UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Natural Language Processing using Deep Learning in Social Media

Author: Maite Giménez

Directors: D. Vicente Botti
D. Javier Palanca

To Ferran, Bran and Jon
and to all the black swans
that live defying the
statistical probabilites

Abstract

In the last years, Deep Learning (DL) has revolutionised the potential of automatic systems that handle Natural Language Processing (NLP) tasks. We have witnessed a tremendous advance in the performance of these systems. Nowadays, we found embedded systems ubiquitously, determining the intent of the text we write, the sentiment of our tweets or our political views, for citing some examples.

In this thesis, we proposed several NLP models for addressing tasks that deal with social media text. Concretely, this work is focused mainly on Sentiment Analysis and Personality Recognition tasks. Sentiment Analysis is one of the leading problems in NLP, consists of determining the polarity of a text, and it is a well-known task where the number of resources and models proposed is vast. In contrast, Personality Recognition is a breakthrough task that aims to determine the users' personality using their writing style, but it is more a niche task with fewer resources designed ad-hoc but with great potential.

Despite the fact that the principal focus of this work was on the development of Deep Learning models, we have also proposed models based on linguistic resources and classical Machine Learning models. Moreover, in this more straightforward setup, we have explored the nuances of different language devices, such as the impact of emotions in the correct classification of the sentiment expressed in a text.

Afterwards, DL models were developed, particularly Convolutional Neural Networks (CNNs), to address previously described tasks. In the case of Personality

Recognition, we explored the two approaches, which allowed us to compare the models under the same circumstances.

Noteworthy, NLP has evolved dramatically in the last years through the development of public evaluation campaigns, where multiple research teams compare the performance of their approaches under the same conditions. Most of the models here presented were either assessed in an evaluation task or either used their setup. Recognising the importance of this effort, we curated and developed an evaluation campaign for classifying political tweets.

In addition, as we advanced in the development of this work, we decided to study in-depth CNNs applied to NLP tasks. Two lines of work were explored in this regard. Firstly, we proposed a semantic-based padding method for CNNs, which addresses how to represent text more appropriately for solving NLP tasks. Secondly, a theoretical framework was introduced for tackling one of the most frequent critics of Deep Learning: interpretability. This framework seeks to visualise what lexical patterns, if any, the CNN is learning in order to classify a sentence.

In summary, the main achievements presented in this thesis are:

- The organisation of an evaluation campaign for Topic Classification from texts gathered from social media.
- The proposal of several Machine Learning models tackling the Sentiment Analysis task from social media. Besides, a study of the impact of linguistic devices such as figurative language in the task is presented.
- The development of a model for inferring the personality of a developer provided the source code that they have written.
- The study of Personality Recognition task from social media following two different approaches, models based on machine learning algorithms and handcrafted features, and models based on CNNs were proposed and compared both approaches.
- The introduction of new semantic-based paddings for optimising how the text was represented in CNNs.
- The definition of a theoretical framework to provide interpretable information to what CNNs were learning internally.

Resumen

En los últimos años, los modelos de aprendizaje automático profundo (AP) han revolucionado los sistemas de procesamiento de lenguaje natural (PLN). Hemos sido testigos de un avance formidable en las capacidades de estos sistemas y actualmente podemos encontrar sistemas que integran modelos PLN de manera ubicua. Algunos ejemplos de estos modelos con los que interaccionamos a diario incluyen modelos que determinan la intención de la persona que escribió un texto, el sentimiento que pretende comunicar un tweet o nuestra ideología política a partir de lo que compartimos en redes sociales.

En esta tesis se han propuestos distintos modelos de PNL que abordan tareas que estudian el texto que se comparte en redes sociales. En concreto, este trabajo se centra en dos tareas fundamentalmente: el análisis de sentimientos y el reconocimiento de la personalidad de la persona autora de un texto. La tarea de analizar el sentimiento expresado en un texto es uno de los problemas principales en el PNL y consiste en determinar la polaridad que un texto pretende comunicar. Se trata por lo tanto de una tarea estudiada en profundidad de la cual disponemos de una vasta cantidad de recursos y modelos. Por el contrario, el problema del reconocimiento de personalidad es una tarea revolucionaria que tiene como objetivo determinar la personalidad de los usuarios considerando su estilo de escritura. El estudio de esta tarea es más marginal por lo que disponemos de menos recursos para abordarla pero que no obstante presenta un gran potencial.

A pesar de que el enfoque principal de este trabajo fue el desarrollo de modelos de aprendizaje profundo, también hemos propuesto modelos basados en recursos lingüísticos y modelos clásicos del aprendizaje automático. Estos últimos modelos

nos han permitido explorar las sutilezas de distintos elementos lingüísticos como por ejemplo el impacto que tienen las emociones en la clasificación correcta del sentimiento expresado en un texto.

Posteriormente, tras estos trabajos iniciales se desarrollaron modelos AP, en particular, Redes neuronales convolucionales (RNC) que fueron aplicadas a las tareas previamente citadas. En el caso del reconocimiento de la personalidad, se han comparado modelos clásicos del aprendizaje automático con modelos de aprendizaje profundo, pudiendo establecer una comparativa bajo las mismas premisas.

Cabe destacar que el PNL ha evolucionado drásticamente en los últimos años gracias al desarrollo de campañas de evaluación pública, donde múltiples equipos de investigación comparan las capacidades de los modelos que proponen en las mismas condiciones. La mayoría de los modelos presentados en esta tesis fueron o bien evaluados mediante campañas de evaluación públicas, o bien emplearon la configuración de una campaña pública previamente celebrada. Siendo conscientes, por lo tanto, de la importancia de estas campañas para el avance del PNL, desarrollamos una campaña de evaluación pública cuyo objetivo era clasificar el tema tratado en un tweet, para lo cual recogimos y etiquetamos un nuevo conjunto de datos.

A medida que avanzabamos en el desarrollo del trabajo de esta tesis, decidimos estudiar en profundidad como las RNC se aplicaban a las tareas de PNL. En este sentido, se exploraron dos líneas de trabajo. En primer lugar, propusimos un método de relleno semántico para RNC, que plantea una nueva manera de representar el texto para resolver tareas de PNL. Y en segundo lugar, se introdujo un marco teórico para abordar una de las críticas más frecuentes del aprendizaje profundo, el cual es la falta de interpretabilidad. Este marco busca visualizar qué patrones léxicos, si los hay, han sido aprendidos por la red para clasificar un texto.

En resumen, los principales logros presentados en esta tesis son:

- La organización de una campaña de evaluación pública para la clasificación del tema tratado en textos extraídos de la red social Twitter.
- La propuesta de distintos modelos de aprendizaje automático que clasifican el sentimiento presentes en textos de Twitter. Además, se presenta un estudio del impacto de los elementos lingüísticos como el lenguaje figurativo en la tarea.
- El desarrollo de un modelo para inferir la personalidad de una programadora o programador a partir del código fuente escrito.

- El estudio de la tarea del reconocimiento de la personalidad a partir del texto compartido en redes sociales siguiendo dos enfoques diferentes que han sido comparados. Por una parte se han desarrollado modelos basados en algoritmos de aprendizaje automático entrenados a partir de características seleccionadas manualmente, mientras que por otra parte se han entrenado modelos basados en redes neuronales en los que las características empleadas han sido seleccionadas automáticamente por el modelo.
- Se ha propuesto un nuevo modelo de relleno basado en la semántica para redes neuronales convolucionales.
- Se ha desarrollado un marco teórico que permite interpretar redes neuronales convolucionales.

Resum

En els últims anys, els models d'aprenentatge automàtic profund (AP) han revolucionat els sistemes de processament de llenguatge natural (PLN). Hem estat testimonis d'un avanç formidable en les capacitats d'aquests sistemes i actualment podem trobar sistemes que integren models PLN de manera ubiqua. Alguns exemples d'aquests models amb els quals interaccionem diàriament inclouen models que determinen la intenció de la persona que va escriure un text, el sentiment que pretén comunicar un tweet o la nostra ideologia política a partir del que compartim en xarxes socials.

En aquesta tesi s'han proposats diferents models de PNL que aborden tasques que estudien el text que es comparteix en xarxes socials. En concret, aquest treball se centra en dues tasques fonamentalment: l'anàlisi de sentiments i el reconeixement de la personalitat de la persona autora d'un text. La tasca d'analitzar el sentiment expressat en un text és un dels problemes principals en el PNL i consisteix a determinar la polaritat que un text pretén comunicar. Es tracta per tant d'una tasca estudiada en profunditat de la qual disposem d'una vasta quantitat de recursos i models. Per contra, el problema del reconeixement de la personalitat és una tasca revolucionària que té com a objectiu determinar la personalitat dels usuaris considerant el seu estil d'escriptura. L'estudi d'aquesta tasca és més marginal i en conseqüència disposem de menys recursos per abordar-la però no obstant i això presenta un gran potencial.

Tot i que el fouc principal d'aquest treball va ser el desenvolupament de models d'aprenentatge profund, també hem proposat models basats en recursos lingüístics i models clàssics de l'aprenentatge automàtic. Aquests últims models ens han

permès explorar les subtileses de diferents elements lingüístics com ara l'impacte que tenen les emocions en la classificació correcta del sentiment expressat en un text.

Posteriorment, després d'aquests treballs inicials es van desenvolupar models AP, en particular, Xarxes neuronals convolucionals (XNC) que van ser aplicades a les tasques prèviament esmentades. En el cas de el reconeixement de la personalitat, s'han comparat models clàssics de l'aprenentatge automàtic amb models d'aprenentatge profund la qual cosa a permet establir una comparativa de les dos aproximacions sota les mateixes premisses.

Cal remarcar que el PNL ha evolucionat dràsticament en els últims anys gràcies a el desenvolupament de campanyes d'avaluació pública on múltiples equips d'investigació comparen les capacitats dels models que proposen sota les mateixes condicions. La majoria dels models presentats en aquesta tesi van ser o bé avaluats mitjançant campanyes d'avaluació públiques, o bé s'ha emprat la configuració d'una campanya pública prèviament celebrada. Sent conscients, per tant, de la importància d'aquestes campanyes per a l'avanç del PNL, vam desenvolupar una campanya d'avaluació pública on l'objectiu era classificar el tema tractat en un tweet, per a la qual cosa vam recollir i etiquetar un nou conjunt de dades.

A mesura que avançàvem en el desenvolupament del treball d'aquesta tesi, vam decidir estudiar en profunditat com les XNC s'apliquen a les tasques de PNL. En aquest sentit, es van explorar dues línies de treball.En primer lloc, vam proposar un mètode d'emplenament semàntic per RNC, que planteja una nova manera de representar el text per resoldre tasques de PNL. I en segon lloc, es va introduir un marc teòric per abordar una de les crítiques més freqüents de l'aprenentatge profund, el qual és la falta de interpretabilitat. Aquest marc cerca visualitzar quins patrons lèxics, si n'hi han, han estat apresos per la xarxa per classificar un text.

En resum, els principals assoliments presentats en aquesta tesi són:

- L'organització d'una campanya d'avaluació pública per a la classificació del tema tractat en textos extrets de la xarxa social Twitter.
- La proposta de diferents models d'aprenentatge automàtic que classifiquen el sentiment presents en textos de Twitter. A més, es presenta un estudi de l'impacte dels elements lingüístics com el llenguatge figuratiu en la tasca.
- El desenvolupament d'un model per a inferir la personalitat d'una programadora o programador a partir d'el codi font escrit.

- L'estudi de la tasca de el reconeixement de la personalitat a partir de el text compartit en xarxes socials seguint dos enfocaments diferents que han estat comparats. D'una banda s'han desenvolupat models basats en algorismes d'aprenentatge automàtic entrenats a partir de característiques seleccionades manualment, mentre que d'altra banda s'han entrenat models basats en xarxes neuronals en els quals les característiques emprades han estat seleccionades automàticament pel model.

- S'ha proposat un nou model d'emplenament basat en la semàntica per a xarxes neuronals convolucionals.

- S'ha desenvolupat un marc teòric que permet interpretar xarxes neuronals convolucionals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory chapter gives the reader a general context to frame the overall work presented in this document. Hereafter, we outline the motivation that draws us to pursue this work, the objectives that we set and the contributions that we have achieved.

Moreover, the structure of the rest of the thesis is summarised, allowing the reader to have a overall hindsight of the work done.

1.1 Motivation

Text is everywhere. We communicate and interact with each other and with automatic systems, mainly using text. Even systems that allow interactions through speech commonly use text as an intermediate step to process the utterance and respond.

Furthermore, with the rise of social media, traditional media has been revolutionised, broadcasting conversations that previously belong to the private sphere and democratising the content generation. Nowadays, more than two-thirds of all internet users have a social media account, and more than 3.5 billion people have access to the internet (Ortiz-Ospina, n.d.). All these interactions can benefit from having intelligent Natural Language Processing (NLP) systems able to understand a text. Additionally, the language used in social media presents its own challenges for NLP systems. The text that can be found in this medium is not academic nor formal, and it is riddled with slang, abbreviatures and other linguistic devices that make understanding it more challenging for automatic systems.

Besides, by the time we establish the goals for this thesis, deep learning models were transforming some areas of knowledge, particularly computer vision. Nevertheless, the same trailblazing effect has not been materialised in NLP at the time. With that context, in the literature, we found a large amount of linguistic and computational linguistic models proposed with handcrafted features. However, these approaches were difficult to generalise. Nonetheless, to start exploring these topics, we proposed machine learning models for different NLP tasks, which allowed us to understand the requirements and the subtleties of this domain. Moreover, there were new NLP tasks to study, like personality recognition, which were not explored at the same depth as other NLP tasks, but that its impact in different fields deserved an intensive study.

Beyond this initial motivation, while we were working on applying deep learning algorithms —concretely Convolutional Neural Networks (CNN)— to NLP tasks, a new line of research arose fueled by the nuances of this new methodology. This motivation could be summarised as a pursue to understanding what these models are learning, which lead to proposing a new methodology for organising how text is modelled within a CNN architecture and a theoretical framework for interpreting what CNNs were learning.

1.2 Objectives and Contributions

As we previously described, our motivation was broad, and it was set in a time of significant advancements in the field.

When we started the work for this thesis, our objective was to propose and implement end-to-end deep learning models able to advance the state-of-the-art on different NLP tasks. Particularly, we were keen on studying Sentiment Analysis and Personality Recognition. Latter, we introduced to our set of objectives a new line of research that looked into deepening our understanding of how DL models were adapted to NLP tasks. Furthermore, we wanted to be able to interpret what the DL models developed for tackling NLP tasks were learning.

The contributions that we present in this thesis are the following.

1. We curated and organised an evaluation campaign that addressed Topic Classification from political tweets in the context of the 2015 Spanish general electoral cycle.

2. We proposed several Machine Learning models tackling the Sentiment Analysis task from texts extracted from social media using handcrafted features and lexical resources. We have also studied how specific linguistic devices such as figurative language affects the performance of the models proposed and the resources needed to encode these phenomenons successfully to improve the Sentiment Analysis classification.

3. We introduced a model for inferring the personality of a developer provided the source code that they have written.

4. We have addressed the novel task of Personality Recognition from social media following two different approaches. On the one hand, we proposed models based on machine learning algorithms and handcrafted features. On the other hand, we have proposed an approach using CNNs and word embeddings and compare both models.

5. A new semantic-based padding was introduced for optimising how the text was represented in CNNs. Our model was validated using a Sentiment Analysis task.

6. A theoretical framework was defined with the objective of providing interpretable information to what CNNs were learning to perform classification tasks in the NLP domain. This framework was also evaluated using a Sentiment Analysis task.

1.3 Structure of the Thesis

The content of this thesis is structured in eight chapters. The first chapter is this introduction, where the topics addressed and the main contributions of the thesis were presented. Hereafter, in this section, the structure of the thesis is outlined, and each chapter is summarised.

Chapter 2. Theoretical Framework. This chapter introduces the reader to a review of the theoretical framework required to understand the topics discussed. Four main subjects are addressed in that chapter. Firstly, the different representations used for model text for machine learning algorithms. Secondly, the machine learning algorithms used in the first part of this work are presented. Following, the reader will find a section devoted to Deep Learning, in particular to Convolutional Neural Networks. CNNs were a central machine learning algorithm applied and studied in the majority of this work. Hence, in this chapter, basic concepts about CNNs will be defined, and the caveats that CNNs present when we apply them in Natural Language Processing tasks were presented. To conclude the chapter, the metrics used to evaluate the models that we proposed are described.

Chapter 3. Related Work. Similarly, as we saw in the previous chapter, this chapter contextualises the work that we did in this thesis. In this case, we summarise the literature of the topics studied. In the first place, a review of the state-of-the-art NLP tasks addressed is presented. The tasks studied were Sentiment Analysis, Personality Recognition and Topic Classification. Secondly, we introduce a review of the public evaluation campaigns that are being carried out in the NLP community and their relevance for the advancement of the field. Finally, considering the relevance that Deep Learning models have and their ubiquitous applications in multiple domains, we present a review on the interpretability of these algorithms.

Chapter 4. Natural Language Processing Evaluation Task. In this chapter, we present an evaluation task that we designed and organised. The objective of this task was to classify the topic discussed in a tweet into one of five topics related to the Spanish 2015 electoral cycle. This chapter describes how we collect and label the dataset, how the proposals received were evaluated and briefly the outcome of this evaluation campaign.

Chapter 5. Sentiment Analysis in Social Media. This chapter introduces a classical machine learning approach to Sentiment Analysis tasks. Here, we contrast the considerable amount of ad-hoc resources needed and the process of manually selecting the features to train a Support Vector Machine model against the CNNs approach where we only rely on pre-trained word embeddings that will be explored in the following chapters. Moreover, in this chapter, we evaluated the impact of how different language devices, such as the irony, in the performance of the Sentiment Analysis models proposed and how resources available that encode human emotions can palliate these problems.

Chapter 6. Personality Recognition. Writing styles allow automatic systems to predict the personality traits of a person. In this chapter, we first tackled the problem proposing models using different ML algorithms and new linguistic features curated for the task at hand. Two domains were addressed with this approach, personality recognition from source code and texts gathered from social media. In contrast, we then applied CNNs to infer personality from social media, concretely from tweets. In both tasks, the personality traits predicted are the big five, a common psychological framework defined in Chapter 3.

Chapter 7. Study of Convolutional Neural Networks for Natural Language Processing. Despite the improvement in the performance granted by deep learning techniques, one common critique it is the lack of interpretability. In this chapter, we propose a padding strategy that improves the performance of CNNs in NLP tasks. In addition, we define a theoretical framework to identify lexical or semantical patterns learned by a trained CNN.

Chapter 8. Conclusions and Future Work. Finally, in this chapter, we summarise the principal discussions derived through the thesis and the conclusions that have been drawn. Moreover, the main lines of future work and extensions are presented. Regardless, each chapter includes its summary and conclusions. Finally, we also include the publications derived from the research presented in this work.

The thesis is structured in such a way that each chapter is self-contained. Each chapter describes the problem tackled, the methodologies, the results and a brief discussion. The reader is encouraged to read both Chapter 2 and 3 in order to have a better overview of the techniques and configurations applied in the different works carried out.

Chapter 2

Theoretical Framework

In this chapter, we describe the theoretical framework upon which we developed this work. We structured the basis of this thesis in four pillars:

- **Text representation** where we described the different ways of representing a text that can be used with machine learning and deep learning algorithms.
- **Machine Learning algorithms** where the classical machine learning algorithms used are explained.
- **Convolutional Neural Networks** where we describe comprehensively the deep learning algorithm employed to solve several NLP tasks during this thesis and later studied in depth.
- **Metrics** where we described the different metrics used for evaluating the models presented in this work.

The objective of this chapter is to provide the reader with the needed background to understand the topics discussed throughout the thesis. Therefore different levels of abstraction were applied to each section depending on the relevance of the topic for this work. In summary, this chapter aims to empower the reader with the necessary background to discuss the contributions of the thesis.

2.1 Text representations

One of the core requirements of NLP is how to represent text in a machine-understandable fashion, which is not a problem unique to NLP but common to all computation. Consequently, NLP and particularly Machine Learning and Deep Learning algorithms, require that text is represented as a vector of floats. Therefore, a method for text feature extraction is required.

In this section, we define succinctly the strategies uses to represent text through this work. Discussing in depth each method is out of the scope of this thesis, but we will highlight the key features of each methodology.

One of the fundamental aspects that each text representation tries to address is to develop strategies where similar texts have similar representations. Provided the vast size of the vocabulary in any language, text representations are commonly sparse. Moreover, a piece of text has variable dimensions depending on the number of words that it contains. Nevertheless, most of the NLP algorithms required a fixed input. Therefore, text representation must produce an input with a fixed size.

2.1.1 Bag of Words

Bag of Words (BoW) is one of the most straightforward methodologies for representing text. A vector of zeros with the size of the vocabulary is created, and each piece of text is represented, setting to 1 the positions of the words in the vector that were present in the sentence.

The work of A. Harris and Jones (2016) cited this concept in a linguistic context for the first time.

The main caveats of this approach are:

- The sparsity of the vector that creates. Only the words in the sentence[1] would have a value different than 0.
- The order of the words is lost. The order found in the representation is determined by the position of words in the vocabulary when the vector was created for the BoW; normally, this is a lexicographic order. Therefore, there is no link between the order of the vector representing a text and the order of the words in the sentence.

