigated, in order to improve the compatibility between languages.

Regarding the emojis studies, we have explored if and how the meaning and usage of emojis varies across languages and locations (by analyzing English tweets posted from United States of America, English tweets from United Kingdom, Spanish tweets from Spain and Italian tweets from Italy) as well as across seasons (by analyzing tweets posted in spring, summer, autumn and winter). We used distributional semantic models to represent the meaning of the emojis in each language, location and season respectively. We have found that some emojis have different meanings over different countries or time of the year. This is in line with many previous findings that suggest that emojis are used in a very subjective way, and that we interpret them differently.

Our results suggest that even if the overall emoji semantics of the languages we studied is similar, we can identify some emojis that are not used in the same way from language to language: this fact may be related to the cultural differences that exist between countries. For instance, the clover emoji 🍀 is used in a friendship and love context in Spain, while in the other countries is mainly used in relation to luck and the symbol of Ireland.

We studied the semantics of emojis and regarding the variations of the meaning of emojis across seasons we figured out that even if most of the emojis preserve their semantics, specific differences can be identified. Two examples are the gift 🎁 and the pine 🌲 emojis that in winter are used as Christmas-related emojis, but in spring and summer are used to respectively point out a birthday present and a tree.

In the further future, we are planning to run more extensive analyses to automatically spot and interpret finer-grained differences in the semantics of emojis. Moreover, we would like to experiment with approaches to detect changes of the semantics of specific emojis as they occur by processing the stream of Twitter posts in real-time: in this way we would like to relate changes in the meaning of emojis to specific events and social trends. We would also like to further investigate the compositional meaning of emojis, that is the meaning conveyed by exploiting two or more emojis together, one next to the other. We will also evaluate how classification systems, useful for instance to determine the sentiment of social media posts, are affected by language, location and season-specific differences in the meaning of emojis. Finally, other avenues of research can be emoji aware language generation and also creative computational systems.

We also explored whether it is possible to predict the most likely emoji used in a text message. This is not an easy task, as we have seen that emojis are used in a very subjective way, and we all use them differently. We proposed a new task, the emoji prediction task, that consists in predicting the emoji present in a text message using only the text. We have shown that this emoji prediction task can be performed by deep-learning systems with good accuracy. We also extended the emoji prediction task to posts that include both visual (images) and textual content and shown that using both modalities improves the emoji prediction accuracy, suggesting that these two modalities are complementary and include important information about the use of the emoji.

Regarding the emoji prediction problem, many future improvements are possible. We have seen that time and location determine the semantics of the emoji and, for this reason, this should be taken into account when predicting the emoji. Also, new modalities should be explored, expanding the research to also videos and audios, as these new modalities are drastically increasing online.

We hope that this dissertation will be useful for future studies on computational models to understand the language of social media.

Bibliography

[Abercrombie and Hovy, 2016] Abercrombie, G. and Hovy, D. (2016). Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of Twitter conversations. *ACL 2016*, page 107.

[Agirre and Soroa, 2009] Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

[Ai et al., 2017] Ai, W., Lu, X., Liu, X., Wang, N., Huang, G., and Mei, Q. (2017). Untangling emoji popularity through semantic embeddings. In *ICWSM*, pages 2–11.

[Amir et al., 2016] Amir, S., Wallace, B. C., Lyu, H., and Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

[Aoki and Uchida, 2011] Aoki, S. and Uchida, O. (2011). A method for automatically generating the emotional vectors of emoticons using weblog articles. In *Proc. 10th WSEAS Int. Conf. on Applied Computer and Applied Computational Science, Stevens Point, Wisconsin, USA*, pages 132–136.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Language Resources and Evaluation Conference*, volume 10, pages 2200–2204.

[Ballesteros et al., 2015] Ballesteros, M., Dyer, C., and Smith, N. A. (2015). Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

[Bamman and Smith, 2015] Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on Twitter. In *ICWSM*, pages 574–577.

[Barbieri et al., 2017] Barbieri, F., Ballesteros, M., and Saggion, H. (2017). Are emojis predictable? In *Proceedings of the 2017 Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain. Association for Computational Linguistics.

[Barbieri et al., 2016a] Barbieri, F., Basile, V., Croce, D., Nissim, M., Novielli, N., and Patti, V. (2016a). Overview of the evalita 2016 sentiment polarity classification task. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016).*

[Barbieri et al., 2014a] Barbieri, F., Ronzano, F., and Saggion, H. (2014a). Italian irony detection in Twitter: A first approach. In *The First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA*, pages 28–32.

[Barbieri et al., 2015a] Barbieri, F., Ronzano, F., and Saggion, H. (2015a). Do we criticise (and laugh) in the same way? Automatic detection of multi-lingual satirical news in Twitter. In *IJCAI*, pages 1215–1221.

[Barbieri et al., 2015b] Barbieri, F., Ronzano, F., and Saggion, H. (2015b). How topic biases your results? A case study of sentiment analysis and irony detection in Italian. In *RANLP*, pages 41–47.

[Barbieri et al., 2015c] Barbieri, F., Ronzano, F., and Saggion, H. (2015c). Is this tweet satirical? A computational approach for satire detection in Spanish. *Procesamiento del Lenguaje Natural*, 55:135–142.

[Barbieri et al., 2016b] Barbieri, F., Ronzano, F., and Saggion, H. (2016b). What does this emoji mean? A vector space skip-gram model for Twitter emojis. In *Language Resources and Evaluation Conference*.

[Barbieri and Saggion, 2014a] Barbieri, F. and Saggion, H. (2014a). Modelling irony in Twitter. In *EACL Student Research Workshop*, pages 56–64.

[Barbieri and Saggion, 2014b] Barbieri, F. and Saggion, H. (2014b). Modelling irony in Twitter: Features analysis and evaluation. In *Language Resources and Evaluation Conference*.

[Barbieri et al., 2014b] Barbieri, F., Saggion, H., and Ronzano, F. (2014b). Modelling sarcasm in Twitter, a novel approach. In *WASSA@ ACL*, pages 50–58.

[Barbosa and Feng, 2010] Barbosa, L. and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 36–44. Association for Computational Linguistics.

[Basile et al., 2014] Basile, V., Bolioli, A., Nissim, M., Patti, V., and Rosso, P. (2014). Overview of the Evalita 2014 sentiment polarity classification task. In *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*, Pisa, Italy.

[Basile and Nissim, 2013] Basile, V. and Nissim, M. (2013). Sentiment analysis on Italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

[Bernardi et al., 2016] Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., and Plank, B. (2016). Automatic description generation from images: A survey of models, datasets, and evaluation measures. *J. Artif. Intell. Res.(JAIR)*, 55:409–442.

[Bharti et al., 2015] Bharti, S. K., Babu, K. S., and Jena, S. K. (2015). Parsing-based sarcasm sentiment recognition in Twitter data. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 1373–1380. IEEE.

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

[Blei and Jordan, 2003] Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[Boia et al., 2013] Boia, M., Faltings, B., Musat, C.-C., and Pu, P. (2013). A :) is worth a thousand words: How people attach sentiment to emoticons and words in tweets. In *Social Computing (SocialCom), 2013 International Conference on*, pages 345–350. IEEE.

[Bontcheva et al., 2013] Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. (2013). TwitIE: An open-source information extraction pipeline for microblog text. In *Proceedings of the International*

Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics.

[Bosco et al., 2013a] Bosco, C., Patti, V., and Bolioli, A. (2013a). Developing corpora for sentiment analysis and opinion mining: The case of irony and SENTI-TUT. *Intelligent Systems, IEEE*.

[Bosco et al., 2013b] Bosco, C., Patti, V., and Bolioli, A. (2013b). Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63.

[Bouazizi and Ohtsuki, 2015] Bouazizi, M. and Ohtsuki, T. (2015). Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 1594–1597. IEEE.