[1] On average, a sentence in English has between 15-20 words. Source `https://www.thoughtco.com/sentence-length-grammar-and-composition-1691948` Accessed on 19-09-2020

- The similarity among words is lost. Similar concepts or synonyms are not encoded in a BoW approach. Hence, comparing sentences using this representation loses many nuances.
- The relevance of a word in a document is not encoded. All words in a document have the same encoding. Therefore, there is no way with this approach to weighing the relevance of a concept in a document.

Subsequent text representations tackle one or more of these limitations. Despite them, BoW is still a valid representation in some scenarios considering that it is fast and easy to compute.

2.1.2 N-grams

BoW describe sets of words without order. However, a simple improvement is the n-gram representations. Here, instead of considering a word independently from its context, each sequence of n linguistic parts is considered as an element in the vocabulary to vectorise. The n-grams are selected applying a sliding window over the text.

N-grams models can select as linguistic parts words but also letters or syllables. *Bigrams* and *trigrams* are two of the most common n-grams models, and they describe a sequence of two items and three items, respectively.

The vocabulary created for n-grams representation is massive. Therefore, depending on the application, it can be compromised, considering only linguistic parts with a frequency of apparition higher than a threshold determined by the practitioner.

N-grams models consider the order of a sequence of n items, but it counts the frequency of an n-gram disregarding the importance of the sequence in the document.

2.1.3 Term frequency - Inverse document frequency

Term frequency - Inverse document frequency (tf-idf) addresses the problem of the uniformity encoding different linguistic tokens. It is commonly viewed as a re-weighting algorithm that seeks to reflect the importance of a linguistic token in the representation of a text.

Tf-idf grows proportionally to the frequency of an item in a document normalised by the number of times that this item appears in the dataset. It assumes an inverse relationship between the number of times that an item appears in a document and the number of times that a term occurs in the whole dataset. A clear example of this phenomenon is the representation of stop words –for instance, function words– that appear very frequently in all documents, but they barely contribute to the meaning of the text. Therefore its representation is weighted by a factor presented in equation 2.1, tha dissect the tf-idf weighting system.

$$\begin{aligned} tf(t,d) &= \frac{\text{Number of times the term } t \text{ appears in a document}}{\text{Number of documents}} \\ idf(t,D) &= \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with the term } t \text{ in it}}\right) \\ tf-idf(t,d,D) &= tf(t,d) \times idf(t,D) \end{aligned} \tag{2.1}$$

Several works explore the statistical interpretations of this text representation (Robertson, 2004; Roelleke and Jun Wang, 2008). Moreover, in the literature, tf-idf variants can be found.(C.-H. Huang, J. Yin, and Hou, 2011; Yun-tao, Ling, and Yong-cheng, 2005; Forman, 2008). In this work, we have used the basic tf-idf developed in the scikit-learn library (Pedregosa et al., 2011) in our models, where we applied it to both n-grams of characters and words.

2.1.4 Word embeddings

At the beginning of the decade of 2000, with the surge of neural networks, the development of technology such as GPUs, and the creation of big datasets, a new family of text representation emerged: word embeddings. Since then, the improvements in the performance that this technique has facilitated made word embeddings, and its variants, the de-facto representation for NLP, and it has fueled most of the breakthroughs in the domain.

Commonly, the first-word embedding representation can be found in Bengio, Ducharme, et al. (2003) as a by-product of training a neural network language model. This work sparked a new field of word representations that was followed shortly after by many others.(Bengio, Senécal, et al., 2003; Mnih and G. Hinton, 2007; Morin and Bengio, 2005; Collobert et al., 2011). All these models rely on the distributional hypothesis, proposed by Z. S. Harris (1954), that assumes that similar words would appear in similar contexts.

In the literature, word embeddings are defined as a dense, distributed, fixed-length word vectors built using word co-occurrence statistics as per the distributional hypothesis. (Almeida and Xexéo, 2019) The models that learn these representations in an unsupervised fashion are generally called distributed word representations models. They map each word from the training vocabulary to a Euclidean space, minimising the perplexity of the language model created, attempting to capture semantic relationships between words.

There are multiple models proposed for computing these vector space representations, but overall they can be categorised into two classes:

- *Prediction-based* models. These methods use the context of the word to learn the embedding. Among the prediction based models, one of the most popular is *word2vect* proposed by Mikolov, K. Chen, et al. (2013) and later on improved in the work of Mikolov, Sutskever, et al. (2013).
- *Count-based* models. Contrarily, these methods use global statistical information, such as word count and frequencies computed on the whole corpus. In this category of word embeddings, one of the most widespread is *GloVe* proposed by Pennington, Socher, and Manning (2014). GloVe is the model that we used predominantly in this work when we trained deep learning models. It learns word embeddings through a logbilinear regression model, which combines global matrix factorisation methods and local context window methods.

Describing the nuances of each method is beyond the scope of this work. Nevertheless, in the literature, there are comprehensive surveys (Almeida and Xexéo, 2019; Y. Li and Yang, 2018) that describe this fascinating topic.

2.2 Machine Learning Algorithms

After describing the most common word representation techniques, in this section, we briefly describe the machine learning (ML) algorithms used in some of the models proposed in the following chapters. Provided that the key contributions of this work are focused on deep learning models, particularly in Convolutional Neural Networks that are explained in detail in 2.3.1, here, the ML algorithms used will be briefly described in order to make this work self-contained.

2.2.1 Regression Models

Regression analysis is one of the simplest statistical methods for estimating the relationship between a dependent variable and an independent variable. The dependent variable is expected to be a linear combination of the independent variable.

Even though this is a family of models, the archetypical regression model is the Linear Regression model that is fitted using the Linear Least Square method, also known as Ordinary Least Squares (OLS) method. It was proposed by Legendre (1805). OLS computes the line (or hyperplane) that minimises the sum of squared distances between the true data –this is the dependent variable observed– and that line (or hyperplane) –(Wikipedia contributors, 2020e). OLS presents several caveats. It is sensitive to outliers, and it is prone to overfitting. Moreover, it assumes independence of the features; if this condition is not met and features are correlated, the method is sensitive to random errors in the observed variable; this is called multicollinearity. All these weaknesses have motivated the development of new methods.

In this work, we have proposed methods using both Ridge regression and Lasso regression models. The former addressed the multicollinearity and the overfitting, imposing a penalty on the size of the coefficients, introducing a bias on the estimator, which reduces the variance; it includes a complexity parameter that controls the amount of shrinkage and applies L_2 norm regularisation. Meanwhile, the latter also uses a shrinkage parameter including a feature selection in order to reduce the number of parameters to adjust; but in this case, it applies an L_1 norm regularisation.

2.2.2 Multilayer Perceptron

Multilayer Perceptron (MLP) is the most well-known Neural Network architecture. The perceptron was introduced in 1957. The controversial work of Minsky and Papert (1969) proved the limitations of the perceptron for learning non-linear problems but also some of their strengths.

MLP is a supervised learning algorithm with three components: an input layer, one or more hidden layers and an output layer. Given an adequate dataset as an input, it can learn non-linear functions through backpropagation(Rumelhart, G. E. Hinton, and Williams, 1986). Each of the layers of the MLP is fully connected. Therefore, a linear function maps the weighted inputs to every neuron in the next layer. Moreover, all the layers except the input layer use non-linear activation functions; the most common activation functions are sigmoids.

Among the caveats of MLP is that it requires hyper tune parameters, it also requires to define the architecture of the model appropriately – this is the number of layers and the number of neurons in each layer, and finally, the hidden layers of the MLP have a non-convex local minimum; therefore, different initialisations can lead to different solutions.

Regardless, MLP is still one of the most common used ML algorithms, and they are the foundational block of many other algorithms.

2.2.3 Decision Trees

Decision trees are a very versatile family of ML algorithms. One of the distinctive benefits that these methods present is that they are interpretable. Namely, given a prediction, it can be explained the decisions, or the paths in the tree, that the model selected. Therefore, trees are white-box models (Piryonesi and El-Diraby, 2020). Contrastingly, most of the Deep Learning models are black-box models, which drove the development of a new field of study to address the interpretability of DL. The efforts for studying explainable Artificial Intelligence models are presented in 3.3 where state of the art is presented. Moreover, in Chapter 7, we proposed a framework for CNN interpretability.

Before exploring the decision trees used in this work, we will proceed to define the characteristics of these models. Given an observation with certain features, a tree structure is built using recursive partitioning (Breiman et al., 1984). Each level of the tree encode one of the features of the observed sample, and each edge in the tree encode one of the possible variables of that feature. Finally, the leaves of the tree encode the targets to predict. The targets can be either discrete variables or continuous values (Quinlau, 1986). Provided that we were to tackle a regression problem, we select trees that can infer continuous variables. Hence we used Decision Tree Regressors (DTR).

Decision trees also present some drawbacks. One of the most common ones is that in order to learn the structure of the tree, heuristics are applied, which might lead to complex structures that do not generalise when new data is presented. Besides, they are also prone to overfitting and susceptible to biases in the data.

In order to mitigate some of these problems, Random Forest (RF) algorithm was introduced by Ho (1995). RF is an ensemble method that fits a number of decision tree classifiers during the training using subsets of the dataset. Random Forest models are less prone to overfitting. As in the previous case, both classifier and regression problems can be tackled with this algorithm; the first will be addressed with Random Forest Classifiers (RFC) and the latter using Random Forest Regressors (RFR).

2.2.4 Support Vector Machines

Finally, in this section, the last machine learning algorithm used in this work is described. Going forward, the rest of the thesis, we have delved into the use of deep learning methods, concretely Convolutional Neural Networks.

Boser, Guyon, and Vapnik (1992) proposed this algorithm that aims to classify each training example in one of two categories finding a hyperplane in an n-dimensional space. The intuition behind Support Vector Machines (SVM) is that it seeks to project data points that seem inseparable into a higher dimensional space where a hyperplane that separates them can be found. Among all the possible hyperplanes that fulfil this requirement of separating the two classes of samples, SVMs look for the hyperplane with the maximum distance between points of each class. SVMs can efficiently predict linear and non-linear classification problems by selecting a kernel function that allows computing the decision boundary in the high-dimensional space without computing the projection to that space by simply computing the inner product in the low-dimensional space. This is commonly known as the kernel trick. The most popular function used is the linear kernel, provided that the samples were linearly separable.

SVMs entails a certain level of meta training because the selection of the kernel function as well as its hyperparameters determines the effectiveness of the model. Therefore, it is crucial to design a training scheme that allows selecting the best model. The most common methodology to select these values is to carry out a grid search and validating the hyperparameters selected using a cross-validation approach.

Thus far, we have described SVMs considering the two-class problem. Nevertheless, most of the problems that we tackled were not binary classification but multiclass. The most common extension of SVMs to multiclass is to reduce the problem to several binary classifications and afterwards selecting the class to infer through a voting strategy.

Similarly, like the previous ML algorithms described, SVMs can perform classification –in that case, they are denominated as SVC– or regression problems –in which case they are abbreviated as SVR.

2.3 Deep Learning

To conclude the general review of machine learning algorithms in this section, deep learning algorithms are discussed.

The overall consensus for classifying an ML model as Deep Learning (DL) model is based on the complexity of the architecture. The increase in the number of layers but also the number of units within a layer allows the model to extract higher-level features from the input, which, in turn, allows learning functions of increasing complexity (Deng and D. Yu, 2014).

In particular, the focus of this work regarding DL models is on Convolutional Neural Networks (CNNs). Therefore, in this section, this is the algorithm described. However, the reader can find extensive literature on the topic (Goodfellow, Bengio, and Courville, 2016).

2.3.1 Convolutional Neural Networks

CNNs were proposed by LeCun (1989), but it was not until recently that CNNs began to be massively used after the success that these networks showed in computer vision (He et al., 2016; Krizhevsky, Sutskever, and G. E. Hinton, 2012). Following the momentum of CNNs in Computer Vision, they began to be used in NLP as well.

Goodfellow, Bengio, and Courville (2016) defined Convolutional Networks as neural networks that use convolutions instead of matrix multiplication in at least one of its layers. A convolution is a linear operation that takes two multidimensional arrays: an input $\vec{x} \in \mathbb{R}^{m,n}$ and a kernel $\vec{w} \in \mathbb{R}^{h,k}$ where $h < m \wedge k < n$. After applying a convolution to these two arrays, it will produce multidimensional arrays called feature maps. Therefore, each element of a feature map is obtained as result of applying the convolution across a window of words $\{\vec{x}_{0:h}, \vec{x}_{1:h-1}, \dots \vec{x}_{n-h+1:n}\}$, and it is defined as:

$$c_i = f(\vec{w} \cdot \vec{x}_{i:i+h-1} + b_i) \tag{2.2}$$

where $\vec{c} \in \mathbb{R}^{n-h+1}$ is the feature map obtained, $b_i \in \mathbb{R}$ is a bias term, and f a non linear function. The size constriction of the dimension of the kernel has several critical characteristics that enable CNNs to learn more efficiently. Moreover, the size of a kernel enhances the generalisation. A CNN model computes, during the forward phase, a convolution of the input data with a linear kernel. Then, in the backward phase, CNN learns the values of its kernels.

Regarding the application of CNNs in NLP, the work of Collobert et al. (2011) proposed for the first time an architecture using CNNs. Besides, CNNs have been applied to several NLP tasks such as Sentence Classification (Y. Zhang and Wallace, 2015; Kim, 2014), Document Ranking (Shen et al., 2014), or Causal Relation Extraction (P. Li and Mao, 2019).

Noteworthy, the most popular DL architecture for addressing NLP problems was at the time of developing this work, Recurrent Neural Networks (RNN) (Rumelhart, G. E. Hinton, and Williams, 1986; Tarwani and Edem, 2017). The recursive nature of these networks captures more naturally the recursive nature of language. In the literature, several works that compare the performance of Recursive Neural Networks (RNNs) against Convolutional Neural networks can be found (W. Yin et al., 2017; L. Zhang, S. Wang, and B. Liu, 2018; Abdalraouf Hassan and Mahmood, 2017). Nevertheless, the performance of RNNs does not surpass CNNs in every NLP task, particularly handling large texts where the vanishing or exploding gradient problems might appear (Bengio, Simard, Frasconi, et al., 1994). In addition, training an RNN does not take advantage of all the benefits of current GPUs.[2] In any case, recently, CNNs architectures have been combined in an ensemble of classifiers with recurrent neural networks to exploit the benefits of both architectures (Jin Wang et al., 2016; Xingyou Wang, Jiang, and Luo, 2016; Araque et al., 2017; T. Chen et al., 2017).

As noted above, CNNs is a particularly interesting DL model to study in the context of the NLP domain.

2.4 Metrics to evaluate the performance of the systems

In this last section of the chapter, the metrics used to evaluate the models presented in this work are summarised.

Firstly, we introduce Mean Square Error (MSE)(Wikipedia contributors, 2020c), which is defined as the average of the square errors. Therefore it is always positive, and models with MSE values closer to zero are better than models with higher MSE. Provided that we have a vector of predictions $\hat{Y} \in R^n$ the MSE is computed following the formula 2.3

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(\hat{Y_i} - Y_i)^2 \tag{2.3}$$

[2] This benchmark presents the performance training different deep learning methods with diverse hardware in the market `https://github.com/baidu-research/DeepBench` Accessed on 04-10-2020

The MSE incorporated the variance and the bias of the estimator. Moreover, the Root Mean Square Error (RMSE) is an analogous metric, though it is obtained applying the square root to the MSE.

The next metric to introduce is the Pearson Correlation Coefficient (PCC) (Wikipedia contributors, 2020d). In this case, the PCC is the bivariate correlation between two variables. Considering again $\hat{Y} \in R^n$ as a vector of predictions and $\hat{Y} \in R^n$ the observed values, the PCC is computed according to the formula 2.4

$$\rho_{\hat{Y},Y} = \frac{cov(\hat{Y},Y)}{\sigma_{\hat{Y}}\sigma_Y} \tag{2.4}$$

Where cov is the covariance, $\sigma_{\hat{Y}}$ is the standard deviation of $\hat{Y}$, and σ_Y is the standard deviation of Y.

The values for the PCC rages between ±1. If the value of ρ is +1 or -1, the correlation is totally positive or negative, respectively; however, if its value is close to 0, it indicates that there is no correlation among the two variables evaluated.

Hereafter, we present the accuracy(Wikipedia contributors, 2020a), which is one of the most intuitive metrics. It computes the number of correct samples detected over the total. In the formula 2.5 a definition of accuracy is found.

$$\begin{aligned} Accuracy &= \frac{(TP+TN)}{(TP+TN+FP+FN)} \\ \text{where: TP} &= \text{True positive; FP} = \text{False positive;} \\ \text{where: TP} &= \text{True positive; FP} = \text{False positive;} \\ \text{TN} &= \text{True negative; FN} = \text{False negative} \end{aligned} \tag{2.5}$$

True positive (TP) and true negative (TN) defines positive and negative samples correctly identified. Meanwhile, false positive (FP) identifies negative samples incorrectly assigned as positive, and false negative (FN), on the contrary, incorrectly identifies a positive sample as negative. Accuracy is a positive ratio. The closer the value of accuracy to 1, the better.

The accuracy measures the bias of the model, whilst the precision measures the variance. Equation 2.6 presents the formula for the precision in the same terms previously used for the accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

A desirable model must have both high precision and accuracy.

The recall is the next metric to be presented and defined as the true positive rate, which evaluates the ability of the system to detect positive samples, and it is computed following the formula 2.7.

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

The values for the recall are positive, and the highest the value, the better the model is considering this metric.

A common metric used to summarise the performance of a model is the F_1 score (Wikipedia contributors, 2020b). It is the harmonic mean of the precision and the recall as described in 2.8.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.8}$$

An extension of this metric is the F_β, where the β parameter is a positive real value that modulates the importance of the precision on the score computed. And it is defined in 2.9

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{2.9}$$

The F-score, as defined previously, it is applicable when the problem has only two classes. Hence, an extension of the formula is applied to multiclass setups. Basically, the multiclass *F-score* is computed as an average. Two different approaches can be followed to achieve this. The *micro* F_1 is computed globally, meanwhile the *macro* F_1 averages precision and recall for each class as defined in 2.10 and 2.11 respectively.

$$
\begin{aligned}
Recall_{micro} &= \sum_{i=0}^{N-1} \frac{TP_i}{TP_i + FN_i} \\
Precision_{micro} &= \sum_{i=0}^{N-1} \frac{TP_i}{TP_i + FP_i} \\
F_{1_{\text{micro}}} = 2 \cdot & \frac{Recall_{micro} \cdot Precision_{micro}}{Recall_{micro} + Precision_{micro}}
\end{aligned} \tag{2.10}
$$

$$
\begin{aligned}
Recall_{macro} &= \sum_{i=0}^{N-1} \frac{Recall_i}{N} \\
Precision_{macro} &= \sum_{i=0}^{N-1} \frac{Precision_i}{N} \\
F_{1_{\text{macro}}} &= \sum_{i=0}^{N-1} \frac{F_{1_i}}{N} \\
&\text{where N is the number of classes}
\end{aligned} \tag{2.11}
$$

The *micro* F_1 considers the unbalance of the classes within the dataset. Meanwhile, the *macro* F_1 ignores this and assumes that all the classes have the same impact on the overall performance of the model.

In summary, to conclude in this chapter, the theoretical framework upon which we have developed our work has been presented. This includes presenting different text representation strategies, a theoretical definition of the algorithms used, and the metrics applied to evaluate the models that we proposed using these algorithms. This review of the literature will allow the reader to delve into the rest of the work uninterruptedly with the fundamental principles covered there. Therefore, instead of repeating the foundations in each model that we propose, we will remit the reader to this chapter.

Chapter 3

Related Work

The primary objective of this thesis was to study how Deep Learning (DL) techniques, in particular, Convolutional Neural Networks (CNN), could improve the performance of Natural Language Processing Tasks (NLP). Nevertheless, initially, we explored how different Machine Learning algorithms perform in these NLP tasks. We have proposed models for solving NLP tasks using both ML and DL algorithms. Mainly, we were focused on using texts from social media because its characteristics made the task more challenging and, therefore, more suitable for the capacities of the models that we proposed. DL gained its vast popularity mainly in computer vision and sequence-to-sequence tasks. However, recently the NLP field has seen a surge in new DL models that had dispaced other approaches as the state-of-the-art models. In order to validate the models proposed, we used public evaluation campaigns which allowed us to compare our models to others developed in the same environment. This led to the development of an evaluation campaign. Secondly, considering the evolution of DL in NLP, an interest in studying the interpretability of these models has arisen. Following this trend, the second objective of this thesis was the study of CNNs applied in NLP tasks.

Accordingly, in this chapter, a review of the most relevant literature on these topics is presented. The objective here is, analogously as in the previous chapter, to provide the reader with all the information for understanding the context of our contributions. Firstly, the research on the NLP tasks that we studied is presented. Following, we introduce the relevant work on how to develop and carry a public evaluation campaign. Finally, the rising work on interpretability in Deep Learning models is shown.