[Brown, 1980] Brown, R. L. (1980). The pragmatics of verbal irony. *Language use and the uses of language*, pages 111–127.

[Bruni et al., 2014] Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(2014):1–47.

[Burfoot and Baldwin, 2009] Burfoot, C. and Baldwin, T. (2009). Automatic satire detection: Are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 161–164. ACL.

[Burgers, 2010] Burgers, C. F. (2010). *Verbal irony: Use and effects in written discourse*. PhD thesis - Radboud University, The Netherlands.

[Buschmeier et al., 2014] Buschmeier, K., Cimiano, P., and Klinger, R. (2014). An impact analysis of features in a classification approach to irony detection in product reviews. In *WASSA@ ACL*, pages 42–49.

[Cambria et al., 2014] Cambria, E., Wang, H., and White, B. (2014). Guest editorial: Big social data analysis. *Knowledge-Based Systems*, (69):1–2.

[Camp, 2012] Camp, E. (2012). Sarcasm, pretense, and the semantics/pragmatics distinction. *Noûs*, 46(4):587–634.

[Carreras et al., 2004] Carreras, X., Chao, I., Padró, L., and Padró, M. (2004). FreeLing: An open-source suite of language analyzers. In *Language Resources and Evaluation Conference*.

[Carvalho et al., 2009] Carvalho, P., Sarmento, L., Silva, M. J., and De Oliveira, E. (2009). Clues for detecting irony in user-generated contents: Oh...!! it's so easy;-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

[Castellucci et al., 2015] Castellucci, G., Croce, D., and Basili, R. (2015). Acquiring a large scale polarity lexicon through unsupervised distributional methods. In *International Conference on Applications of Natural Language to Information Systems*, pages 73–86. Springer.

[Charalampakis et al., 2016] Charalampakis, B., Spathis, D., Kouslis, E., and Kermanidis, K. (2016). A comparison between semi-supervised and supervised text mining techniques on detecting irony in Greek political tweets. *Engineering Applications of Artificial Intelligence*, 51:50–57.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[Clark and Gerrig, 1984] Clark, H. H. and Gerrig, R. J. (1984). On the pretense theory of irony. *Journal of Experimental Psychology: General*.

[Colletta, 2009] Colletta, L. (2009). Political satire and postmodern irony in the age of Stephen Colbert and Jon Stewart. *The Journal of Popular Culture*, 42(5):856–874.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine learning*, 20(3):273–297.

[Csurka et al., 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

[Currie, 2006] Currie, G. (2006). Why irony is pretence. *The architecture of the imagination*, pages 111–33.

[Davidov et al., 2010] Davidov, D., Tsur, O., and Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

[Deng et al., 2009a] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009a). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Deng et al., 2009b] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009b). ImageNet: A large-scale hierarchical image database. In *CVPR09*.

[Desai and Dave, 2016] Desai, N. and Dave, A. D. (2016). Sarcasm detection in Hindi sentences using support vector machine. *International Journal*, 4(7).

[Dhingra et al., 2016] Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., and Cohen, W. (2016). Tweet2Vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany. Association for Computational Linguistics.

[Dyer et al., 2015] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

[Dyer et al., 2016] Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

[Dynel, 2014] Dynel, M. (2014). Linguistic approaches to (non) humorous irony. *Humor*, 27(4):537–550.

[Dzmitry et al., 2014] Dzmitry, B., Kyunghyun, C., and Yoshua, B. (2014). Neural machine translation by jointly learning to align and translate. In *In Proceeding of the third International Conference on Learning Representations*, Toulon, France.

[Eisner et al., 2016] Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., and Riedel, S. (2016). emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

[Farıas et al., 2015] Farıas, I. H., Benedı, J.-M., and Rosso, P. (2015). Applying basic features from sentiment analysis for automatic irony detection. *Pattern Recognition and Image Analysis. Lecture Notes in Computer Science*, 9117:337–344.