3.1 Natural Language Processing Tasks

In this section, we summarise the literature of the three tasks that we tackled in this thesis. Among all possible NLP tasks to address, we choose three classification problems with entirely different characteristics. The three tasks were Sentiment Analysis (SA), Personality Recognition, and Topic Classification (TC). SA is a classical task in NLP, and the literature on the topic is profuse. Besides, it is a well-studied topic where multiple approaches have been tested, and there are extensive resources available for practitioners. Whilst PR is, at the time of writing this work, a novel task that was practically unexplored by DL approaches and where the resources available were scarce. Finally, TC is also a profusely study topic. However, we proposed a new perspective organising a task that aims to classify political topics in discussions carried out over social media.

The two tasks for which we have developed models in this work, SA and PR, strike a balance among NLP classification tasks, making them suitable for evaluating the capabilities of DL algorithms.

Following, we describe the tasks, summarise the existing literature and establish all the basic concepts allowing the reader to dive into the methodology and experimentation reported in the following chapters.

3.1.1 Sentiment Analysis

Sentiment Analysis (SA) has been widely studied in the last decade in multiple domains.

Formally, B. Liu (2012) defined SA as “the field of study that analyses people’s opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organisations, individuals, issues, events, topics, and their attributes.”

Sentiment Analysis has been a fundamental topic in the literature for its impact on multiple domains from business to social interactions. Being able to identify the opinion of users towards an entity automatically is a crucial resource for marketing. As B. Liu (2012) affirmed: “[...] opinions are central to almost all human activities and are key influencers of our behaviours. Our beliefs and perceptions of reality and the choices we make are conditioned mainly on how others see and evaluate the world."

SA is one of the classic tasks that had attracted the interest of computational linguistics deeply. The pioneering works in this field developed both supervised (Pang, L. Lee, and Vaithyanathan, 2002) and unsupervised –knowledge-based– (Peter D Turney, 2002) models. The work of Pang, L. Lee, and Vaithyanathan (2002) evaluate the performance of different machine learning classifiers predicting the polarity of movie reviews. Besides the supervised approach presented in this work, there were unsupervised approaches like the one proposed by Peter D Turney (2002) who defined part-of-speech patterns to identify subjective sentences in reviews to then estimate their semantic orientation.

Most of the works in the literature focus on developing systems able to classify the polarity of a text as positive, negative, mixed, or neutral. Nevertheless, there are models developed to identify fine-grained polarity levels as well.

In this thesis, we focus on SA tasks that predict the sentiment carried by the whole text. Nonetheless, there are more specialised tasks that focus on predicting the sentiment expressed towards an entity or a target (Cambria et al., 2013).

In Pang, L. Lee, et al. (2008), one can find a thorough study of the different techniques used to identify the polarity of a long text, such as surveys or blog posts. Also, B. Liu (2012) presents a comprehensive review of the state-of-the-art SA approaches for long written texts.

However, the research of SA using texts from social media is more recent. Twitter was released in the year 2006, and the pioneering works in this field are from 2009, when Twitter started to achieve its popularity. Early, this platform attracted the attention of the community as the vast literature in the domain proves (Barbosa and Feng, 2010; Jansen et al., 2009; O'Connor, Krieger, and Ahn, 2010; Weller et al., 2014). Many efforts have been made to transfer the knowledge obtained previously in works with normative texts to the language extracted from social media, which is characterised for being short and riddled with slang and grammatical mistakes.

The massive advance of machine learning (ML) since 2010 brought these techniques to SA tasks in texts from social media. At the beginning of the decade, practitioners applied different ML approaches such as SVM, Maximum Entropy, Naive Bayes. (Barbosa and Feng, 2010; O'Connor, Balasubramanyan, et al., 2010; X. Zhu, Kiritchenko, and S. Mohammad, 2014). Additionally, most of the approaches used expert knowledge for handcrafting features that were extracted from the text. At best, these works achieve $F_1 - score$ close to 70%. Therefore, there was still room for improvement of these proposed systems.

Vinodhini and Chandrasekaran (2012) published one of the most relevant surveys that summarised the first years of development of this field of study. Also, Hussein, 2018 published an updated survey with the latest advances.

The construction of polarity lexicons is another widely explored field of research. Lexicons were a crucial resource to use in ML systems to achieve positive results. Hence, linguists and computational linguists curated a copious number of lexicons. Since the meaning of the words depends on the context and the domain, one could find lexicons for most of the contexts. In addition, lexicons can not be automatically translated, which means they should be developed for each language. For example, there are opinion lexicons in English (B. Liu, Hu, and Cheng, 2005; Wilson et al., 2005), Spanish (Perez-Rosas, Banea, and Mihalcea, 2012), French (Bestgen et al., 2008; Abdaoui et al., 2017), or German (Clematide et al., 2010; Remus, Quasthoff, and Heyer, 2010; Waltinger, 2010). In fact, there are efforts to build lexicons for all the major languages (Yanqing Chen and Skiena, 2014) and for translating lexicons (Cui et al., 2011; Balahur and Turchi, 2012; Balahur and Turchi, 2014). Notwithstanding, building lexicons is an expensive task. Hence, even though lexicons are helpful, we might not be able to have one that covers all the languages and all the acceptions of each word.

Up to this point in the chapter, we have been discussing the classification of texts from different domains. However, one of the main caveats when handling language is its nuances which can take many forms. Figurative language such as irony or sarcasm is one of the most common devices used to express opinions in social media. Therefore, it deserves to be studied independently because it has massive implications for the performance of models that predict SA. Figurative language detection is a Natural Language Processing task on its own, and there is profuse literature that tackles this problem. Veale and Hao (2007) proposed to use language constructions as identifiers of irony. They proposed to crawl information available on websites to build a knowledge database for its semi-automatic irony detector. The authors demonstrate that approximately 20% of the similes found on the web were ironic. Still, their work could not be generalised because it relies on the language structures found and stored in their knowledge database. Most of the literature addressing the detection of figurative language exploits surface features of the language such as the syntactical order of the words, lexical properties or affective properties of the words that can be found in the text(Reyes, Rosso, and Buscaldi, 2012; Reyes, Rosso, and Veale, 2013). In addition, there are works that investigate how *hashtags* are used by used to emphasise that a message contains figurative language, in particular irony and sarcasm (Sulis et al., 2016). However, the particular task of irony and sarcasm detection is out of the scope of this work, and we will not explore the literature on this topic in more depth. Nevertheless, since SA is heavily influenced by the presence of figurative language where multiple meanings are present –the literal meaning of a sentence can contrast massively with the sentiment that it tries to convey– we will explore the pertinent literature that tackles the impact of figurative language in SA. The work of Carvalho et al. (2009) describes that the presence of verbal irony is a significant source of error (about 35% of the cases). According to M. Wiegand et al. (2010), this error can be explained because irony can be an implicit negation, and it negates what is conveyed through the literal use of the words. Carvalho et al. (2009) define verbal irony as "the rhetorical process of intentionally using words or expressions for uttering a meaning different (usually the opposite) from the one they have when used literally." Most of the approaches found in the literature create linguistic knowledge rules to tackle irony in SA using surface features. For example, Maynard and Greenwood (2014) studied thoroughly how figurative language impacts the polarity of a sentence and created hashtag[1] tokeniser and a set of rules to improve the performance of the SA model. Meanwhile, Veale and Hao (2007) developed a semi-automatic model to mine the knowledge required to identify figurative language from smilies. In their work, they proved that almost 20% of smilies were ironic. Nevertheless, these approaches cannot be generalised directly because they depend on rules created from surface structures of the language that might not be present in every domain or instance.

A complete survey on subjectivity and sentiment analysis can be found in the work of Montoyo, Martínez-Barco, and Balahur (2012).

With the rise of interest regarding the Sentiment Analysis problem with its multiple variants and applications, shared evaluation campaigns started to be organised. An evaluation campaign offers a standard environment to test different approaches. The organising entity collects and shares a dataset to train and evaluate all the participating systems. Besides, they decide the metrics considered for ranking the performance of the models presented by the participants. In some cases, they also provide infrastructure to run the trained models against a test dataset and predict the class of each example; these predictions are used to evaluate then the model. SemEval (Semantic Evaluation) is an ongoing series of evaluation campaigns started in 2003, which in 2007 started to include a sentiment analysis task (Strapparava and Mihalcea, 2007). Noteworthy, SemEval has held a standard SA task until 2017, which proves the great interest of the scien-

[1] A hashtag is a word or phrase starting with the sign # that users employ to self-identify the topic of a message in social media. Common hashtags to identify figurative language are *#irony* or *#sarcasm*

tific community in this field. However, SemEval is rather not the only evaluation campaign that addressed a SA problem. Simply to cite some other evaluation tasks, the TASS workshop has proposed different SA tasks in his case focused on the Spanish language (Villena Román et al., 2013; Villena-Román et al., 2014), Patra et al. (2015) organised a SA task considering Indian languages, and Evalita (Evaluation of NLP and Speech Tools for Italian) is a periodic evaluation campaign to assess Italian NLP systems (Barbieri et al., 2016; V. Basile et al., 2014; P. Basile et al., 2018).

Recently, state-of-the-art results in SA tasks are being achieved using Deep Learning algorithms as we explained previously, in Section 2.1.4, word embeddings transformed how to represent words. Hight dimensional vector representations of text considered not only the words but their context, which led to a richer representation of texts since semantical and syntactic information are encoded into them. In addition, as we discussed previously, word embeddings can be initialised with its parameters learned from one domain and adjusted to a new domain. This re-training allowed to get a better representation of the texts. Off-shelf embeddings, for example, those embeddings which parameters were learned from tweets gathered from Twitter can be used to analyse texts from other social media platforms such as Facebook (Pool and Nissim, 2016; Krebs et al., 2017) or Instagram (Hammar et al., 2018) adjusting the embeddings as we train the model to the new domain.

Besides how practitioners represent texts, Deep Learning algorithms have revolutionised how to predict the sentiment conveyed in a text. The conjunction of DL algorithms with embeddings representations of a text improved the state-of-the-art performance drastically, becoming a de facto solution for SA tasks. In 2.3, the principles of DL algorithms were explained, as well as a description of CNNs. Therefore, here we describe only which approaches were applied to solve SA tasks. In the last years, CNNs and Recursive Neural Networks (RNNs) gained popularity because their architecture allows them to learn relationships between words (L. Zhang, S. Wang, and B. Liu, 2018). Noteworthy, Kim (2014) proposed a CNN architecture that was successfully applied to several well-known SA tasks. Besides, the author studied the impact of the initialisation and finetuned word embeddings in a CNN architecture. Since this work, many more followed their approach proposing several improvements of the CNN architecture, such as Dynamic Convolutional Neural Networks (Kalchbrenner, Grefenstette, and Blunsom, 2014), or a combination of word-embeddings and character embeddings (CharSCNN) (Dos Santos and Gatti, 2014). The most popular RNNs used in SA are Long-Short Term Memory Networks (LSTMs) that have been used independently to predict SA (Shirani-Mehr, 2014; Tang et al., 2015; Xin Wang et al., 2015) or in an ensemble of classifiers in combination with CNNs (C. Zhou et al., 2015).

L. Zhang, S. Wang, and B. Liu (2018) wrote a survey describing the usage of DL algorithms in SA tasks. Moreover, W. Yin et al. (2017) compare the performance of CNNs and RNNs in different NLP tasks, including SA.

In spite that the focus of this thesis is on CNNs, we cannot fail to mention that current advances are lead by attention-based architectures, such as attention-based LSTMs (Y. Wang, M. Huang, Zhao, et al., 2016; Baziotis, Pelekis, and Doulkeridis, 2017; Tang et al., 2015) or BERT (Devlin et al., 2018; Sun, L. Huang, and Qiu, 2019).

To conclude this section, we want to emphasise to the reader that this summary of the literature in SA is just the tip of the iceberg. The profuse body of work of this NLP task indicates its relevance. Moreover, it justifies why we use this task as a relevant NLP task to study.

3.1.2 Personality Recognition

On the other side of the spectrum of the NLP tasks, one can find the Personality Recognition task that recently has caught the attention of the NLP community. It is a groundbreaking task that can impact many areas of our daily life.

Firstly, we formally define the task. Personality Recognition (PR) consists of identifying the personality traits of a person given texts that they wrote. A more general task would be Author Profiling (AP), whose goal is to identify the demographic features of an author. The most common demographics studied are gender, age, and personality traits. Another similar task is user identification which objective is to reveal the identity of a user. In contrast, personality recognition does not try to identify a user but its personality characteristics.

Vinciarelli and Mohammadi (2014) describe personality as “a psychological construct aimed at explaining the wide variety of human behaviors in terms of a few stable and measurable individual characteristics.”

The most widely accepted paradigm for personality description is the Big Five or Five Factor Model(Boyle, Matthews, and Saklofske, 2008b). [2] In this approach, the personality of an author could be described in terms of five traits:

[2] Even though in this work we focused on the Big Five model, there are other approaches to evaluate personality, such as the MBTI model proposed by Boyle (1995). The reader can find a thorough survey on personality recognition in Štajner and Yenikent (2020)

- *Extraversion*: Describe people who are active, assertive, energetic, outgoing, and talkative.
- *Agreeableness*: Describe people who are appreciative, kind, generous, forgiving, sympathetic, and trusting.
- *Conscientiousness*: Describe people who are efficient, organised, reliable, responsible, and thorough.
- *Neuroticism*: Describe people who are anxious, self-pitying, tense, touchy, unstable, and Worrying.
- *Openness*: Describe people who are artistic, curious, imaginative, insightful, original, and with broad interests.

The personality of a person is assessed through a questionnaire or expert interviews designed to surface the adjective that better describe them. Nevertheless, this process can not be scaled cost-effectively. Therefore automatic models can be advantageous.

Several statistics studies (Alastair J Gill and Oberlander, 2002; Alastair James Gill, 2003) showed that there is a direct correlation between personality and language. The personality of a subject is projected through language. Consequently, text can be employed to infer the personality of their author. In the literature, works that infer personality traits from essays (Argamon, Dhawle, et al., 2005), blog posts (Argamon, Koppel, Pennebaker, et al., 2007), phone interactions (Celli, Lepri, et al., 2014), social media (Plank and Hovy, 2015), and source code (F. Rangel, F. González, et al., 2016) can be found.

The work of Argamon, Koppel, Fine, et al. (2003) was a pioneer in computational PR; their work was focused on distinguishing only two traits, neuroticism and extraversion, in authors of informal texts using Support Vector Machines (SVMs) trained with handcrafted features such as function words, conjunction words, and assessment taxonomies. Likewise, a varied set of n-gram features, handcrafted features, and resources have been employed to cluster bloggers personalities (Oberlander and Nowson, 2006). Furthermore, a similar approach was followed to study the Big Five traits in both informal conversation and normative text (Mairesse et al., 2007). Noteworthy, users share a vast amount of data in social media, and relevant information can be mined from these resources. Concerning social media texts, several studies have tried to predict the personality of an author using datasets extracted mainly from Twitter (Quercia, Kosinski, et al., 2011; Celli and Rossi, 2012), and Facebook (Bachrach et al., 2012; H. A. Schwartz et al., 2013; J. Yu and Markov, 2017; Laleh and Shahram, 2017). Furthermore, the work of Youyou, Kosinski, and Stillwell (2015) has proven that computer-based personality judgments are more accurate than those made by humans. In addition, Farnadi et al. (2016) developed state-of-the-art personality prediction models and studied the variance of those models over datasets extracted from Facebook, Twitter, and Youtube. They used different content-based and context-based resources and trained several machine learning algorithms like SVMs or Decision Trees. However, they were not able to improve the performance of a model trained with data from a social media domain with data from another social media domain. Hence, there is a strong dependence on the data domain and the resources used.

As far as we know, we proposed the first deep learning model to tackle this problem. Our approach is described in Chapter 6. Only very recently, deep learning models can be found that address personality recognition in the literature achieving state-of-the-art (J. Yu and Markov, 2017; Xue et al., 2018; Mehta et al., 2019).

Noteworthy, personality recognition raises ethical concerns in different areas such as the usage of data shared on social media, how that psychographic target can be used by advertising and marketing (Štajner and Yenikent, 2020), and how obsolete psychology ideas impact different groups of the population.

In summary, personality recognition is an upcoming NLP task with much potential that has not been exploited to the same degree as other NLP tasks. However, its vast impact makes it relevant. Our rationale for selecting sentiment analysis and PR as the NLP tasks to explore in this work is because the former presents a robust literature where we can validate and explore our proposed models, where the latter is an exciting novel task that allowed us to propose new models and ideas.

3.1.3 Topic Classification of Political Themes

In this part, we discuss the state of the art of Topic Classification (TC) of political tweets, which is the research question addressed in the evaluation campaign developed and discussed in Chapter 4.

Topic classification is one of the classical problems of NLP. In the literature, one can find that this task has been tackled following a wide variety of approaches[3]. Beyond the intrinsic interest for solving this task, TC can be employed as a first step for extracting relevant information from a text (Kao and Poteet, 2007)

[3] For more information, please review the survey that can be found in the following reference Aggarwal and Zhai, 2012, Chapter 6

which magnifies the interest for having satisfying performing models. The work of Hillard, Purpura, and Wilkerson, 2008 depicts an example of how automatic classification systems can assist human annotators in labelling the topic discussed in a document.

Nevertheless, the first step to carry out Topic Classification is to agree on the relevant topics to study. Noteworthy, Conway, Kenski, and D. Wang, 2015 presented a study of the symbiotic relationship for agenda-setting between Twitter posts and traditional news beginnings analysing the time-series for topic discussion among the different channels. In order to carry on this study, the authors provided a list of topics per which tweets will be classified. Similarly, this methodology was used for identifying political influencers (Dubois and Gaffney, 2014). Conversely, to label the dataset for topic classification, the approach with a broader consensus within the scientific community was proposed by Patterson, 1980 where they already distinguished among four kinds of fundamental issues present in the media during the campaign. According to Patterson, 1980, the media's messages during the campaign can be classified in the following categories:

1. *Political issues*: dealing with the most abstract aspects of electoral confrontation.
2. *Policy issues*: dealing with sectorial policies.
3. *Personal issues*: regarding the candidates' lives and pastimes.
4. *Campaign issues*: dealing with the evolution of the campaign.

Mazzoleni, 2014 studied topic classification, assuming the taxonomy proposed by Patterson in his studies on mediatised politics. Hereafter we took Mazzoleni approach as a baseline. Besides, in the evaluation campaign that we organised, we decided to add a fifth miscellaneous category to label those communications that, even though they are utter in the political space, does not belong to the political discourse. Given the above, we have considered five categories in the development of the dataset for the evaluation campaign on Topic Classification that we would discuss in chapter 4.

By the time we considered organising an evaluation campaign in this domain, the content classification of tweets in political research has been addressed mainly on lexicon-based methods. The utility of these methodologies relies on the set of words that distinguish among the topics, such as economy or national security. In a structured text, state of the art has achieved competent results in most domains (R. Schwartz et al., 1997; Alghamdi and Alfalqi, 2015). However, TC can be particularly challenging when dealing with short texts with many grammatical mistakes found on social media (Yan Chen et al., 2012; K. Lee et al., 2011; Sriram et al., 2010). Moreover, social media has been used extensively during the last elections, and this trend is soarings. Therefore, researchers have been interested in working both on computational linguistics and social science studies to solve this problem with a growing impact on society (Gayo Avello, Metaxas, and Mustafaraj, 2011; Larsson and Moe, 2012; Zirn et al., 2016). Consequently, in order to address this new challenging scenario, novel classification methods have been developed. For instance, Sudhahar, Veltri, and Cristianini, 2015 proposed a methodology based on network graphs for uncovering word patterns. In addition, in the literature, different machine learning algorithms have been developed to predict the outcome of the elections, e.g. Support Vector Machines (SVMs) (Conover et al., 2011), or Linear Discriminant Analysis (LDA) (Quercia, Askham, and Crowcroft, 2012; Menini et al., 2017). Likewise, some works linked the outcome of the election with the sentiments expressed on Twitter (Taboada et al., 2011; Tumasjan et al., 2010; H. Wang et al., 2012).

Despite the time it has elapsed since we addressed this task and the relevance that social media has to swing the electoral vote, the models evaluated do not reflect the full capacity of modern NLP systems. Most of the work in political studies remain isolated. Therefore, there is a need for public datasets like the one we developed to address this topic and more research from the NLP community to close the gap.

3.2 Development of a Natural Language Evalution Task

This section contextualises the work that will be presented in Chapter 4. There is a concern among the community (Drummond, 2009) that the fast development of machine learning and Artificial Intelligence is being carried out without the necessary consideration to the repeatability, replicability or reproducibility; which some authors, as Hutson, 2018, claims that might lead to a replication crisis, this is, the models proposed can not be validated or generalise for different domains. Before exploring the problems that the discipline faces and the recommendations proposed, we will define the differences between these three terms: repeatability, reproducibility and replicability. Noteworthy, all of them tackle the problem of the lack of generalisation. However, there are differences between them. The Association for Computing Machinery (ACM) (Plesser, 2018) defined them as:

- **Repeatability** (Same team, same experimental setup): The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat Their own computation.

- **Replicability** (Different team, same experimental setup): The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- **Reproducibility** (Different team, different experimental setup): The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts that they develop completely independently.

Other associations, like the American Statistical Association (ASA), define these concepts slightly differently (Broman et al., 2017). Nevertheless, we will work with ACM definitions considering the area of study of this thesis. The most specific requirement is repeatability, and it is expected that researchers perform repeatability tests before submitting their achievements. Following, we have replicability, which guarantees a level of robustness of the findings presented, but it requires that researchers share the data, the code and the experimental setup. Finally, the most robust proof of an experiment is reproducibility, but in order to achieve this, the data must be available. The lack of any of these properties diminish the importance of the experimental findings and hinders the possibility of advance the research. Provided that reproducible and replicability studies might lead to negative results, and the literature has a positive publication bias (Mlinarić, Horvat, and Šupak Smolčić, 2017), most researchers are reluctant to invest their efforts in these experiments. However, there have always been voices that campaign for publishing negative results (Nosek, Spies, and Motyl, 2012; Sterling, Rosenbaum, and Weinkam, 1995), to avoid that false results persist and to improve the credibility of the literature.

Noteworthy, in order to replicate and reproduce results published, data and methodology must be available. Evaluation campaigns tackle these problems. Firstly, the dataset employed for the campaign is made available among all participants. Secondly, papers are written explaining the approaches developed, and even in some cases, the code that implemented the models is shared. Besides, all the participants are encouraged to describe their systems regardless of the results that they achieved, promoting both positive and negative results to be published.

In summary, we have established the importance of the evaluation campaigns. Furthermore, in the rest of the section, we present a brief overview of the evaluation campaigns and where the campaign that we developed fits in this landscape.

3.2.1 Evalution campaings

Following, we will explore some of the most notorious evaluation campaigns in Natural Language Processing related to this work. Provided that the number of evaluation campaigns is growing at a fast pace, and there are already a significant number of them available, this section does not pretend to be a comprehensive list of all the evaluation campaigns. On the contrary, we focused on those tasks where we have participated and what key lessons we extracted to develop our own evaluation campaign that we will describe in Chapter 4.

Previously, in 3.1.1, we have described one of the most popular evaluation campaigns in NLP, SemEval. The state of the art of several NLP tasks is evaluated in different tracks of the SemEval evaluation campaign. Earlier, we discussed the sentiment analysis campaign where we participated, but other tasks were being carried out as well. Notably, the tasks proposed every year have been changing to adapt themselves to the new challenges and the advances in Machine Learning. SemEval has been organised uninterrupted since 2010 and is sponsored by the Special Interest Group on the Lexicon of the Association for Computational Linguistics (SIGLEX)[4]. Some of the most relevant tasks proposed in the last years are Word Sense Disambiguation (Lefever and Hoste, 2010; Navigli, Jurgens, and Vannella, 2013; Moro and Navigli, 2015), Textual Similarity and Question Answering (Nakov, Hoogeveen, et al., 2017; Nakov, Màrquez, et al., 2015; Mihaylova et al., 2019), and Semantic Similarity (Agirre et al., 2014; Xu, Callison-Burch, and Dolan, 2015; Jurgens, Pilehvar, and Navigli, 2014). A relevant characteristic of the tasks proposed is multilingual, which triggers the development of NLP techniques for niche languages in research with a large number of speakers. All the tasks hitherto have been supervised tasks. Therefore, in each case, the organisers provide a gold standard as well as the training dataset to the participants to train their models. Label data require expert knowledge that guarantees that the corpora annotated has the quality required to satisfy desirable specifications for

[4] For more information, please see Wikipedia contributors, 2020f.

training a machine learning system[5]. Therefore, when we developed our campaign on topic classification, we took this nuance into account and recruited political experts for labelling the dataset.

Moreover, it is noteworthy that in recent editions, they have made their datasets and evaluation metrics publicly available for participants online through CodaLab[6]. This is an open-source platform that provides an ecosystem for conducting computational research. Having this infrastructure to evaluate all the systems that participated is a crucial step because it addresses the problem of replicability directly. The objective of this platform is also to capture the code and the data used for developing a research idea. Hence, all the competitors published their approaches in the proceedings of the workshop, but they also shared the code needed to reproduce it. We followed the same approach for our evaluating campaign, and we enabled a shared environment for executing all the proposed models. Besides, using the same amount of resources for all the models for inference and evaluation restricts its size. Therefore, making their comparison more straightforward. To sum up, SemEval is a comprehensive evaluation campaign that every year arranges competitions on some of the more critical problems in NLP, offering a shared environment to fairly compare the methodologies proposed by researchers.

Following, we will briefly describe the evaluation campaigns carried out whilst the Forum for Information Retrieval Evaluation (FIRE). As introduced in 3.1.2, we have participated in the shared evaluation campaign on personality recognition using source code. A complete description of the model proposed can be found in Chapter 6. In this section, the focus will be on the shared evaluation campaigns themselves. The objective of this forum is broader than SemEval. In addition, the tasks proposed tackle diverse and novel problems in NLP with a focus on text forensics (Potthast et al., 2019). Some of the tasks that the organisation have addressed are Authorship Identification in Source Code (Rosso et al., 2016), Hate Speech and Offensive Content Identification in Indo-European Languages (Mandl et al., 2019), Gender Identification in Russian texts (Litvinova et al., 2017), or Information retrieval from microblogs during disasters (Basu, S. Ghosh, and K. Ghosh, 2018) to just name a few of the tasks developed since 2008 when the forum was held for the first time. Noteworthy, the multilingual focus of these tasks, with a particular interest in languages spoken in Asia. The evaluation campaigns developed during FIRE follow a similar philosophy than SemEval. Therefore, we will not reiterate the same points. We only emphasise the key

[5] Adversarial examples are beneficial for training robust systems. However, even in an adversarial training regime, the system needs correct annotated examples to train the model. W. E. Zhang et al., 2020 work presents a survey on the application of adversarial examples in NLP.

[6] To read more about this project, please visit `https://competitions.codalab.org/`.

points. Firstly, the labelled dataset was made available for all the participants. Secondly, a predefined evaluation framework to test all the systems was defined. Last, the organisers evaluate participants' predictions independently, validating the performance of the models on an unseen test dataset. Unquestionably, FIRE enables the evaluation of niche or innovative multilanguage tasks in an open, collaborative manner enabling the progress towards reliable, robust models.

To conclude this section, we found the reader to the last evaluation campaign relevant for this work: IberEval. In this case, we have participated in the evaluation forum proposing a shared task that will be described in Chapter 4. IberEval is one of the evaluation campaigns organised by the Spanish Society of Natural Language Processing (*Sociedad Española de Procesamiento del Lenguaje Natural* SEPLN) The workshops that the SEPLN organises have evolved over the years, tackling relevant NLP problems in Spanish and other Iberian languages (Portuguese, Catalan, Basque and Galician). Usually, the bulk of the research is done in English, neglecting other languages; likewise, most of the evaluation campaigns consider problems in English. Hence, campaigns such as IberEval or FIRE, that addresses a broader range of languages tackling the problem of reproducibility and replicability on less served communities. To illustrate the scope of IberEval, some of the latest tasks tackled were Misogyny Identification (Fersini, Rosso, and Anzovino, 2018), Humor Analysis (Castro, Chiruzzo, and Rosá, 2018; Chiruzzo et al., 2019), or Political Analysis. Concretely, Taulé et al., 2017 organised a campaign to evaluate the performance of stance and gender prediction in tweets on Catalan independence. Following the same line of research, we organised a task of topic classification on Twitter. A detailed explanation of this evaluation campaign can be found in Chapter 4. Similarly, as the other two campaigns presented in this section, a labelled dataset is shared among the participants. There is an evaluation framework to validate the performance of all models coherently, and participants publish their research whilst the same conditions.

Therefore, it can be concluded that having evaluation campaigns has a massive impact on the development of robust research. Additionally, these campaigns curate datasets that are made available to the community. Finally, it is noteworthy that everyone is encouraged to publish a paper with their approaches, which create a propitious environment to publish positive and negative results. Besides, all participants test their approaches within the same environment. Hence, shared evaluating campaigns offer a valuable common framework to compare approaches. Moreover, suppose the evaluation campaign counts with a high volume of participants. In that case, there usually is some redundancy among the models that allow testing variability even whilst applying the same machine learning algorithm. As a final point, highlight the importance of organising and participating in public evaluating campaigns to address the challenges of developing Machine Learning models.

3.3 Deep Neural Network Interpretability

In this last section of the review of the related work for this thesis, we will discuss the interpretability of Deep Learning (DL) Algorithms.

Nowadays, DL models applied to NLP tasks has grown massively, becoming the state-of-the-art algorithms for most NLP tasks. However, these trailblazing algorithms present a common flaw that prevents them from being applied in sensitive domains where interpretability and explainability are required. This has lead to the development of a new field that aims to shed light on all DL models. The goal is to lead the advancement of the field towards Responsible Artificial Intelligence as described in the work of Arrieta et al. (2020) which is to develop models focused on fairness, model explainability and accountability.

Initially, we must present a definition of interpretability. The definition proposed by Gilpin et al. (2018) characterises interpretability as a methodology that describes the internals of a system in a way that is understandable to humans considering the cognition, knowledge, and biases of the user.

Samek, T. Wiegand, and Müller (2017) enumerates four reasons to study the interpretability: verification of the system, improvement of the performance, transfer the learnings to other domains, and interestingly comply with legislation.

The approaches proposed to achieve network interpretability can be divided into two categories, on the one hand, frameworks developed for interpreting existing models and on the other hand, deep learning models designed with interpretability systems in place. In this work, we are focus on the former approach. However, the latter presents compelling state-of-the-art models worth exploring, but it is out of the scope of this work. (H. Liu, Q. Yin, and W. Y. Wang, 2018; Snidaro, Ferrin, and Foresti, 2019; Landecker et al., 2013)

The first works on network interpretability methodologies were developed on the computer vision field where methods for visualising the most relevant patches that maximise the activation units on CNNs (Zeiler and Fergus, 2014; B. Zhou et al., 2014) or generating salient image features (Mahendran and Vedaldi, 2015; Simonyan and Zisserman, 2014) for the classification of an image were developed. One of the common caveats of these approaches was the generalisation of their findings. However, Bau et al. (2017) proposed a general framework for quantifying the interpretability of latent representations of CNNs automatically. Their approach evaluates the alignment of the most salient feature in the trained CNNs against a dataset of images labelled with semantic concepts and proposed an interpretability measure to compare the layers of the CNN. We have based the model presented in 7.2 partially in this work and extended to apply it to text.

Nevertheless, in the literature works inspecting the interpretability of NLP models have also been developed rapidly. Danilevsky et al. (2020) presented an in-depth survey with the latest developments in the field. Interestingly, one of the latest architectures, at the time of this writing, which are attention models, have proven that the weight of their inner layers can be interpretable (Vashishth et al., 2019).

Particularly relevant for the development of the approach proposed in 7.2 is the work of Jacovi, Shalom, and Goldberg (2018) where they proposed a method for understanding how CNNs classify text. They proposed a system to investigate if the filters of CNNs were learning different semantic classes of n-grams studying the saliency maps of the filters learned by the model. Moreover, they proposed a formula to discard uninformative n-grams that reduce the complexity of analysing the model pruning the problem.

As can be seen, the ability to interpret and explain the decisions proposed by deep learning models are a blooming area of research. The state-of-the-art models in this topic presented here are less grounded and more prone to evolve. Considering the surge omnipresence of AI in every aspect of our society, inspecting the decisions and the biases presented in the models that we develop is critical. We should be encouraged to innovate responsibly.

To recap the content of this chapter, we have presented a review of the literature of the different topics addressed in this thesis. We have presented the state of the art of sentiment analysis and personality recognition tasks that we proposed models to solve in chapters 5, and 6 respectively. Besides, we have presented the state of the art on topic classification for political themes and review the literature of public evaluation campaigns, setting the context for the campaign that we organised and presented in chapter 4. Finally, we have examined the literature on the interpretability of Deep Learning models in preparation of the study of CNNs applied in NLP problems presented in 7. Consequently, this chapter will be referenced during the rest of the thesis, and like the previous chapter, it allows the reader to have all the information needed to read this work.

Chapter 4

Natural Language Processing Evaluation Task

In the chapter 5, we will present the results on two shared sentiment analysis evaluation campaigns, which established an unbiased set-up where different approaches can be tested. These campaigns allowed us to evaluate a baseline for methodologies that used handcrafted features in contrast to using Deep Learning models as we proposed in this thesis to tackle that same type of problems. Similarly, in chapter 6, we evaluated the methodology proposed for personality recognition through a pair of evaluation campaigns. As we explained in Section 3.2, these tasks tackled the reproducibility and replicability of the scientific method, contributing to the advance of the fields of study.

For these reasons, we were also compelled to curate an evaluation task to address the task of Topic Classification in Political campaigns. Nowadays, politics has upended by the usage of social media. A political campaign cannot be strategised using only the traditional media. During the election cycle, both politicians and voters engage in conversations about different topics. Politicians and their campaign staff share their policy approaches and bits of the candidates' personal lives with the public through social media. Characterising the influence processes in the public space is one of the fascinating topics in political communication research. Political parties, media and citizens send messages through a complicated media network, where understanding who has the power of agenda-setting becomes critical. In this sense, social media logic has boosted more active user participation in delivering political messages, accessing more sources, and mobilising for political action. The analysis of this complex media network requires innovative research tools capable of evaluating the different elements in the political information flow (Chadwick, 2017). Provided that the reader can find an overview of the State of the Art in Section 3.2, here we will delve into the description of the task itself and the summary of the results achieved by the teams who participated.

In order to establish a robust experimental set-up to study this problem, we have proposed a shared task: the Classification of Spanish Election Tweets (COSET) task organised as part of the IberEval workshop by Giménez, Baviera, et al., 2017, which tackles the problem of topic classification of political tweets in five categories. The political background of the COSET evaluation campaign was set during one of the most uncertain electoral contests in Spain's recent political history: the December 20, 2015, General Elections. The European Elections of the previous year had consolidated two new parties in the national political landscape. Both sought to challenge the bipartisanship entrenched in Spanish democracy. For the 2015 General Elections, the campaign uncertainty, as well as the increased number of candidates with possibilities of success, made the citizenry more interested in the campaign than ever in recent history. The traditional media, particularly TV and social media, widely covered politics during the weeks before Election Day (López García and Valera Ordaz, 2017). This agitated political environment emphasised the relevance of developing a sharded evaluation task for this particular political campaign.

In summary, in this chapter, the reader will find the efforts to organise the COSET shared task with particular attention to the new dataset curated to that end and a brief overview of the results achieved by the seventeen participants.

4.1 Evaluation Framework

Hereafter, the task is defined formally, the corpus construction is outlined, highlining the nuances of the annotation process, and finally, the metrics used to evaluate the participants' models performance are described.

Firstly, the strategies adopted for corpus construction are presented, as well as how the annotation process was carried out. The curation of datasets is in itself a relevant task. Datasets allowed that different research groups developed their approaches and compared their results using the same data. In the latest years, we have witnessed a proliferation of shared tasks and evaluation campaigns with a correlation of the improvement of the methodologies that were developed to solve these problems. Nonetheless, there were not a considerable variety of political datasets in Spanish, particularly from the Spanish political environment. The works of citezappavigna2011ambient, nooralahzadeh20132012, sang2012predicting, kaczmirek2013social provide an example of the interest in collecting and studying the impact of social media on political campaigns as its own field of study. Therefore, inspired by the work the scientific community was promoting, we decided to collect a dataset for the 2015 Spanish political campaign.

In order to carry out this task, we gathered a collection of tweets from November 2, 2015, to December 21, 2015. These 50 days comprehended the following stages of the political campaign:

- 32 days that corresponded to the pre-campaign period when candidates can begin to campaign for their parties, but there are not official rallies.
- Fifteen days corresponded with the electoral campaign when all the political apparatus of a party is actively lobbying for their candidatures.
- One day that covered the reflection day. A day where no active campaign can be done.
- The election day.
- One more day after the election. This last day is particularly useful because the conversations after knowing the results on Election Day ended at midnight.

Among all the tweets generated during this period, three criteria were established to filtering tweets relevant for this task:

1. Two general terms that users employed to self-identify tweets related to the elections: #20D and 20-D.
2. The names and Twitters' handles of the four major political parties: PP, PPopular, PSOE, @PSOE, ahorapodemos, Ciudadanos; #CiudadanosCs, and Cs.
3. The names of the four prime minister candidates along with their Twitter handles: Rajoy, @marianorajoy, Pedro Sanchez; Pedro Snchez, @sanchez-

castejon Pablo Iglesias, @Pablo Iglesias y Albert Rivera Rivera; Albert Rivera.[1].

Finally, messages written in languages other than Spanish were not included in the dataset. We collected a total of 15,806,057 unlabelled tweets. However, only labelled tweets could be used for this supervised machine learning task. Accordingly, three experienced political researchers labelled a subset of this dataset. A subsample of 4.000 tweets was randomly extracted and labelled separately by each of the three coders. The complete details of this annotation process can be found in the work of Baviera, Calvo, and Llorca-Abad, 2019.

Following, we describe in detail the task proposed. At this point, it is noteworthy to remark the importance of analysing political conversations on social media and their impact on the outcome of elections. This task is carried out methodologically in a semi-supervised fashion by political researchers but increases its relevance when a new electoral cycle is approaching. The objective of the evaluation campaign that we proposed was to boost this study process carried out by political researchers every election cycle. Therefore, participants were asked to classify tweets depending on the political topic discussed. Five categories were taken into account:

1. Political Issues (PI): Tweets related to the most abstract electoral confrontation.

2. Policy Issues (PoI): Tweets about sectorial policies.

3. Campaign Issues (CI): Tweets related to the evolution of the campaign.

4. Personal Issues(PeI): In this case, the topic discussed in the tweet is the personal life and activities of the candidates.

5. Other Issues (O): The rest of the tweets that did not fit in any of the previous categories.

Summing up, the objective of the task was that given a tweet, the participants proposed a machine learning system able to predict its topic automatically.

Participants were provided with password-protected labelled data sets for training and developing their systems. Later, their systems were evaluated against a test

[1] Including the name of the political party *Podemos* –which translated in English means "we can"– was not feasible because it generated a vast quantity of noise since this word in Spanish can appear in multiple contexts unrelated to this political party

Table 4.1: Distribution of tweet for each topic and data set.

	Training	Development	Testing
PI	530 (23.64 %)	57 (22.8 %)	151 (24.2 %)
PoI	786 (35.06 %)	88 (35.2 %)	228 (36.54 %)
CI	511 (22.79 %)	71 (28 %)	136 (21.79%)
PeI	152 (6.78 %)	9 (4 %)	38 (6.09%)
O	263 (11.73 %)	25 (10 %)	71 (11.38%)
Total	2242	250	624

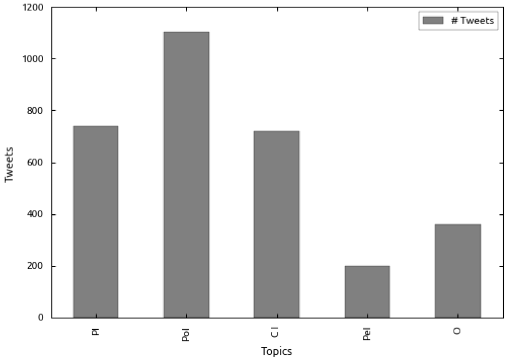


Figure 4.1: Distribution of the number of tweets for each topic in the dataset labelled.

data set. Table 4.1 presents the distribution of tweets for each topic and data set. Likewise, Figure 4.1 presents the distribution of the topics over the whole dataset (including the training, testing, and developing partitions).

In order to evaluate participants' models fairly, we needed to define an evaluation framework. Provided that the datasets were heavily unbalanced, as we just described, we proposed to rank the participants' proposals using the macro F_1-score. The F-score can be interpreted as a weighted average of the precision and the recall. In the case of the F_1-score is the harmonic mean of precision and recall as we described in 2.4. Besides, considering that the systems faced a multi-class task, we additionally need to take into account the weighted average of the F_1-score of each class since we wanted to penalise those systems that have a bias towards the most populated classes, we proposed to use the macro average F_1-score, that calculates the unweighted mean for each label as described in the equation 4.1.

$$F_{1-macro} = \frac{1}{|L|} \sum_{l \in L} F_1(y_l, \hat{y}_l) \tag{4.1}$$

Beyond the official F_1-score, used to rank each system, we evaluated the systems using Evall(Amigó et al., 2017), an open-source tool that provides a unified evaluation framework to validate the performance of different models. The best model from each team was evaluated using this tool. The results can be found in Section 4.2.

In addition, we proposed a second evaluation phase to test how well the systems were able to generalise to new examples and to perform given a more considerable amount of data points. For this second phase, the agreement of best-performing submissions was used as ground truth. Besides, the dataset labelled in this automated fashion was used for opening a new line of work for the politician researchers with whom we worked for this evaluation campaign.

Once we described the task and the evaluation framework, we proceed to review the submitted approaches. Nevertheless, even though the technical details about how we build the infrastructure lie beyond the scope of this thesis, and it was part of a collaborative effort, we want to specify that a series of scripts in Python for mining, filtering and annotating the tweets were developed, as well as the infrastructure needed to allow participants to submit and evaluate their approaches.