[Felbo et al., 2017] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

[Feng and Lapata, 2010] Feng, Y. and Lapata, M. (2010). Visual information in semantic representation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 91–99. Association for Computational Linguistics.

[Fersini et al., 2015] Fersini, E., Pozzi, F. A., and Messina, E. (2015). Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–8. IEEE.

[Filatova, 2012] Filatova, E. (2012). Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Language Resources and Evaluation Conference*, pages 392–398.

[Fowler, 2010] Fowler, H. W. (2010). *A dictionary of modern English usage: The classic first edition*. Oxford University Press.

[Frenda, 2016] Frenda, S. (2016). Computational rule-based model for irony detection in Italian Tweets. In *CLiC-it/EVALITA*.

[Ghosh and Veale, 2016] Ghosh, A. and Veale, T. (2016). Fracking sarcasm using neural network. In *WASSA@ NAACL-HLT*, pages 161–169.

[Ghosh et al., 2015] Ghosh, D., Guo, W., and Muresan, S. (2015). Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *EMNLP*, pages 1003–1012.

[Gibbs and O'Brien, 1991] Gibbs, R. W. and O'Brien, J. (1991). Psychological aspects of irony understanding. *Journal of pragmatics*, 16(6):523–530.

[Gimpel et al., 2011] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

[Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Go et al., 2009] Go, A., Huang, L., and Bhayani, R. (2009). Twitter sentiment analysis. *Entropy*, 17.

[González-Ibánez et al., 2011] González-Ibánez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*, pages 581–586. Association for Computational Linguistics.

[Grant, 2004] Grant, D. (2004). *The Sage handbook of organizational discourse*. Sage.

[Graves and Schmidhuber, 2005] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland.

[Grice, 1978] Grice, H. P. (1978). Further notes on logic and conversation. *1978*, 1:13–128.

[Grice et al., 1975] Grice, H. P., Cole, P., Morgan, J., et al. (1975). Logic and conversation. *1975*, pages 41–58.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hogenboom et al., 2013] Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F., and Kaymak, U. (2013). Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 703–710. ACM.

[Hogenboom et al., 2015] Hogenboom, A., Bal, D., Frasincar, F., Bal, M., De Jong, F., and Kaymak, U. (2015). Exploiting emoticons in polarity classification of text. *J. Web Eng.*, 14(1&2):22–40.

[Howard, 2013] Howard, A. G. (2013). Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*.

[Ide and Suderman, 2004] Ide, N. and Suderman, K. (2004). The American national corpus first release. In *Proceedings of the Language Resources and Evaluation Conference*.

[Jansen et al., 2009] Jansen, B. J., Zhang, M., Sobel, K., and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

[Jiang et al., 2015] Jiang, F., Liu, Y.-Q., Luan, H.-B., Sun, J.-S., Zhu, X., Zhang, M., and Ma, S.-P. (2015). Microblog sentiment analysis with emoticon space model. *Journal of Computer Science and Technology*, 30(5):1120–1129.

[Jibril and Abdullah, 2013] Jibril, T. A. and Abdullah, M. H. (2013). Relevance of emoticons in computer-mediated communication contexts: An overview. *Asian Social Science*, 9(4):201.

[John and Langley, 1995] John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

[Joshi et al., 2016a] Joshi, A., Bhattacharyya, P., and Carman, M. J. (2016a). Automatic sarcasm detection: A survey. *arXiv preprint arXiv:1602.03426*.

[Joshi et al., 2016b] Joshi, A., Jain, P., Bhattacharyya, P., and Carman, M. (2016b). Who would have thought of that! A hierarchical topic model for extraction of sarcasm-prevalent topics and sarcasm detection. *arXiv preprint arXiv:1611.04326*.

[Joshi et al., 2015] Joshi, A., Sharma, V., and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *ACL (2)*, pages 757–762.

[Joshi et al., 2016c] Joshi, A., Tripathi, V., Bhattacharyya, P., and Carman, M. J. (2016c). Harnessing sequence labeling for sarcasm detection in dialogue from TV series Friends. In *CoNLL*, pages 146–155.