4.2 Overview of the Submitted Approaches

Hereafter, we present a summary of the proposed models as well as the results that each model achieved. We should note that each participant was allowed to submit up to five unique proposals in order to allow them to test different approximations. In total, seventeen teams participated in the task, and a total of thirty-nine models were submitted. We aggregate and characterise all the submissions considering three main features.

- Preprocess: whether the participants applied or not a preprocessing phase to the corpus provided by the organisation.
- Feature selection: which features were extracted from the raw or preprocessed corpus for training each model.
- Classification approaches: machine learning models used to determine the political topic discussed in a tweet.

Preprocess Most of the participants did not preprocess the tweets from the data sets that we provided them and worked with the raw data. However, among the techniques used by those teams who did preprocess the data sets were:

- Tokenisation (carried out by teams LuSer (Chuliá and S. F. Sánchez, 2017), Carl Os Duty (Alba and Pérez, 2017), UC3M (Fernandez Hernandez and Segura Bedmar, 2017), and ivsanro1 (I. Sánchez, 2017))
- Conversion to lowercase (teams LuSer (Chuliá and S. F. Sánchez, 2017), UC3M (Fernandez Hernandez and Segura Bedmar, 2017), and Electa (Juárez and Peralta, 2017)).
- Removal of several tokens such as:
  - User handles (teams LuSer (Chuliá and S. F. Sánchez, 2017), ELiRF-UPV (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017), and slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017)).
  - Numbers (teams ELiRF-UPV (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017), and slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017)).
  - Punctuation marks (teams Electa (Juárez and Peralta, 2017), slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017), and ivsanro1 (I. Sánchez, 2017)).
  - URLs (teams Electa(Juárez and Peralta, 2017), slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017), and ivsanro1 (I. Sánchez, 2017)).
  - Stopwords (teams Electa (Juárez and Peralta, 2017), UC3M (Fernandez Hernandez and Segura Bedmar, 2017), and slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017)).
  - Flooding characters (team slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017), and UC3M (Fernandez Hernandez and Segura Bedmar, 2017)).
  - Emoticons (team Electa(Juárez and Peralta, 2017)).

Features The features used to train the participants' classifiers were diverse. Participants' models used some classical features in NLP such as:

- Word n-grams (teams LuSer (Chuliá and S. F. Sánchez, 2017), LTRC II-ITH (Khandelwal et al., 2017), ConradCR (Bernath, 2017), Electa (Juárez and Peralta, 2017), Team 17 (Lafuente and Díaz-Munío, 2017), Carl Os Duty (Alba and Pérez, 2017), Citripio (Maluenda Maez and Garcìa Ferrando, 2017), LichtenwalterOlsan (Lichtenwalter and Olősan, 2017), slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017), Puigcerver (Puigcerver, 2017), and ivsanro1 (I. Sánchez, 2017)).

- Character n-grams (team LTRC IIITH (Khandelwal et al., 2017)).
- Tf-Idf (teams CD team (De la Peña Sarracén, 2017), Carl Os Duty (Alba and Pérez, 2017), LichtenwalterOlsan (Lichtenwalter and Olősan, 2017), and Puigcerver (Puigcerver, 2017)).
- Word embeddings (teams LTRC IIITH (Khandelwal et al., 2017), ELiRF (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017), atoppe (Ambrosini and Nicoló, 2017), UC3M (Fernandez Hernandez and Segura Bedmar, 2017), and MìVal (Mìguez and Valdiviezo, 2017)).
- Sentence embeddings (Team 17 (Lafuente and Díaz-Munío, 2017)).
- A multi-dimensional vector approach (team UT text miners (Gharavi and Bijari, 2017)).

Moreover, the work of LTRC IIITH (Khandelwal et al., 2017) used an extensive set of handcrafted features that included top tokens, hashtags, hashtag decomposition, mentions, and URLs, among others.

Classification approaches Finally, the models trained for the task were:

- Neural Networks. The most used model for addressing the task were models based on Neural Networks (teams LTRC IIITH (Khandelwal et al., 2017), ELiRF (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017), Team 17 (Lafuente and Díaz-Munío, 2017), and UT text miners (Gharavi and Bijari, 2017)). Chuliá and S. F. Sánchez, 2017, added normalisation techniques such as Gaussian Noise to the architecture, and Alba and Pérez, 2017, included batch normalisation with dropout in their model.
- Support Vector Machines (teams LTRC IIITH (Khandelwal et al., 2017), MìVal (Mìguez and Valdiviezo, 2017), and Citripio (Maluenda Maez and Garcìa Ferrando, 2017)).
- Random Forests (teams LTRC IIITH (Khandelwal et al., 2017), ConradCR (Bernath, 2017), and Electa (Juárez and Peralta, 2017)).

- Naive Bayes (teams slovak (Mahiques Sifres and Lyeuta Tykhovod, 2017) and ivsanro1 (I. Sánchez, 2017)).
- Logistic Regression (team Puigcerver (Puigcerver, 2017)).
- Ensamble of models. CD team (De la Peña Sarracén, 2017) proposed a combination of classifiers that included a Logistic Regression, an SVM, Naive Bayes, and a K-Nearest Neighbours classier.
- Deep learning models. Ambrosini and Nicoló, 2017 considered using Convolutional Neural Networks, Long Short Term Memory Networks, and Bidirectional Long Short-Term Memory (LSTM) Networks. Also, team UC3M (Fernandez Hernandez and Segura Bedmar, 2017) addressed this task using LSTMs and Gated Recurrent Units.
- Language models. Team 17 (Lafuente and Díaz-Munío, 2017) trained five different language models for each topic and then classified each tweet minimising the perplexity of language models.

Table 4.2: Evaluation results of each model submited, and the baseline models that were created using: term frequency-inverse document frequency (Tf-idf), random forest (RF), Support Vector Machines (SVM), and bag of words (BOW).

Team	Run	F_1 macro
ELiRF-UPV	**run 1**	**0.6482**
ELiRF-UPV	run 4	0.6400
LuSer	run 1	0.6337
ELiRF-UPV	run 3	0.6330
ELiRF-UPV	run 2	0.6233
Puigcerver	run 1	0.6176
atoppe	run 3	0.6157
atoppe	run 2	0.6065
LTRC IIITH	run 2	0.6054
LTRC IIITH	run 4	0.6049
Puigcerver	run 2	0.5997
LTRC IIITH	run 3	0.5960
LTRC IIITH	run 1	0.5959
atoppe	run 5	0.5952
Carl Os Duty	run 1	0.5902
CD team	run 1	0.5859
MìVal	run 2	0.5852
Carl Os Duty	run 2	0.5822
Electa	run 1	0.5784

atoppe	run 1	0.5745
MìVal	run 1	0.5733
Citripio	run 1	0.5676
ConradCR	run 1	0.5639
LichtenwalterOlsan	run 1	0.5590
UT text miners	run 3	0.5541
atoppe	run 4	0.5476
Puigcerver	run 3	0.5275
ivsanro1	run 1	0.5234
LTRC IIITH	run 5	0.4435
Baseline: Tf-idf & RF	-	0.4236
slovak	run 1	0.4233
UT text miners	run 1	0.3631
UT text miners	run 2	0.3341
UC3M	run 3	0.2755
UC3M	run 4	0.2755
Baseline: BOW & SVM	-	0.2644
UC3M	run 5	0.2615
UC3M	run 1	0.2571
UC3M	run 2	0.2558
Team 17	run 2	0.2446
Team 17	run 1	0.241
Baseline: Most frequent	-	0.107

Previous to the start of the competition, we developed three baselines to perform different complexity levels. The first baseline is the simplest one, and it always predicts the most common class Policy Issues (PI). The second is a traditional machine learning approach that uses a Bag of Words (BOW) to train an SVM with a linear kernel. Finally, the last baseline that we introduced applies an improved word representation following a term frequency-inverse document frequency (Tf-idf) (Clark, Fox, and Lappin, 2013) to train a Random Forest (RF) approach for classifying the training samples. To maintain fairness, none of these baselines has their hyperparameters adjusted to the task, and they were developed using the off-shelve models available in scikit-learn (Pedregosa et al., 2011). The results of all the participants' models were presented in Table 4.2.

Before diving into the analysis of the different approaches presented, it is worth noting that this is a complex task since several topics are similar and, therefore, the distinct vocabulary is shared across multiple topics. Only the first ten systems are able to achieve an F_1 macro over 0.6. The best result was obtained by J.-Á.

González, Pla, and Lluís-Felip Hurtado, 2017, who used Neural Networks and word embeddings to train their systems but also included a technique for handling the imbalance present in the data. Likewise, Chuliá and S. F. Sánchez, 2017 applied neural networks, but in this case, they used 3-grams as features and included Gaussian Noise to address the problem of overfitting. All the submissions improved the results of the most basic baseline, and the majority of the teams got better results than the three baselines proposed. However, the 25.64 % of the submissions were unable to improve the results achieved by the baseline systems.

Table 4.3: Metrics for the best performing model from each team.

Team	Accuracy	Precision	Recall	F_1 micro
ELiRF-UPV	0.6923	0.6503	0.6480	0.6482
LuSer	0.6683	0.6694	0.6214	0.6337
atoppe	0.6651	0.6815	0.5905	0.6157
Puigcerver	0.6554	0.6978	0.5846	0.6176
CD team	0.6458	0.6756	0.5550	0.5856
LTRC IIITH	0.6458	0.6214	0.5957	0.6054
Citripio	0.6410	0.6318	0.5454	0.5676
Carl Os Duty	0.6362	0.6450	0.5662	0.5902
Electa	0.6266	0.6359	0.5510	0.5784
Baseline: Majority class	0.3654	0.3654	0.2000	0.1070

In addition, as we introduced previously, the best performing model from each team was evaluated considering more evaluation metrics using the open-source tool, Evall(Amigó et al., 2017), that can be set up to evaluate multiple well-known metrics or custom ones developed for a particular evaluation campaign. In this instance, we evaluated accuracy, precision, recall and F_1 micro[2], all these metrics were described previously in 2.4. Moreover, the tool added a simple baseline system that consistently predicted the most common class and evaluated its performance.

Noteworthy, the variation in the ranking of each model considering the metric emphasises the importance of selecting a metric that characterises the problem that was being addressed in order to select the best model. Examining these results, firstly, they showed that every system in the evaluation outperforms the non-informative output, this is the predictions generated by the baseline model. Therefore, all the systems learned some patterns in the dataset. Secondly, for all

[2] F_1 micro is calculated globally and it is a less robust metric when facing unbalanced classes than F_1 macro used as the official metric.

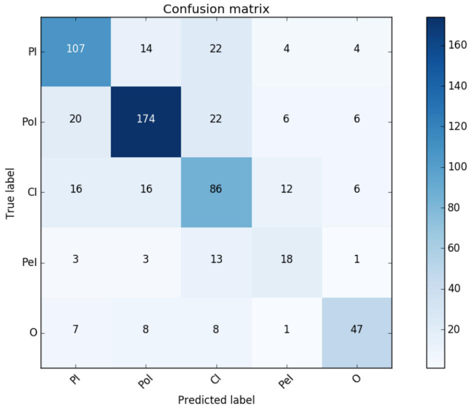


Figure 4.2: Confusion matrix for the run 1 from ELiRF team.

the metrics, the model proposed by J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017 achieved the best performance, improving 0.024 percentual points the result achieved by the second-best model considering the accuracy, and it improved 0.0145 percentual points if the F_1 micro is examined.

Following, we present the study of the best-performing systems. We have studied the confusion matrix of the three best-performing systems. These systems were the first and fourth runs from the ELiRF-UPV team (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017) and the only run from the team LuSer (Chuliá and S. F. Sánchez, 2017). Figures 4.2, 4.3, and 4.4 presented the confusion matrix between topics for each one of these three submissions. One can observe that the predictions made for the topics PI, PoI, and CI manifest a significant level of confusion between them. Remarkably, PoI is the easiest topic to classify. In contrast, PeI is the most challenging one.

As an extension of the task organised, we designed a second experimental study to test the scalability of the proposed approaches. In addition, this phase had the objective of increasing the number of annotated tweets provided that manually annotating a large dataset is time-consuming and very expensive. As a result of this phase, a larger dataset with 15.8 million tweets was constructed. Four teams submitted their runs to this evaluation phase. The best performing team in the previous phase (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017) submitted two runs, and the other teams (Ambrosini and Nicoló, 2017; Khandelwal et al., 2017; Lafuente and Díaz-Munío, 2017) submitted a run each one.

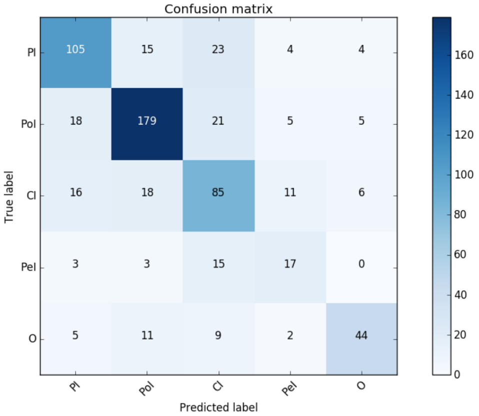


Figure 4.3: Confusion matrix for the run 4 from ELiRF team.

Table 4.4: Corpus size before and after labelling.

Corpus	Size	Percentage
Complete	15,806,058	-
Labelled	10,417,058	65.91%

We have built a silver standard using pooling techniques (Spark-Jones, 1975) over the predictions of the five runs submitted and labelled the entire dataset with the agreement of at least four of these models – that is an 80% agreement. Table 4.4 shows the corpus size before and after labelling. Furthermore, Table 4.5 presents the distribution of labels in the newly annotated dataset. As can be seen, the labelled corpus, which entails the agreement of three runs, comprises 65.91% of the original unlabelled corpus.

Finally, we evaluated the results of the teams that submitted proposals to this second phase with the newly labelled dataset. Table 4.6 presented this results. The best performing team ELiRF-UPV in the first phase also achieved the highest F_1 value in this one. On the other hand, Team 17 has improved its performance considerably. They introduced the use of fastText (Wu and Manber, 1992) for representing the tweets in this second phase, boosting their performance.

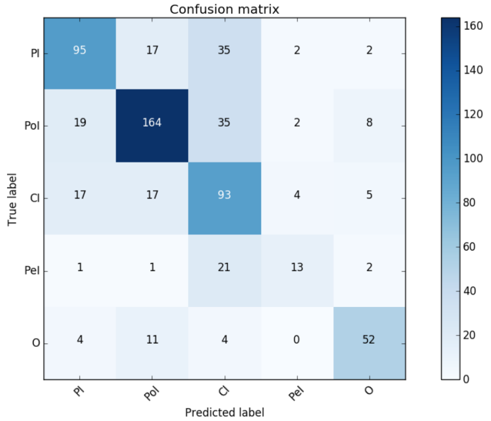


Figure 4.4: Confusion matrix for the run 1 from LuSer team.

Table 4.5: Distribution of labels (using the pooling technique).

Label	Size	Percentage
PI	2,153,236	20.67%
PoI	3,732,610	35.83%
CI	3,127,160	30.02%
PeI	581,089	5.58%
O	822,963	7.90%

4.3 Conclusions

To summarise, in this chapter, we presented the first edition of an evaluation task that we have curated. This task, denominated COSET was one of the tasks of the IberEval workshop, which was part of the annual conference held by the Spanish Society for Natural Language Processing (SEPLN in Spanish). Having these kinds of workshops has crucial importance on the advancement of the field. It set up an equal environment for the scientific community to test their approaches and to evaluate and compare them fairly. Mainly on languages less served, like Spanish. Moreover, it had the benefit of evaluating how specific resources and techniques perform in different languages. As a result of this work, a new dataset of political tweets was curated, proving very useful for new lines of research in political theories (Baviera, Calvo, and Llorca-Abad, 2019).

Table 4.6: Results in the second phase evaluation in terms of F_1 macro.

Team	$F_1 macro$
ELiRF-UPV run 1	0.9586
ELiRF-UPV run 2	0.9523
Team 17	0.9482
atoppe	0.8960
LTRC IIITH	0.8509

The objective of this task was to classify the topic discussed in a political text from social media. In particular, given a set of tweets from the 2015 election cycle, participants were asked to classify the topic discussed in them from a list of five topics that included: political issues, policy issues, campaign issues, personal issues, and other issues. Seventeen participants participated in the evaluation workshop. The best results were achieved by ELiRF-UPV (J.-Á. González, Pla, and Lluís-Felip Hurtado, 2017) who scored 0.6482 in the $F_1 macro$. They applied NNs, word embeddings, and handled the imbalance present in the data.

Provided the performance results of the participants' approaches presented here, we can confirm that topic classification from tweets is a difficult task, especially when the topics are similar. Since tweets discussing different topics might use words that belong to the same vocabulary. Therefore the importance of evaluation campaigns like this one that we just presented to help advance this task.

In summary, similarly, as we assessed the methods proposed in this thesis in evaluation campaigns like sentiment analysis and personality recognition, we acknowledge the importance of shared evaluation campaigns. Therefore, we have developed ourselves an evaluation campaign in a topic and a language not intensely investigated by the research community contributing to the advancement of the field.

Chapter 5

Sentiment Analysis in Social Media

In this chapter, we present a machine learning (ML) approach that addresses Sentiment Analysis (SA) using texts from social media. This was the first approach to the problem of SA in social media that will be explored by applying a Deep Learning approach in chapter 7. As we explained in Section 3.1.1, this task is intensively studied in NLP due to its relevance in multiple domains. Provided that we wrote a review of the literature previously, in this chapter, we focus on the models we introduced.

The goal of the models introduced in this chapter is to prove the capabilities of ML systems but also to highlight how the performance of the systems is heavily linked with lexicographical resources. We proposed two models based on Support Vector Machines (SVM) trained using classical text representations and handcrafted features. Finally, we applied these models to two SA evaluation campaigns organised within the 2015 SemEval workshops.

The capabilities of the models suggested were assessed in an evaluating campaign where two SA tasks were proposed, SemEval-2015[1]. The objective of the first task (Rosenthal et al., 2015) was to classify tweets among positive, negative, and

[1] SemEval `http://alt.qcri.org/semeval2015/` Accessed: 18-11-2019

neutral polarity. Whereas in the second task (A. Ghosh et al., 2015), we had to address the use of figurative language, and the polarity to predict was a score that varies in the range [-5..5]; this score represents the degree of the sentiment.

The two approaches developed to classify tweets shared some points for solving both tasks. For example:

- Supervised techniques were used for solving the SA tasks
- The preprocessing and feature extraction processes from the corpora were similar.
- The SA models presented address common problems when dealing with text from social media and in particular from Twitter: short texts, slang, peculiarities of the language (*hashtags*, retweets, user mentions, etc.). These characteristics can be found in both datasets evaluated.
- External lexical resources were used to extract features from the texts.

The method proposed used the Support Vector Machine formalism due to the fact of its ability to handle ample feature space and to determine the relevant features.

Both tasks used texts gathered from Twitter. Twitter[2] is a micro-blogging service, which, according to the latest statistics, it has 330 million active users, 80% outside the US, that generate 500 million tweets a day in 61 different languages.

The study (Analytics, 2009) estimates that 50.9% of tweets have some useful information that can mobilise opinions on the Internet and also in the real world. Therefore, social media users opinions have significant strategic value for different organisations. These numbers justify the great interest in the automatic processing of this information.

Moreover, in this chapter, we present a study that we carried out using one of the datasets to investigate the role of emotions in Sentiment Analysis. This work emphasises how the use of lexicons to handcraft the features has an impact on the performance of the model.

In summary, in this chapter, we present two classical computational approaches to address a SA task, and we want to stress how these solutions were tightly correlated to the lexical features handcrafted by the practitioners, which introduce a bias in the model. Besides, each new data set or domain requires evaluating

[2] About Twitter,inc. `https://about.twitter.com/company`. Accessed: 18-11-2019.

which resources would be the more valuable, hindering the automatisation of the task.

5.1 A machine Learning Approach

In this Section, we describe the models that we proposed for addressing two SA tasks. Firstly, we define the two datasets and the objectives of each task that we used for validating our models. Following, we present the models developed and the results achieved.

5.1.1 Corpora and Task Description

Hereafter the corpora used to evaluate the SA models proposed and the nuances of each task are presented.

Sentiment Analysis in Twitter

This and the following Section present the characteristics of the SA evaluations campaigns and the datasets used to evaluate the models proposed in section 5.1.2. The setup was defined by the organisers of the public evaluation campaign. Initially, we describe the objective of the task and how the models presented were evaluated. After that, we describe the data set used in each task.

The objective of this task was a classical SA classification task where each tweet must be classified depending on whether it expresses a positive, negative or neutral/objective sentiment. The annotation of the tweets was done using Amazon's Mechanical Turk[3]. At least five trained annotators label each tweet, and the annotations were consolidated using majority voting.

This task was labelled as task 10B.

The complete description of the task can be read in work by Rosenthal et al. (2015).

The corpora supplied by the organisation of the task was composed of 7,236 tweets for training, 1,242 tweets for tuning (development set) and 2,880 tweets for test-time development, which were part of the corpora used in a previous edition of the evaluation campaign (Nakov, Rosenthal, et al., 2013). For the evaluation of

[3] About Amazon's Mechanical Turk`https://www.mturk.com/` Accessed: 25-11-2019.

the candidates' systems, two sets of data were used, an official test with 2,390 tweets and a progress test with 8,987 tweets.

The performance of the models was measured using the F_1 score. This metric was described in 2.4.

Provided that our objective was to test handcrafted feature systems, we studied the characteristics of these datasets.

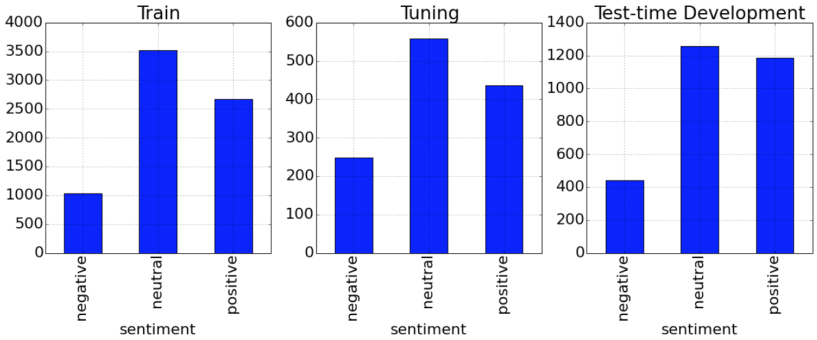


Figure 5.1: Polarity distribution studied over train, tune (dev) and test-time development corpora in Task 10B.

Figure 5.1 plots the polarity distribution over these train, tuning and test-time development corpora. On average, 16.53% of the tweets are negatives, 45.75% are neutrals, and 37.72% are positives. Vocabulary from the training corpus has 25,973 words, development corpus has 6,700 words, and test-time development corpus has 13,672 words after we deleted the stop-words.

Stop words are ubiquitous words that appear in the text. However, they encode little to no value in discerning the meaning of a sentence.(Schütze, Manning, and Raghavan, 2008) Hence, those words are commonly discarded. Prepositions and articles are common stop-words. [4]

We found that 57.57% of the words from test-time development were never seen in training. This variation in the vocabulary is particularly challenging because the system needs to deal with the out-of-vocabulary (OOV) word. As we discussed previously in Section 2.1, fine-tuned word embeddings can adjust their representation of words to improve the performance of the model. In spite of

[4] One can find a list of these words in most of the software toolkits. Besides, Rosenberg (2014) published a list of stop-words.

this advantage, we restrict ourselves to classical text representation to establish a consistent baseline of the models used before the expansion of Deep Learning in NLP.

We studied the Zipf's distribution[5] of the words from train, tune and test-time development corpora, and we find out that words with less number of synsets, i.e. less ambiguity, appear with more frequency.

Once we described the goal of this first task and the corpus available in the next Section, we proceed to follow the same scheme and present the second task in the next Section.

Sentiment Analysis of Figurative Language in Twitter

Following, we describe the second task that we used to test our models. As introduced, the corpora used for this task was riddled with figurative language that adds a level of complexity to the prediction of the sentiment conveyed in a text. Irony and sarcasm are used to mock or criticise. Therefore, these linguistic phenomenons skew the polarity towards negative sentiments. Hence, using the literal meaning of the words will not be enough to determine the sentiment of a tweet. Besides, this task is particularly interesting to emphasise the role of emotions and the impact that the usage of lexical resources have on the performance of the model. Section 5.2 is devoted to a further study of this phenomenon. Nevertheless, it is worth noting here that the handcrafted selection of features entails consequences in the performance of the model, especially in more complex scenarios.

Similarly, as in the previous task, the goal was to classify the overall sentiment in a tweet. In contrast, the dataset contained examples with creative devices such as metaphor and irony, but there was no guarantee that every tweet presents a figurative phenomenon. The presence of figurative language, nevertheless, enriched the task and skewed the overall sentiment towards negative allocations. The organisers decided to establish an 11-point scale, ranging from -5 –very negative– to +5 –very positive; zero was used to mark neutral tweets. Seven humans annotated each tweet, three of them were native English speakers, and the rest were competent English speakers. The overall sentiment was assigned as the weighted mean of all the votes, where the native speakers vote count double.

[5] Zipf's law define that the frequency of any word is inversely proportional to its rank considering the frequency (Manning, Manning, and Schütze, 1999).

To evaluate the systems submitted, the organisers used the Mean Square Error (MSE) defined in 2.4 as the official metric.

This task was labelled as task 11.

The organizers selected a set of markers of figurative language such as the *hashtags #irony* or *#sarcasm* and the words *literally* or *virtually*. As expected, the corpora for this task has a myriad of figurative language; at least 46.22% of the corpus has one of these markers.

Tweet	Polarity
"erikaekengren: From 50 to 100 degrees in less than a week #kansas" #cantwait #sarcasm	-3
Updated my router and it froze. Now I can't access the internet to google a solution. #irony #thankfulforsmartphones	-3.48
I've had a lot of wake up calls in my day, but I've always been good at hitting the snooze #metaphor #nailedit	0.22

Table 5.1: Examples of figurative tweets from task 11. The polarity value indicates the avarage polarity that was assigned to its corresponding tweet.

An example of the tweets containing *hashtags* marking figurative language can be found in Table 5.1.

The dataset was gathered for four weeks, from the 1st to the 30th of June of 2014. Tweets with less than 30 characters and tweets not written in English were discarded. Thus, this guarantee to be a monolingual task.

For this task, two corpora were provided to train the models.

- A trial corpus with 1,000 figurative tweets annotated. We were able to retrieve 925 tweets –86.6 % from the total.
- A train corpus with 8,000 tweets, of these we recover 6,928 tweets –92.5 % from the total.

However, these two corpora share some tweets. We had 7.135 unique tweets to train and tune our systems. The vocabulary of this task was more extensive than the previous one. There were 22,227 unique words without stop words.

Similarly, as in the previous task, we plotted the polarity distribution of the tweets. In this case, as was expected, figure 5.2 shows that the majority of tweets conveys a negative sentiment. Hence, the relevance of the continuous scale to predict the sense of the tweet accurately.

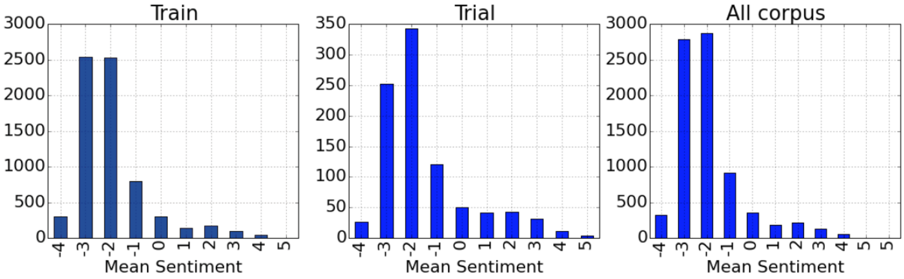


Figure 5.2: Polarity distribution in the development corpora in Task 11.

Finally, a remarkable 85.58% of tweets have at least one *hashtag*. Therefore when we designed a set of handcrafted features, *hashtag* were relevant, and we included them.

The reader can find a complete description of the task as well as an explanation of the annotation process in the work of A. Ghosh et al. (2015).

Noteworthy, both tasks are similar. However, provided that we needed to determine which features use to train the model, practitioners need to understand in-depth the domain and the peculiarities of the problem. In this example, this second task has a lot of figurative languages which bias the selection of features. Nevertheless, in the wild, it is unlikely that we have this knowledge.

5.1.2 Description of the model proposed

Hereafter, we present the models proposed and the features handcrafted for solving a SA problem.

Provided that a priori we do not have any information about which features would perform better in each dataset, we select a set of features that in the literature reports good results and then manually select the best ones for each task. We delve into these details in 5.1.2.

Nevertheless, in the first place, we developed a baseline for each task. This baseline model selected the most probable class in the training set for all the examples. The first task, 10B, the model obtained 26.49% of $F_1 - score$, 43.61% of precision, and a 43.61 % of recall. Meanwhile, the second task where the figurative language was prominent and tweets were labelled with an 11-point scale, and the baseline model achieved a 19.5% of $F_1 - score$, 36.51% of precision and a 36.51% of recall.

Feature Extraction

Following, we described the features that we considered for addressing this task. We have included ideas that we have developed in previous works in the context of two SA tasks competitions for Spanish tweets and a study of political tendency identification (Pla and Lluıs-F Hurtado, 2013; L. F. Hurtado and Pla, 2014; Pla and Lluís-F Hurtado, 2014).

The features considered were:

N-Grams This is a primary feature to include. As we described in 2.1.2 it is a representation of the words from each tweet as a feature vector that does not conserve the order. Nevertheless, we have not used a binary bag of n-grams; we use the tf-idf coefficients[6] as a weighting factor in order to ponder the importance of an n-gram in the corpora. The coefficients were computed independently for each task using the training corpus after preprocessing the datasets. Preprocessing is a common practice that prunes some information to simplify the computational cost of training and inferring results. The hypothesis is that the information remove was not critical for the evaluation. Ergo, one should do a study for each task to determine the validity of this hypothesis. Nonetheless, the computational restrictions force us to preprocess the corpora to select the most relevant features. We tokenise each tweet using an English tokeniser which splits tweets into words. Next, stop-words were deleted. Finally, we extracted character n-grams and computed its tf-idf coefficients. We studied all the combinations from 1-grams to 9-grams and selected the best combination for each task during the experimental phase.

Negation In order to predict the polarity of a tweet correctly, one needs to deal with negation appropriately. A negation particle can flip the polarity of a text. Thus, we labelled every word in a negation context with a suffix. We assumed that a negation context begins with a negation word such s "never", "no", "nothing", "none", . . . , and ends with a punctuation mark, following the approach introduced in Pang, L. Lee, and Vaithyanathan (2002). This process modifies words in negated context, and this modification is reflected in the n-grams used to represent a tweet.

[6] See 2.1.3 for a detailed description

Lexicons This is one of the most overused resources for handcrafting features. Lexicons are dictionaries of words glossed with different information. For sentiment analysis tasks, these resources have words annotated with their semantic orientation (Taboada et al., 2011). In lexicons, one can find pertinent information about terms verified by human experts. However, due to how expensive they are to create and to evaluate, there are not comprehensive lexicons; not all the words in the vocabulary appear in each lexicon, providing a complete description of all the words used in any given task. Moreover, different practitioners have created their lexicons to tackle particular nuances of the language. Therefore, several lexicons are commonly combined to extend the scope covered. In order to find a word in a lexicon, it should be converted to lowercase. Following, we included information about five different lexicons considered in this phase of the study.

1. *Pattern* (Smedt and Daelemans, 2012): This resource contains 2,888 words scored for polarity, subjectivity, intensity and reliability. The words are mostly adjectives. We used the polarity and subjectivity score assigned to a text by the Python module that encapsulates this resource[7]. These scores are the average of the adjectives that appear in the tweet. The polarity of a tweet ranges between -1.0 and +1.0 and its subjectivity between 0.0 and 1.0.

2. *AFFIN-111* (Hansen et al., 2011): In this dataset 2,477 words, including 15 phrases, were manually labelled with a sentiment strength ranging from -5 up to +5. This lexicon was constructed considering the relevance of social media in Sentiment Analysis. Hence, in this list, one can find obscene words and Internet slang, including acronyms such as WTF and LOL. We summed the polarity of every word in a tweet to get a score for the whole tweet. $\text{AFINN\_score}(W) = \sum_{w \in W} \text{AFINN}(w)$

3. *Jeffrey*(Hu and B. Liu, 2004): This is a simpler lexicon that only contains two sets of terms: 2,005 positive words and 4,782 negative words. These words were gathered from customer reviews. We computed two scores from this lexicon. The first score counts the number of positive words ($Jeffrey_{pos}(W)$) analogously the second score counts the number of negative words ($Jeffrey_{neg}(W)$).

$$Jeffrey_{neg}(W) = \sum_{w \in W} \begin{cases} 1 & \text{if } w \in Jeffrey_{neg} \\ 0 & \text{otherwise} \end{cases}$$

[7] https://www.clips.uantwerpen.be/pages/pattern-en Accessed: 17-12-2019

$$Jeffrey_{pos}(W) = \sum_{w \in W} \begin{cases} 1 & \text{if } w \in Jeffrey_{pos} \\ 0 & \text{otherwise} \end{cases}$$

4. *NRC* (Saif M. Mohammad and Peter D. Turney, 2013): Similarly, this resources contains 14,182 binary labelled as positive or negative. Likewise, we obtained a positive ($NRC_{pos}(W)$) and a negative score ($NRC_{pos}(W)$) for each tweet that we normalise by the length of the words in a tweet.

$$NRC_{pos}(W) = \frac{1}{|W|} \sum_{w \in W} \begin{cases} 1 & \text{if } w \in NRC_{pos} \\ 0 & \text{otherwise} \end{cases}$$

$$NRC_{neg}(W) = \frac{1}{|W|} \sum_{w \in W} \begin{cases} 1 & \text{if } w \in NRC_{neg} \\ 0 & \text{otherwise} \end{cases}$$

5. *SentiWordNet* (SWN)(Baccianella, Esuli, and Sebastiani, 2010): This lexical resource was created based on WordNet(Miller, 1995)[8] In this last lexicon, words are grouped considering their meaning into sets of cognitive synonyms, named *Synsets* (S). *SentiWordNet* assign three numerical scores to each one of the 117,000 synsets of WordNet. The scores are measuring the positivity ($SWN(W)_{pos}$), negativity ($SWN(W)_{neg}$), and objectivity ($SWN(W)_{obj}$)of each word. Similarly, we compute three scores for each tweet, summing the scores of the possible meanings of a word and normalising this sum by the number of synsets found in a tweet.

$$SWN_{pos}(W) = \sum_{w \in W} \frac{1}{|S_w|} \sum_{s \in S_w} \begin{cases} 1 & \text{if } w \in SWN_{pos} \\ 0 & \text{otherwise} \end{cases}$$

$$SWN_{neg}(W) = \sum_{w \in W} \frac{1}{|S_w|} \sum_{s \in S_w} \begin{cases} 1 & \text{if } w \in SWN_{neg} \\ 0 & \text{otherwise} \end{cases}$$

$$SWN_{obj}(W) = \sum_{w \in W} \frac{1}{|S_w|} \sum_{s \in S_w} \begin{cases} 0 & \text{otherwise} \end{cases}$$

[8] Wordnet is a lexical database for English, it provides a short definition of each word, examples of usage and links between related words. `https://wordnet.princeton.edu/` Accessed: 18-12-2019

Lexicon	Train	Dev
Pattern	4.28%	5.21%
AFFIN	3.14%	3.85%
Jeffrey	4.01%	4.56%
NRC	29.42%	33.26%
SWN	45.21%	51.26%

Table 5.2: Percentage of words from task 10B training and development corpora that appears in different lexicons.

Lexicon	Train
Pattern	5.69%
AFFIN	5.75%
Jeffrey	5.64%
NRC	38.09%
SWN	43.23%

Table 5.3: Percentage of words from task 11 training corpus that appears in different lexicons.

Considering that we use the scores obtained from these lexicons as features that help to represent a tweet for training the model that we proposed, we studied the coverage of these resources in our case of study. Ideally, the Percentage of words for which there has a score in a lexicon must be high to ensure the validity of the score assigned to represent a tweet.

However, Table 5.2 highlights how less than 10% of the vocabulary from the corpus can be found in the lexicons; with the exception of the lexicons NRC and SentiWordNet, but in this last lexicon, we have to deal with the semantic ambiguity of the words.

Features from Twitter Another type of relevant information is the type that models particular characteristics of the way people express themselves on Twitter. These are *hashtags*, *retweets*, *mentions* and *URLs*. Noteworthy, *hashtags* such as *#irony*, *#sarcasm* or *#not* are very relevant in order to identify the presence of figurative language in a tweet. Hence, we added four new features counting the number of *hashtags*, *retweets*, *mentions* and *URLs* that appear in a tweet.

Encoding Finally, we considered the number of capitalised words and the number of words with elongated characters[9]

In addition, we also evaluated other features, such as the number of part-of-speech tags and binary bag of words. However, during the training and development phase, these features did not deem helpful for these tasks.

Description of the models proposed

After studying the corpora of each task and selecting the features that we used to model each tweet, we used these features extracted from the text and the lexicon to train and tune different machine learning classifiers. To optimise the proposed model, we used the development corpus in the cases it was available, and we carried out a 10-cross validation when we only had a training corpus.

Since this was a baseline model used to prove the possibilities and the shortcomings of traditional machine learning algorithms, we used a Support Vector Machine Model (SVM). Provided that the reader can find the details of this method in 2.2.4, in this Section, we describe the characteristics of the model that we propose for tackling the two SA tasks considered.

The first task –task 10B– was tackled as a classification problem because there were only three discrete classes, and we applied a linear kernel for classification. Whilst, we modelled the second task –task 11– as a regression problem due to the granularity of the scores. The kernel trained was also a linear kernel but, in this case, modified to address the regression task.

Among all the features considered, we selected the best ones for each problem during development. We tuned our system using the official measure for each task; the F_1 score for the first task and the cosine distance for the second task. In the next Section, we describe the top models that we proposed and that were submitted to the competition.

We used *scikit-learn* toolkit (Pedregosa et al., 2011), and we developed a framework to define functional classification models that could be ensembled together. Therefore, in addition to single classifiers, we evaluated the performance of an ensemble of classifiers. This framework received 1 to N models and produced a prediction using the most voted category in the classification task and the mean of the predictions in the regression task.

[9] Also called flooding words. In these words, a character is repeated to emphasise or modify the meaning of a word, i.e. *boooooring*.

In conclusion, we proposed to perform the two SA tasks using SVMs models trained with a set of features that describe the tweet.

5.1.3 Experimental Setup and Results

In this Section, the best models that we proposed for both tasks are presented. We carried out an exhaustive test of the possible configurations in order to obtain a competitive classifier. Down below, we describe the characteristics of the best models for each. The organisers only allowed a submission. Hence, only the best performing model during development was submitted.

Following, the characteristics of the top four performing models for the fist task are presented:

Model 1 A linear SVM model trained with the following set of features:

- 1-gram to 6-grams of characters from a tweet.
- 1-gram to 6-grams of characters from negation labelled tweet.
- Features extracted from the lexicons Pattern, AFFIN, and SentiWordNet.
- Features extracted from Twitter.

Model 2 A linear SVM model trained with the following set of features:

- 1-gram to 6-grams of characters from negation labelled tweet.
- Features extracted from all the lexicons described previously.
- Features extracted from Twitter.

Model 3 A linear SVM model trained with the following set of features:

- 1-gram to 6-grams of characters from a tweet.
- Features extracted from the lexicons Pattern, AFFIN, and SentiWordNet.
- Features extracted from Twitter.

Model 4 An ensemble model constituted by three linear SVM models trained each one of them with the following set of features:

	Accuracy	Precision	Recall	F_{1_neg}	F_{1_neu}	F_{1_pos}	F_1
Model 1	0.6899	0.7035	0.6942	0.5014	0.7303	0.6994	**0.6826**
Model 2	0.7073	0.7201	0.7024	0.5365	0.7407	0.7209	**0.7013**
Model 3	0.6989	0.7146	0.7026	0.4802	0.7391	0.7162	**0.6901**
Model 4	0.6920	0.7074	0.6190	0.4759	0.7307	0.7060	**0.6816**

Table 5.4: Performance in development phase from our best systems in Task 10B. This task was evaluated using the F_1 scored, but here the results per class –positive (F_{1_pos}), negative (F_{1_neg}), and neutral (F_{1_neu})– are also reported.

- 1-gram to 6-grams of characters from a tweet.
- Features extracted from a lexicon. Each model had features extracted from either Pattern, AFFIN, or SentiWordNet.

The final prediction was selected by the majority voting between the predictions of each classifier.

Table 5.4 presents the performance of the best systems during the development phase. Results are decoupled to study the performance of the model per class. As expected, the model makes more mistakes for the negative tweets since this class was less represented in the training dataset; therefore, the model was not able to learn correctly the properties that differentiate this class.

Hereafter the best performing model that we proposed for the second task is presented.

Regressive Model A linear regression SVM model trained with the following set of features:

- 3-gram to 9-grams of characters from a tweet.
- Features extracted from the lexicons Pattern, AFFIN, and SentiWordNet.
- Features extracted from Twitter, including the number of figurative *hashtags*.

Table 5.5 presents the official evaluation results. The SemEval 2015 evaluation campaign was very popular, which is proven by the high participation; forty teams participated. The model that we proposed achieved the 24th position in the official test. The organisers also provided the evaluation results for the test

		Performance			
		F_1	Rank	Best	Worst
Official Test	Twitter 2015	58.58%	24	64.84%	24.80%
Progress Test	LiveJournal 2014	68.33%	28	75.34%	34.06%
	SMS 2013	60.20%	28	68.49%	26.14%
	Twitter 2013	57.05%	32	93.62%	32.14%
	Twitter 2014	61.17%	35	74.42%	32.2%
	Twitter 2014 sarcasm	45.98%	24	59.11%	35.58%

Table 5.5: Evaluation results of the model proposed for task 10B.