[Joulin et al., 2017] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 2017 Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain. Association for Computational Linguistics.

[Karoui et al., 2017] Karoui, J., Benamara, F., Moriceau, V., Patti, V., Bosco, C., and Aussenac-Gilles, N. (2017). Exploring the impact of pragmatic phenomena on irony detection in Tweets: A multilingual corpus study. In *EACL (2)*.

[Karpathy and Fei-Fei, 2015] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

[Khattri et al., 2015] Khattri, A., Joshi, A., Bhattacharyya, P., and Carman, M. J. (2015). Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *WASSA@ EMNLP*, pages 25–30.

[Kiela and Bottou, 2014] Kiela, D. and Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *EMNLP*, pages 36–45.

[Kiela and Clark, 2015] Kiela, D. and Clark, S. (2015). Multi-and cross-modal semantics beyond vision: Grounding in auditory perception. In *EMNLP*, pages 2461–2470.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[Knight, 2004] Knight, C. A. (2004). *The literature of satire*. Cambridge University Press.

[Kreuz and Roberts, 1993] Kreuz, R. J. and Roberts, R. M. (1993). The empirical study of figurative language in literature. *Poetics*, 22(1-2):151–169.

[Kriegeskorte et al., 2008] Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[Kumon-Nakamura et al., 1995] Kumon-Nakamura, S., Glucksberg, S., and Brown, M. (1995). How about another piece of pie: The allusional pretense theory of discourse irony. *Journal of Experimental Psychology: General*, 124(1):3.

[LaMarre et al., 2009] LaMarre, H. L., Landreville, K. D., and Beam, M. A. (2009). The irony of satire political ideology and the motivation to see what you want to see in the Colbert report. *The International Journal of Press/Politics*, 14(2):212–231.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

[Lazaridou et al., 2015] Lazaridou, A., Pham, N. T., and Baroni, M. (2015). Combining language and vision with a multimodal skip-gram model. *arXiv preprint arXiv:1501.02598*.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Liaw et al., 2002] Liaw, A., Wiener, M., et al. (2002). Classification and regression by random forest. *R news*, 2(3):18–22.

[Liebrecht et al., 2013] Liebrecht, C., Kunneman, F., and van Den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.

[Ling et al., 2015] Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado. Association for Computational Linguistics.

[Littman and Mey, 1991] Littman, D. C. and Mey, J. L. (1991). The nature of irony: Toward a computational model of irony. *Journal of Pragmatics*, 15(2):131–151.

[Liu et al., 2014] Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., and Lei, K. (2014). Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*, pages 459–471. Springer.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

[Lucariello, 1994] Lucariello, J. (1994). Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129.

[Lukin and Walker, 2013] Lukin, S. and Walker, M. (2013). Really? Well, apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 30–40.

[Lunando and Purwarianti, 2013] Lunando, E. and Purwarianti, A. (2013). Indonesian social media sentiment analysis with sarcasm detection. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 195–198. IEEE.

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

[Mann, 1973] Mann, J. (1973). *Chaucer and medieval estates satire: The literature of social classes and the general prologue to the Canterbury tales*. Cambridge University Press Cambridge.

[Marina and Peter, 2010] Marina, L. and Peter, S. (2010). *Contemporary stylistics*. Continuum International Publishing Group.

[Maynard et al., 2013] Maynard, D., Dupplaw, D., and Hare, J. (2013). Multimodal sentiment analysis of social media. *Proceedings of the BCS SGAI Workshop on Social Media Analysis*.

[Maynard and Greenwood, 2014] Maynard, D. and Greenwood, M. A. (2014). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Language Resources and Evaluation Conference*, pages 4238–4243.