	Cosine	Rank	Best	Worst
Overall	0.6579	**5**	0.758	0.059
Sarcasm	0.904	**1**	0.904	0.412
Irony	0.905	4	0.918	-0.209
Metaphor	0.411	5	0.655	-0.023
Other	0.247	8	0.584	-0.025

Table 5.6: Official evaluation results in Task 11.

sets from previous editions. We also included here the result of the best and the worst system. The wide range of outcomes proves the difficulty of the task. Nevertheless, despite the variations on the systems, most were in the higher range of the results. The selection of resources used had a notable impact on the results. The model we proposed achieved the 24th position in the official rank and the 35th position in the progress test.

Analogously, table 5.6 shows the official results achieved by our system in the second task –task 11. Fifteen teams submitted their models to be evaluated. Most of these systems trained regression models or Support Vector Machines. However, each team used a wide range of lexicon resources.

	MSE	Rank	Best	Worst
Overall	3.096	**8**	2.117	6.785
Sarcasm	1.349	9	0.934	4.375
Irony	1.034	8	0.671	7.609
Metaphor	4.565	4	3.155	9.219
Other	5.235	5	3.411	12.16

Table 5.7: MSE evaluation results in Task 11.

In this second task, the model we proposed obtained notable results. Nevertheless, it was developed with the same premises and resources. The organisers disaggregate the results considering if there was figurative language present and what linguistic device was used –irony, sarcasm, metaphor– and if there was not figurative language present –other. On the overall rank, our proposed model achieved the 5th position and the first position predicting the sentiment for the subset of tweets that contained sarcasm. However, the proposed model presented a penalised performance dealing with non-figurative language achieving the 8th position in the rank.

In addition to these results, the organisers released the performance of the systems using as metric the mean square error metric (MSE). Table 5.7 presents the results that our model got considering this metric. Notably, the performance's outcome is modified by the metric used. This highlights the common critique that models are tuned for a development setup, which is more problematic in the case of models with handcrafted features, where the selection of the features to represent the input, and the election of the hyper-parameters are selected during training to improve one or several metrics, and that can lead to overfitting. This explains why when the evaluation is carried out against a new metric, the performance of the model can vary widely.

In summary, in this Section, we described the models that we presented to the 10B to the SemEval 2015 evaluation campaign and the results achieved by them.

Interestingly, even though we handled both tasks uniformly with regard to the preprocessing, feature extraction and feature representation, the models behaved very differently. Particularly, in the second task, when figurative language was present, the performance of the systems were affected drastically. Therefore, we decided to investigate the causes of this behaviour. We proposed to carry out a study focused on the impact of emotions in the detection of the sentiment in text with figurative language. In the next Section, we delve into this work.

5.2 The role of emotions in Sentiment Analysis

As we just introduced, figurative language is a challenge for predicting the sentiment conveyed in a sentence. Vast differences exist between figurative language devices. For example, irony tends to be more subtle; meanwhile, metaphor can be represented in multiple ways. Differentiating between these language devices is a challenging task, even for humans. Nevertheless, in general, sarcasm is easily detected.[10]

In this Section, we explore the role of emotions in the detection of the sentiment conveyed in a tweet. Our working hypothesis was that not all the lexicons contribute equally to the correct prediction of the sentiment; they can even penalise the performance of the systems. In order to prove this hypothesis, a series of experiments were developed to evaluate how lexicons that encode emotions affects the performance of SA models with and without figurative language.

Following, we describe the methodology proposed. The study was divided into two phases. In the first phase, we consider the impact on the performance of SA models training and evaluating the complete dataset from the second task described in 5.1.1. In the second phase, a detailed analysis of the impact of each resource is presented for each figurative device found in this task's corpus.

Provided that Section 3.1.1 contains a description of the related work for this study, we encourage the reader to revisit that chapter. However, when we proposed this experimentation, there was no in-depth study of the impact of different lexicon resources to represent a text and its diverse impact in a dataset with figurative language and with literal language. In the evaluations campaigns –like SemEval– each research laboratory proposed, as we did in the previous Section, a subset of resources to train their SA. For example, the model proposed by Farías et al. (2015) used a profuse variety of lexical resources to represent a tweet; in particular, they incorporated resources that encode emotional and psycholinguistic features in a tweet. These new resources were particularly interesting because its granularity and the nuances encoded. Therefore, when we designed this experiment, we decided to include them.

In the next sections, we focus on providing a framework to evaluate how each resource impacts the SA model. Even though we studied lexicons that encode emotions, this methodology can be expanded to any other subset of lexicons.

[10] A complete study of the impact of irony and sarcasm in sentiment analysis can be found in Farias and Rosso (2017)

5.2.1 Methodology proposed

As introduced here, we describe the methodology that we proposed to study the impact of different lexical resources in the detection of the polarity of a tweet. We considered lexicons that encode two levels of information: the polarity –positive or negative– and the emotion that is associated with a word. Interestingly, there is no common framework to describe the emotions, and each lexicon defines its own set of emotions.

Firstly, we carried out a series of experiments to determine the sentiment of a tweet using only the polarity information from lexicons. Secondly, we repeated the same set of experiments using all the categories –polarity and emotions– from each lexicon. The objective was to determine the impact of each resource on the performance. To that end, we carried out an ablative test over a subset of resources.

Nevertheless, in all the cases, tweets were tokenised, and we encode the words using the two most common word representations at the time: Bag of Words (BOW) and Term frequency - Inverse document frequency (Tf-idf).

Independently of the representation and the lexicon studied, all the experiments were developed –similarly to the model created for the competing campaign– using the scikit-learn toolkit (Pedregosa et al., 2011). Provided that our interest was to evaluate the impact of the representation using different resources that encode emotions, in all the experiments, we trained the same classifier, a Support Vector Regression Machine. This is the same system that we proposed for task 11, which achieved excellent results in the competition.

Moreover, the study was performed considering the global impact on the whole dataset. Furthermore, we also evaluated how the performance of the systems predicting the polarity is conditioned on the type of figurative language present. *Hashtags* were used for dividing literal language from figurative language. Besides, the same technique was used to separate the different kinds of figurative language. Assuming that the user self-label their own tweets correctly with *hashtags*, it is a common practice in the literature (Farias and Rosso, 2017; Sulis et al., 2016).

Three lexical resources that encode both the polarity and the emotion conveyed by a word were studied. Notably, there is no definition of the different emotions in the literature upon which all authors can agree on. Hence, even though most of the resource follows the solution proposed by Berscheid (1980), others define their subset of emotions. In sum, there are two main theories to classify emotions. The previously mentioned that defines eight basic emotions: anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. The second one is a theory developed in

Word	Category	Association
dark	anger	0
dark	anticipation	0
dark	disgust	0
dark	fear	0
dark	joy	0
dark	negative	0
dark	positive	0
dark	sadness	0
dark	surprise	0
dark	trust	0

Table 5.8: EmoLex: example of the representation of the word *dark*

the word of Ekman et al. (1987) where six basic emotions are defined: joy, fear, anger, disgust, surprise and sadness. Nevertheless, all of them encode the polarity of a word as positive or negative. Down below, the lexicons are described.

1. *NRC Word-Emotion Association Lexicon* (EmoLex)(Saif M Mohammad and Peter D Turney, 2010): This resources associates a word with its emotion or emotions and its polarity. They followed the more standard approach using the eight basic emotions proposed by Berscheid. In addition, they also encoded the polarity of the word considering two sentiments: positive and negative. This lexicon was labelled by humans. The format is shown in the example in Table 5.8, the resources is a tab-separated file where there is a one if the word belongs to the category and zero otherwise. The user can found 14,182 words annotated in this lexicon.

2. *Linguistic inquiry and word count* (LIWC) (Pennebaker, Francis, and Booth, 2001): In this resource, each word is associated with a series of emotions. In total, there are 64 different categories, and 4,485 words appear in this lexicon.

3. *Smilies* (Suttles and Ide, 2013): Finally, this resource encodes the emotion of 176 smilies with its emotion. Here, the authors developed their own subset of emotions based on the work from Berscheid (1980). In the end, the authors decided to use fifteen categories, increasing the eight basic emotions with the ones found in emotional *hashtags*. The emotions considered were: happy, laugh, love, annoyed, sad, cry, disgust, surprise, kiss, wink, tongue, sceptical, indecision, embarrassed, and evil. An example of how several emojis are labelled is presented in Table 5.9

Emoticon	Emotion
:D	Laugh
:@	Sad
;-)	Wink
3:-)	Evil

Table 5.9: Smilies classification.

In summary, we tried to determine the impact of each one of these resources, individually and combined, in the prediction of the polarity of a tweet that might contain figurative language.

As previously introduced, the experimental phase was carried out in two phases. Firstly, we investigated the impact of each resource over the whole dataset. Secondly, we focused our study on the impact of each resource depending on the type of figurative language tackled by this corpus.

Here, the preprocess was slightly different because we wanted to focus on tokenising the smilies properly in order to increase the coverage of the Smilies lexicon. A custom NLTK tokeniser (Bird, Klein, and Loper, 2009) was developed to achieve this goal. Again, stop-words were removed, and all the text was lowercased to simplify the process of searching for words in each lexicon studied. Afterwards, tweets were represented using a bag-of-words approach and tf-idf approach using one more time scikit-learn(Pedregosa et al., 2011). Experiments with each representation were carried out.

In addition to the representation of words, since our focus was studying the impact of emotions, a vector was appended, counting the number of words that a tweet had conveying an emotion. Emolex and LIWC also included a label for positive and negative words, and we did experiment with those values as well. Firstly, we only used the positive and negative labels, and then we included the information of all the categories. To recap, we set up a suite of thorough experimentation that tried all the combinations of the two most common word representations at the time with the values provided by emotional lexicons.

5.2.2 Experimental Setup and Results

Text Representation	Features	Lexicon	All corpus
BoW	Polarity	Emolex	4,6025
BoW	Polarity	LIWC	4,6074
BoW	Polarity	Emolex and LIWC	4,6022
BoW	Emotions	Smilies	4,6360
BoW	Emotions and polarity	Emolex	4,5972
BoW	Emotions and polarity	LIWC	4,5897
BoW	Emotions and polarity	Emolex and LIWC	**4,5756**
Tf-Idf	Polarity	Emolex	4,6825
Tf-Idf	Polarity	LIWC	4,6825
TF-IDF	Polarity	Emolex and LIWC	4,6796
TF-IDF	Emotions	Smilies	4,6719
TF-IDF	Emotions and polarity	Emolex	4,6487
Tf-Idf	Emotions and polarity	LIWC	4,6628
TF-IDF	Emotions and polarity	Emolex and LIWC	4,6487
Baseline	-	-	5,6720

Table 5.10: Results achieved evaluating the whole test corpus with models training with polarity and emotion lexicons using the Mean Square Error.

Table 5.10, presents the overall MSE results achieved over the whole evaluation dataset. Each row from that table describes the set of features used to train an SVM. Moreover, the performance of the baseline provided by the organisation of the task is also included in the table to facilitate the evaluation of each mode which was a Naive Bayes model trained with a bag-of-words approach.

As we described previously, our objective was to explore in detail the impact of emotions over diverse figurative language. Therefore, we explored how each assemblage of handcrafted features could help to determine the polarity of the most common figurative language devices present in this dataset. Four sets of tweets have been considered in this dataset, self-tagged by the author using *hashtags* as the literature suggested (Sulis et al., 2016). The figurative devices considered were #irony (765 tweets), #sarcasm (536 tweets), #not (981 tweets) and others (1,718 tweets). If a tweet contains more than one *hashtag*, the tweet is considered to present both figurative devices, and it will appear in both subsets.

The second experimental phase evaluates the performance of the previously described systems over these subsets of tweets. Table 5.11 presents the results achieved by each model.

Text	Features	Lexicon	#irony	#sarcasm	#not	Others
BoW	Polarity	Emolex	0,8466	0,5790	5,8184	6,8359
BoW	Polarity	LIWC	0,8471	0,5817	5,8257	6,8421
BoW	Polarity	Emolex and LIWC	0,8451	0,5788	5,8203	6,8421
BoW	Emotions	Smilies	0,8484	0,5817	5,8227	6,8407
BoW	Emotions and polarity	Emolex	0,8459	0,5787	5,8136	6,8269
BoW	Emotions and polarity	LIWC	**0,8331**	0,5780	5,8076	
BoW	Emotions and polarity	Emolex and LIWC	0,8338	0,5759	**5,7883**	**6,7972**
Tf-Idf	Polarity	Emolex	0,8781	0,5794	5,9199	6,9498
Tf-Idf	Polarity	LIWC	0,8754	0,5821	5,9148	6,9451
TF-IDF	Polarity	Emolex and LIWC	0,8756	0,5796	5,9173	6,9460
TF-IDF	Emotions	Smilies	0,8748	0,5813	5,9041	6,9355
TF-IDF	Emotions and polarity	Emolex	0,8770	0,5800	5,9101	6,9374
Tf-Idf	Emotions and polarity	LIWC	0,8583	0,5782	5,8972	6,9265
TF-IDF	Emotions and polarity	Emolex and LIWC	0,8586	**0,5755**	5,8778	6,9054

Table 5.11: Results achieved evaluating each type of figurative language found in the test corpus with models trained with polarity and emotion lexicons optimising the Mean Square Error.

In summary, as these results show, the inclusion of new resources do not guarantee a significant improvement in the performance of the model trained.

Hereafter, we analyse the results that the different models presented achieved. The reader can observe that all the models showed a performance over the baseline. Therefore, the SVMs trained using lexicons and word representation learned a more accurate solution than the baseline. Hence, there is information acquired from these handcrafted resources, which prove them as a valuable tool to predict the sentiment of a tweet because the information encoded is useful. Overall, adding more resources improves the performance of the model. Emotions, and not only polarity categories, help training a better system, regardless we are dealing with literal or figurative language –even the *smilies* lexicon, which has low coverage improves the sentiment analysis. However, when we disaggregate the results by the type of figurative language present, we found notable differences. For example, the best performing model for the whole test set is the model trained using all the labels present in EmoLex and LIWC –polarity and emotions– and a bag of words representation. Besides, this model is also the best for the *#not* and the set of tweets without figurative language. Provided that these two sets correspond to 67,75% of the whole test dataset, an improvement in the performance in this set has a massive impact on the overall performance of the model. Although, when we investigate these results in detail, we can see that tweets with the *hashtag #sarcasm* were better classified when the text was represented using a Tf-Idf approach. On the other hand, the model that achieved the best results for tweets with the *hashtag #irony* was trained also using a bag of words representation but only using lexicon LIWC.

Unquestionably, there is an intertwined relationship between the text representation, the lexical resources, and the performance of the model for different linguistic devices, broadly figurative and literal language. Therefore, the impact that these decisions have on the performance of the model might be significant. In the wild, we cannot evaluate the effect on each linguistic device because we do not have that information. Hence, the researcher might be including their bias in selecting resources that worked previously on a different setup.

5.3 Conclusions

In this chapter, we have presented two SVM approaches trained to predict the polarity of a text extracted from social media. The models proposed were trained using simple word representations –namely bag of words and tf-idf– and lexical resources. All the models presented achieved better results than the baseline, and some of them could be seen in production systems because their performance is acceptable for specific domains. We also carried out a study in depth of the impact of lexicons that encode emotions in the sentiment analysis of datasets with figurative language.

Our focus was to study the consequences entailed by the use of handcrafted lexical resources since this phenomenon was not intensely studied in the literature. The more relevant is the lack of generalisation that these systems present. Provided that we have a model with satisfactory performance for a dataset, we can not extrapolate this performance to another dataset where the coverage of the lexical resources might vary. In addition, even though the SVMs are a white-box model that allows explicability, evaluating the impact of each lexicon is tedious and requires a thorough test suite to validate the impact on the models. Moreover, lexical resources are incomplete. They did not provide a label for each word verified by a human. Researchers have created their lexical resources to tackle different problems or approaches. Therefore, there is a myriad of resources to study. Also, one can only find complete resources for languages with a large number of speakers. A lexical based approach perpetuates the lack of machine learning approaches for minority languages.

For the rest of the thesis, we propose to use Deep Learning models to address these problems. Firstly, because as the literature proved, at the time of writing this work, the performance of DL models are superior to SVMs models, and secondly, because they tackled most of the problems introduced described. Nonetheless, this approach presents a different collection of difficulties that we will describe in the following chapters.

Chapter 6

Personality Recognition

This Chapter continues the work initiated previously on Sentiment Analysis in Chapter 5, presenting two machine learning models to address a novel problem in NLP: personality recognition (PR). Henceforward, we discuss the two models that we developed for participating in two different evaluation campaigns. Moreover, we propose an innovative approach using deep learning models for solving PR.

The models developed were validated in two public evaluation campaigns. The first campaign where our models were tested, which we will discuss in this Chapter, (F. Rangel, F. González, et al., 2016), aimed to predict personality traits from source code. As we discussed in 3.1.2, there were previous works that aimed to predict personality from a text. Nevertheless, this evaluation campaign was significantly groundbreaking since it was the first time when the objective was to determine the personality of developers from the source code they wrote. This setup is particularly challenging since source code permits less flexibility to the writer. However, some parts of the source code, like comments or variable names, can exhibit some personality traits. Therefore, we have addressed this problem as an NLP task. Noteworthy that previous studies (Pabón et al., 2016) have already proven the impact of the personality traits in the behaviour of developers in the FLOSS community[1].

[1] Free/Libre Open Source Software `https://www.gnu.org/philosophy/floss-and-foss.en.html` Accessed: 23-08-2020

Next, we discuss the second personality recognition task where we participated (F. Rangel, Rosso, et al., 2015). In this case, it is an evaluation campaign that aimed to assess the capabilities of the participant systems classifying the personalities of Twitter users using their tweets. Therefore, here we were using texts extracted from this social media. Twitter(*Twitter Investor Relations* n.d.) is a microblogging service that, according to the latest statistics, has 186 million daily active users, more than 80% outside the US, and there are tweets in more than 40 different languages. These numbers justify the great interest in the automatic processing of this information. Nowadays, users share a vast amount of data on social media, and relevant information can be mined from these resources. Several NLP tasks are devoted to this matter, like Author Profiling (AP). As we introduced in 3.1.2, AP also covers other demographic features such as age and gender that were also considered in the second evaluation campaign that we discuss in this Chapter. We do want to stress here the dangers of developing automatic AP tools without mitigations in place. These systems can contribute to reinforcing bias against women, trans people and disregard entirely in the current setup non-binary people. However, this is out of the scope of this thesis, and we will not delve into these implications.

Following, using the dataset from the second task described, we introduced a new approach to tackle personality recognition using convolutional neural networks. This was the first time that this task was addressed following a deep learning approach. Previously, both our work and the work found in the literature used handcrafted features curated by psychologists and experts on author profiling. Nevertheless, with the development of the technology and the resources to train deep learning models, the feature selection used to classify a sample can be delegated to the model itself. During training, more important features are considered, and the least relevant are discarded as we discussed in Section 2.3.1 which allowed us to employed word embeddings directly to predict the personality traits of a user.

To summarise, in this Chapter, we discuss the main characteristics of PR. Since we already presented the state-of-the-art regarding this problem in 3.1.2, we will not repeat it here. Following, we introduced our contribution to two different PR tasks, the first predicting personality traits from source code and the second using texts from social media. Lastly, we presented the first approach to PR using deep learning models, in particular, a model based on Word Embeddings (WE) and Convolutional Neural Networks (CNNs).

6.1 An Introduction to Personality Recognition

In this section, we briefly review the basic definition of the Personality Recognition task and remind the reader that a review of the literature can be found in Section 3.1.2.

PR is one of the emerging research areas in NLP, which seeks to classify the personality traits of the author of a text. Norman (1963) proposed a taxonomy for describing the personality along with five dimensions that are known as *Big Five*, which are:

- Agreeableness.
- Conscientiousness.
- Extroversion.
- Openness to experience.
- Emotional Stability.

Moreover, this work determined that our personality traits have a strong in influence on our individual behaviour. In addition, the work carried out by Alastair James Gill (2003) outline that the personality is projected through the language. Therefore, by exploiting different kinds of NLP techniques, it is possible to infer the personality of the author of a text, and we validated this work in this Chapter using automatic approaches.

In this work, we tackle Author Profiling problems which try to determine the author's demographic features or personality traits. In the literature, this problem has been addressed with medium or lengthy texts where it is more likely to find statistically significant features that identify the author. However, we worked with restricted text from source code and using short texts from Twitter. Statistical methods used require a massive amount of data to train the models correctly. Therefore, convergence is a problem in the systems that we were building. The issue of lacking sufficient data to classify users is tackled differently in each task and is discussed in the following Sections.

6.2 Personality Recognition from Source Code

In this section, we discuss the machine learning models that we proposed for predicting the personality traits of the author of a snippet of code. The models presented were validated in the evaluation campaign (F. Rangel, F. González, et al., 2016) developed within the framework of the PAN/CLEF evaluation lab. Over more than ten years, PAN has been developing evaluation campaigns focused on text forensics, becoming one of the main forums of digital text forensic research (Rosso et al., 2016). The objective of the Personality Recognition from Source Code evaluation campaign that we used to validate our models was to predict the personality of developers given a collection of their source code. As we introduced previously, the Five-Factor Theory or Big Five (Boyle, Matthews, and Saklofske, 2008a; Costa Jr and McCrae, 2008; Norman, 1963) was the approach followed. Therefore, the models were required to predict five traits: Agreeableness (A), Conscientiousness (C), Extroversion (E), Openness to experience (O), and emotional stability/Neuroticism (N). Each trait was labelled within a continuous range. The models developed were evaluated by the organisers using two metrics: the average Root Mean Squared Error (RMSE) as well as the Pearson Product-Moment Correlation (PC). The reader can find these metric's definitions in Section 2.4.