[Mihalcea, 2012] Mihalcea, R. (2012). Multimodal sentiment analysis. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 1–1. Association for Computational Linguistics.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[Mikolov et al., 2013c] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Miller, 1995] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

[Miller and Charles, 1991] Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

[Miller et al., 2016] Miller, H., Thebault-Spieker, J., Chang, S., Johnson, I., Terveen, L., and Hecht, B. (2016). "Blissfully happy" or "ready to fight": Varying interpretations of emoji. *ICWSMâ16*.

[Miller et al., 2017] Miller, H. J., Kluver, D., Thebault-Spieker, J., Terveen, L. G., and Hecht, B. J. (2017). Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication. In *ICWSM*, pages 152–161.

[Morency et al., 2011] Morency, L.-P., Mihalcea, R., and Doshi, P. (2011). Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176. ACM.

[Muresan et al., 2016] Muresan, S., Gonzalez-Ibanez, R., Ghosh, D., and Wacholder, N. (2016). Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology*, 67(11):2725–2737.

[Nakov et al., 2013] Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., and Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in Twitter. *Proceeding of the North American Chapter of the Association of Computational Linguistics*.

[Nister and Stewenius, 2006] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee.

[Novak et al., 2015] Novak, P. K., Smailović, J., Sluban, B., and Mozetič, I. (2015). Sentiment of emojis. *PloS one*, 10(12):e0144296.

[Owoputi et al., 2013] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics Conference.

[Pang and Lee, 2008] Pang, B. and Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

[Park et al., 2014] Park, J., Baek, Y. M., and Cha, M. (2014). Cross-cultural comparison of nonverbal cues in emoticons on Twitter: Evidence from big data analysis. *Journal of Communication*, 64(2):333–354.

[Park et al., 2013] Park, J., Barash, V., Fink, C., and Cha, M. (2013). Emoticon Style: Interpreting Differences in Emoticons Across Cultures. In *ICWSM*.

[Pavalanathan and Eisenstein, 2015] Pavalanathan, U. and Eisenstein, J. (2015). Emoticons vs. emojis on Twitter: A causal inference approach. *arXiv preprint arXiv:1510.08480*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Peter, 1956] Peter, J. (1956). Complaint and satire in early English literature.

[Picard, 1997] Picard, R. (1997). *Affective Computing*. MIT Press.

[Plank et al., 2016] Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR*, abs/1604.05529.

[Platt et al., 1999] Platt, J. et al. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methodssupport vector learning*, 3.

[Poria et al., 2015] Poria, S., Cambria, E., and Gelbukh, A. F. (2015). Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *EMNLP*, pages 2539–2544.

[Poria et al., 2016a] Poria, S., Cambria, E., Hazarika, D., and Vij, P. (2016a). A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.

[Poria et al., 2016b] Poria, S., Cambria, E., Howard, N., Huang, G.-B., and Hussain, A. (2016b). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174:50–59.

[Potts, 2011] Potts, C. (2011). Developing adjective scales from user-supplied textual metadata. *NSF Workshop on Restructuring Adjectives in WordNet. Arlington,VA.*

[Ptácek et al., 2014] Ptácek, T., Habernal, I., and Hong, J. (2014). Sarcasm detection on Czech and English Twitter. In *COLING*, pages 213–223.

[Quintilian and Butler, 1959] Quintilian, M. and Butler, M. (1959). *The Institutio Oratoria of Quintilian: With an English Translation by H. E. Butler, MA (Vol. 3). London: William Heinemann (Original work published around AD 95).*

[Rajadesingan et al., 2015] Rajadesingan, A., Zafarani, R., and Liu, H. (2015). Sarcasm detection on Twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106. ACM.

[Rakov and Rosenberg, 2013] Rakov, R. and Rosenberg, A. (2013). Sure, i did the right thing: A system for sarcasm detection in speech. In *INTERSPEECH*, pages 842–846.

[Reyes and Rosso, 2012] Reyes, A. and Rosso, P. (2012). Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision Support Systems*, 53(4):754–760.

[Reyes and Rosso, 2014] Reyes, A. and Rosso, P. (2014). On the difficulty of automatically detecting irony: Beyond a simple case of negation. *Knowledge and Information Systems*, 40(3):595.

[Reyes et al., 2012] Reyes, A., Rosso, P., and Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

[Reyes et al., 2013] Reyes, A., Rosso, P., and Veale, T. (2013). A multidimensional approach for detecting irony in Twitter. *Language resources and evaluation*, pages 1–30.

[Riloff et al., 2013] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, volume 13, pages 704–714.

[Russakovsky et al., 2015a] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015a). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Russakovsky et al., 2015b] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015b). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[Salzberg, 1994] Salzberg, S. L. (1994). Programs for machine learning. *Machine Learning*, 16(3):235–240.

[Sermanet and LeCun, 2011] Sermanet, P. and LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2809–2813. IEEE.

[Shelley, 2001] Shelley, C. (2001). The bicoherence theory of situational irony. *Cognitive Science*, 25(5):775–818.

[Shutova et al., 2016] Shutova, E., Kiela, D., and Maillard, J. (2016). Black holes and white rabbits: Metaphor identification with visual features. In *HLT-NAACL*, pages 160–170.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE.

[Sperber and Wilson, 1981] Sperber, D. and Wilson, D. (1981). Irony and the use-mention distinction. *Philosophy*, 3:143–184.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

[Tang et al., 2014] Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014). Building large-scale Twitter-specific sentiment lexicon: A representation learning approach. In *COLING*, pages 172–182.

[Titov and McDonald, 2008] Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120, Beijing, China. ACM.

[Toutanova et al., 2003] Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

[Tsur et al., 2010] Tsur, O., Davidov, D., and Rappoport, A. (2010). ICWSM A great Catchy Name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*, pages 162–169.

[Turney, 2002] Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th ACL*, pages 417–424. Association for Computational Linguistics.

[Turney et al., 2010] Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

[Veale et al., 2013] Veale, T., Feyaerts, K., and Forceville, C. (2013). *Creativity and the agile mind: A multi-disciplinary study of a multi-faceted phenomenon*, volume 21. Walter de Gruyter.

[Vinyals et al., 2015a] Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015a). Grammar as a foreign language. In *Proceeding of the conference on Neural Information Processing Systems*, Montreal, Canada.

[Vinyals et al., 2015b] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015b). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

[Wallace, 2015] Wallace, B. C. (2015). Computational irony: A survey and new perspectives. *The Artificial Intelligence Review*, 43(4):467.

[Wallace et al., 2015] Wallace, B. C., Choe, D. K., and Charniak, E. (2015). Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *ACL (1)*, pages 1035–1044.

[Wallace et al., 2014] Wallace, B. C., Do Kook Choe, L. K., Kertz, L., and Charniak, E. (2014). Humans require context to infer ironic intent (so computers probably do, too). In *ACL (2)*, pages 512–516.

[Wallach, 2006] Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, Pittsburgh, USA. ACM.

[Wang et al., 2016] Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2016). Learning distributed word representations For bidirectional LSTM recurrent neural network. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 527–533, San Diego, California. Association for Computational Linguistics.

[Wang et al., 2015] Wang, Z., Wu, Z., Wang, R., and Ren, Y. (2015). Twitter sarcasm detection exploiting a context-based model. In *Proceedings, Part I, of the 16th International Conference on Web Information Systems Engineering—WISE 2015-Volume 9418*, pages 77–91. Springer-Verlag New York, Inc.

[Wilson and Sperber, 1992] Wilson, D. and Sperber, D. (1992). On verbal irony. *Lingua*, 87(1-2):53–76.

[Wood and Ruder, 2016] Wood, I. and Ruder, S. (2016). Emoji as emotion tags for tweets. In *Language Resources and Evaluation Conference*.

[Yang et al., 2007] Yang, C., Lin, K. H.-Y., and Chen, H.-H. (2007). Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 133–136. Association for Computational Linguistics.

[Zhang et al., 2015] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

[Zhou et al., 2016] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929.