We proposed a model for a groundbreaking new task. This was the first time that ML models are trained to learn personality traits. Before this evaluation campaign was defined, there was not a dataset to tackle this problem.

Hereafter, we present the model that we created to address this task.

Firstly, we describe this unique corpus. The organisers have gathered 2492 source code programs written in Java by 70 students of computer science. Moreover, the personality traits for each person who shared their code were included in the dataset. The values for the personality traits were obtained based on the 25-item BFI questionnaire called Big Five locator that each participant completed. The values for each attribute could range between 0 and 100. However, in this corpus, the extremes values –from the range [0, 20) and [80, 100]– were not present. For the training set, we had access to the code samples and the personality traits of 49 people, and the remaining 21 were held to validate the results.

In Table 6.1, a summary of the total number of training and test samples are shown.

Provided this the distribution of this dataset and the scarcity of the data to train models, we work under two hypotheses.

Dataset	Source Code Programs	Number of Authors
Train	1,741	49
Test	751	21

Table 6.1: Dataset distribution among the splits.

- Consider each source of code and the personality of a given person as an independent sample. Following this approach, we can train with 1741 samples. We call this a *code-based* (CB) approach.
- Consider all the samples of code for an author. This approach restricts the number of training samples to only 70. We call this an *author-based* (AB) approach.

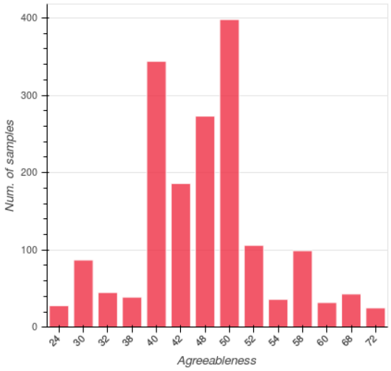


Figure 6.1: Number of code samples for each value of Agreeableness to classify (Code-based approach).

To illustrate the problem that we are trying to address with these two approaches, we plot the distribution of one of the personality traits in the training dataset. Figures 6.1 and 6.2 show the distribution of the number of samples available for the trait Agreeableness following the code-based approach in the first figure and an author-based approach in the second one. The other four traits presented a similar distribution. The sparsity depicted in 6.2 indicates that we might not have enough samples to train a machine learning model. However, if we consider each piece of code as an independent training sample, we will have more training data points available, which is always helpful for fighting the curse of dimensionality(Keogh and Mueen, 2011).

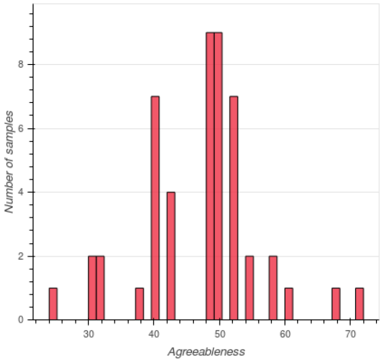


Figure 6.2: Number of authors for each value of Agreeableness to classify (Author-based approach).

Despite the considerations described, training a deep learning model with this amount of data would be quite challenging. Hence, we opted for training classical machine learning models. Provided that the personality traits vary in a continuous range, we tackled this task as a regression problem. The machine learning algorithms considered were:

- Epsilon-Support Vector Regression (SVR) model.
- Linear Regression (LR) model.
- Linear Least Squares model with l2 regularisation and $\alpha = 0.5$ (Ridge)
- Linear model trained with l1 prior as regulariser $\alpha = 0.5$ (Lasso)
- Multi-layer Perceptron classifier (MLP)
- Decision Tree Regressor (DTR)
- Random Forest Regressor (RFR)

We explore the possibility of addressing this task as a classification problem using Support Vector Machines, and Random Forest. Nevertheless, the classification approach behaved worse than the regression approach during the training and tuning phase. Therefore, the classification approach was discarded.

Regarding the features, we used only the text in the CB approach. As text representation, several vectoriser methods were evaluated for each model. The algorithms for representing text considered were:

- Term Frequency-Inverse document frequency (Tf-Idf) from one to four words (tf-words).
- Tf-Idf from one to four n-grams of words ignoring the terms that have a frequency strictly lower than the threshold set to 0.5 and applying sub-linear scaling (sublinear-1:4).
- *Idem* but exploring n-grams from one to six words (sublinear-1:6).
- Tf-Idf from one to six characters (tf-chars)
- Bag of words(BOW).

However, in the AB approach, we included handcrafted features. There was not literature that tackled personality recognition in source code. Inspired by the works that considered formal text (Argamon, Dhawle, et al., 2005), we hypothesis and propose several novel features for addressing this problem. The features considered were:

- The number of examples of code that implemented the same class (hf_1).
- The number of allocations (hf_2).
- The number of loops (hf_3).
- The appearance of pieces of code suspicious of plagiarism (hf_4)[2].
- The number of imports(hf_5).
- The number of functions (hf_6)
- The number of exceptions handled (hf_7).
- The number of classes developed (hf_8).
- The number of different classes developed (hf_9).
- The number of comment lines (hf_{10}).
- The number of prints (hf_{11}).

In addition, we carried out a preprocessing phase where code snippets (e.g. the sequence reserved words that define a loop) were replaced by tokens. Nevertheless,

[2] We assumed that those samples of code that instantiate classes that do not belong to the standard library are suspicious of plagiarism, e.g. the class *SeparateChainingHashTable*.

the systems that included this phase obtained worse results than those systems without preprocessing. This phenomenon of the negative impact of the performance of the models that preprocessed the corpora was previously reported in the author profiling literature (Argamon, Dhawle, et al., 2005; F. Rangel, Rosso, et al., 2015). Our results confirm that the preprocessing phase also has a negative impact on the personality recognition task from source code.

Finally, noteworthy that the CB approach inferred a prediction for each piece of code without considering any aggregated information from each author. Therefore, the final forecast for each personality treat of an author during test time must be computed, combining the predictions for each piece of code that an author wrote. We considered several central statistics measures. The models that used the mean achieved better results during the training.

We have developed all these algorithms and trained with all the possible combinations of features described previously using the toolkit scikit-learn (Pedregosa et al., 2011).

Since we did not have data to spare to create a development dataset, we selected the best models trained with a subset of features following a cross-validation approach. Concretely, we used a 5-cross validation approach. The loss function adopted for training the models was the official metric of the competition RMSE.

In order to select the best CB model to participate in the shared evaluation, we opted for ranking the models considering the mean RMSE for all the traits and folds as defined in equation 6.1. The selection of the models was a compromise solution among the performance for predicting any individual characteristic. This has allowed us to obtain competitive models for all traits measured with the RMSE.

$$RMSE_{mean} = \frac{\sum_{trait \in A,C,E,N,O} \frac{\sum_{fold=1}^{num\_folds} RMSE_{trait\_fold}}{num\_folds}}{num\_traits} \tag{6.1}$$

Conversely, in the AB approach, the handcrafted features were selected applying an ablation test which yielded the best combination of stylistic features. Afterwards, we combined them with the word n-grams of the most reliable model obtained for the CB approach.

A description of the results achieved by our best models is presented below. Firstly, Table 6.2 shows the RMSE of our best CB models at development time. As we previously described, we selected these models considering the official RMSE

Model	A	C	E	N	O
run_1	6.08 (±0:65)	4.82 (±0:44)	5.53 (±0:87)	8.26 (±0:94)	4.95 (±0:55)
run_2	6.10 (±0:67)	4.81 (±0:41)	5.55 (±0:89)	8.30 (±0:95)	4.93 (±0:52)
run_5	6.11 (±0:85)	4.79 (±0:47)	5.94 (±0:89)	8.54 (±1:02)	4.85 (±0:43)
run_4	6.07 (±0:81)	4.83 (±0:47)	5.89 (±0:84)	8.49 (±1:01)	4.91 (±0:44)

Table 6.2: RMSE achieved using a 5-fold validation over the train dataset following the Code Based approach. The mean RMSE and the standard deviation for the 5-fold validation for each trait is reported.

metric. Afterwards, using the parameters from the best CB approach, an AB model was trained, including the following handcrafted features: the number of samples of code that implemented the same class hf_1, the appearance of pieces of code suspicious of plagiarism hf_4, the number of classes developed hf_8, and the number of different classes developed hf_9. The models sent for evaluation in the shared campaign were:

run_1 CB approach using as text representation *sublinear-1:4* for training a Ridge model.

run_2 CB approach using as text representation *sublinear-1:6* for training a Ridge model.

run_3 AB approach using as text representation *sublinear-1:4* and the handcrafted features $hf_1 \oplus hf_4 \oplus hf_8 \oplus hf_9 \oplus hf_{10}$ for training a Ridge model.

run_4 CB approach using as text representation *sublinear-1:4* for training a Logistic Regession model.

run_5 CB approach using as text representation *sublinear-1:6* for training a Logistic Regession model.

Two baselines were provided by the organisers: a bag of words 3-grams with frequency weight (bow) and an approach that consistently predicted the mean value observed in the training data (mean). The official evaluation results for each personality trait over the test set can be found in Table 6.3a. Eleven teams have presented their respective systems. In total, 48 models were submitted for evaluation, and all of them have performed better than the mean baseline model, and all our approaches outperformed the second baseline as well. Despite our results achieved during the development phase, our best performing system was the one that followed the Author-Based approach. This system was able to

Model	A	C	E	N	O
run_1	9.29	9.02	8.75	10.67	7.85
run_2	9.36	8.99	8.79	10.46	7.67
run_3	8.79	8.69	9.0	10.22	7.57
run_4	9.62	8.86	8.69	10.73	7.81
run_5	9.71	8.89	8.65	10.65	7.79
baseline bow	9.0	8.47	9.06	10.29	7.74
baseline mean	9.04	8.54	9.06	10.26	7.57
min	8.79	8.38	8.60	9.78	6.95
max	28.63	22.36	28.80	29.44	33.53
mean	9.72	10.74	12.27	12.75	10.49

(a) RMSE achieved in the test dataset.

Model	A	C	E	N	O
run_1	0.03	-0.23	0.31	-0.22	-0.12
run_2	0.0	-0.19	0.28	-0.07	0.05
run_3	0.33	-0.12	0.18	0.09	0.03
run_4	-0.03	-0.09	0.28	-0.15	-0.05
run_5	-0.06	-0.12	0.3	-0.16	-0.02
baseline bow	0.20	0.17	0.12	0.06	-0.17
baseline mean	0.0	0.0	0.0	0.0	0.0
min	-0.32	-0.31	-0.37	-0.29	-0.36
max	0.38	0.33	0.47	0.36	0.62
mean	-0.01	-0.01	0.06	0.04	0.09

(b) Pearson Correlation achieved in the test dataset.

Table 6.3: Evaluation of our participation in the PR-SOCO shared task. The first five rows, run 1 up to run 5, show the results achieved by our systems. The traits are: agreeableness (A), conscientiousness (C), extroversion (E), neuroticism (N), and openness to experience (O). Moreover, the performance of the baseline systems are included, as well as the minimum, maximum and mean performance obtained by the participants at the shared task.

achieve the best RMSE result in the personality trait Agreeableness among all the participants.

During the evaluation phase, the organisers add a second evaluation metric: Pearson correlation between the prediction and the gold standard. The results of this second evaluation can be found in Table 6.3b. Neither our models' predictions nor other participant models succeeded to find a correlation with the gold standard following the Pearson coefficient metric. The best correlation found by the participants was 0.62 for the trait Openness, which can not be considered a strong positive correlation. Noteworthy, our systems were only optimised for the RMSE, which might have affected the performance using the Pearson Correlation since there is no direct reciprocity between the RMSE and the Pearson Correlation.

In summary, in this section, several models that tackle personality recognition from source code were presented. These models constituted our participation in the PAN@FIRE Personality Recognition in Source Code 2016 evaluation campaign. In order to be able to train with the small dataset gathered for the task, we carried out our development experiments assuming independence between each data point and followed a Code-based approach. In addition, we included an Author-Based strategy where we had included stylistic features designed manually for this task. Even though this latter approach was performing worse in the development phase, during the evaluation with the test set achieved the best results. This can be explained for multiple reasons: a) because the assumption of independence was faulty, b) because even with the cross-validation, we overfitted the models, and c) because the handcrafted features characterise stylistic artefacts linked with different personality traits. Nevertheless, the features designed to address this task require intensive study of the training dataset and could not be generalised to other personality trait classification tasks. Therefore, this approach can only be used when facing particular problem definitions.

Moreover, it is noteworthy that the minimum error achieved by the participants' proposals in the RMSE is close to the baseline models for all the personality traits, and only for some characteristics, a correlation with the gold standard was found. This highlights the complexity of the task. Therefore, personality recognition in source code remained an open problem that requires a less constrained dataset to tackle with modern NLP techniques.

6.3 Personality Recognition in Social Media

In this section, we continue to explore the personality recognition problem. Although, in order to pursue this investigation, we required a more suitable dataset with enough data points that let us explore machine learning and deep learning models. Therefore, we employed the data gathered for the PAN/CLEF evaluation campaign (F. Rangel, Rosso, et al., 2015), which allowed us first to test a classical NLP approach and secondly to use the dataset of the campaign to investigate a deep learning approach. The following two sections are devoted to each one of these bodies of work.

As we introduced in 3.1.2, one could find extensive work done in personality recognition applied to normative texts. Nonetheless, the exponential growth of social networks, as we previously discussed, has led to new challenges in the study of NLP. Therefore, this is the line of work that we explored in this section.

The corpora for the evaluation campaign was created, compiling tweets in different languages. As discussed in the introduction, the impact and extension of Twitter make the social platform a compelling research environment. Short texts found in social media also present a challenge for NLP techniques. In spite of that, the large number of data generated in social media networks allows practitioners to apply statistical NLP methods.

The objective of the organisers was to develop an author profiling evaluation campaign. Unlike user identification, author profiling does not try to identify the author's identity. Author profiling tries to determine the author's features as demographic features or personality traits. Our aim was to study personality traits. However, for completeness, the results achieved identifying the demographic characteristics proposed in this evaluation campaign are discussed in the next section. The demographic features considered were age –only in the English and Spanish dataset– and gender [3].

In order to address this task, multilingual corpora were provided by task organisers. Corpora contain 14166 tweets from 152 English authors, 9879 tweets from 100 Spanish authors, 3687 tweets from 38 Italian authors and 3350 tweets from 34 Dutch authors. Half of the tweets were written by self-identified females and the other half by self-identified male users. Regarding the distribution of the age feature, four classes were considered: a) 18-24, b) 25-34, c) 35-49, and d) 50+.

[3] In this evaluating campaign and in the corpora that the authors collected, gender was regarded as a binary feature ignoring other genders different than male or female. This is a very biased assumption that does not represent the real population. Sadly, there was nothing that we could do but raise awareness about this bias here.

However, this user characteristic was unbalanced. There were much more tweets from users whose ages range between 25-34. Nevertheless, according to Twitter's statistics, it is a safe guess to assume that age distribution is representative of the population in the platform.

6.3.1 A machine learning approach

In this section, we examine the machine learning models developed for participating in the PAN/CLEF evaluation campaign.

English	18-24	username, HTTP, m, like, know, love, want, get, RT, 3, one, people, time.
	25-34	HTTP, username, via, m, w, NowPlaying, like, others, 2, Photo, new, pic.
	35-49	HTTP, username, via, new, Data, RT, New, Big, Life, m, data, Facebook.
	50-XX	username, HTTP, RT, via, know, 2, like, m, good, day, love, 3, time, new.
Spanish	18-24	username, HTTP, si, día, quiero, ser, 3, mejor, bien, vida, hoy, voy, ver.
	25-34	username, HTTP, q, si, vía, RT, d, Gracias, ser, ver, bien, día, va, hacer.
	35-49	username, HTTP, si, q, ví, RT, México, ser, hoy, Si, d, jajaja, Gracias, 1.
	50-XX	username, HTTP, q, RT, si, i, els, l, 2, 0, 1, Mas, d, amb, és, tasa, per. 2

(a) Most common words by age group

English	Female	username, HTTP, via, m, like, love, know, RT, 3, get, want, one.
	Male	uusername, HTTP, m, via, like, RT, 2, new, w, NowPlaying, know.
Spanish	Female	username, HTTP, q, si, vía, ser, d, RT, vida, Gracias, ver, mejor, día.
	Male	HTTP, si, RT, ser, ver, q, d, hoy, día, xD, 1 va, bien.

(b) Most common words by gender

Table 6.4: Most frecuent words set in corpora.

Firstly, we studied the vocabulary of each language in the corpora. We carried out a preprocess phase where punctuation signs and stop-words[4] were removed to perform this study. Moreover, the URLs found in the tweets were substituted by the token 'HTTP', and all the usernames handles were replaced by the token 'username'. Besides, we tokenised words in order to obtain the relevant words from the vocabulary. The next step in our work was to study the vocabulary distribution aggregated by age and gender for every language that has information about these features. Table 6.4 shows the most frequent words group by gender and age, both for English and Spanish. Consistently, most frequent words were terms related to Twitter dynamics such as RT, HTTP, username, via and abbreviations. Noteworthy, even though one of the datasets is labelled as Spanish, some of the more frequent words are written in Catalan. This phenomenon of mixing languages is widespread in bilingual communities. However, it represents a challenge for solving the task.

Finally, we studied hashtags. Hashtags are relevant in Twiter because it is how users self annotate their tweets. We found out that 37.9% of English tweets, 26.7% of Spanish tweets, 59.9% of Italian tweets and, 27.3% of Dutch tweets have hashtags. It is interesting to highlight that English words are present in other corpora. In this case, English is found across all the languages due to the massive use of English in social media.

Afterwards, based on this analysis, we studied the machine learning models that could be trained to identify personality traits. We employed the Scikit-learn toolkit (Pedregosa et al., 2011) in our research and experimental settings. In order to perform a training process that allowed to use of lexicons and stylistic features, we developed a module on top of the aforementioned toolkit.

Following, we describe the features considered. We aggregate the features into three categories.

- **Textual features.** This category relies only on textual content. Still, a simple preprocessing step consisting of lowercasing the text was applied.

    1. TF-IDF coefficients.
    2. Inter-word chars with TF-IDF coefficients.
    3. Intra-word chars with TF-IDF coefficients
    4. Bag of words.

[4] stop-words is a common term in NLP that defines words that carry little lexical meaning in a sentence. Commonly, function words are considered stop-words.

For textual features 1 to 4, we considered four configurations using different n-grams sizes: 1-3, 1-4, 1-6, 3-6 and 3-9.

- **Stylistic features**.

  1. Frequency of words with repeated characters.
  2. Frequency of uppercase words.
  3. Frequency of hashtags, mentions, URL and RT.

- **Lexicon-based features**. Four lexicons were considered. A score was computed considering the polarity of every word in each lexicon and normalising it by the number of words following the equation: $\frac{1}{|W|} \sum w \in Wlexicon(w)$. Stop-words were removed before computing these features. The lexicons used were:

  1. Afinn (Hansen et al., 2011) This resource consists of a list of words with polarity values between the range -5 and +5.
  2. NRC (Saif M Mohammad, Kiritchenko, and X. Zhu, 2013) Polarity dictionary, each word in the dictionary has a real value that represents its polarity.
  3. NRC hashtags (Saif M Mohammad and Kiritchenko, 2015) List of positive and negative hashtags. These polarities have been normalised in the range ± 5.
  4. Jeffrey (Hu and B. Liu, 2004) This resource contains two different lists of words: positives and negatives. Consequently, two scores were computed using this resource – a positive and negative score.

One of the limitations of using lexicon-based features is the lack of resources in languages different from English. All the resources mentioned only exist in English. Consequently, we carry on an automatic translation of the lexicons using Google translate API [5]. Hence, English lexicons used were translated to Spanish, Italian and Dutch.

All the features mentioned were used to train several machine learning algorithms. Each author profiling characteristics to predict –this is age, gender and personality traits– was inferred independently. Therefore, seven models were developed. During the evaluation phase, we investigated the following algorithms:

[5] `https://cloud.google.com/translate/docs` Accessed on 08-09-2020

- Linear Support Vector Machine
- Polynomial Kernel Support Vector Machine
- Naive Bayes
- Descendent gradient
- Logistic Regression
- Random Forest

Similarly, as we did to predict personality traits from the source code, two approaches were evaluated. On the one hand, all the tweets from an author were joined together, forming a unique training example. This reduces the number of samples, but it has allowed us to train with longer pieces of text. On the other hand, each tweet was used independently as a training sample which has the advantage of including more examples for training. Still, each one is shorter – the maximum length would be 140 characters which were the maximum length allowed by the social platform at the time. To select the best model, we performed ten-fold cross-validation for each model trained with all the features. The chosen evaluation measure for ranking the models was precision for gender and age and the RMSE for the personality traits. These experiments allowed us to compare the performance of our models using different configurations. Finally, in order to classify gender and age, the Linear Support Vector Machine achieve the best results whilst the Linear Regression model was the best predicting personality traits.

Once we have presented the models selected, here, we discuss the results achieved by our best models both during the development phase and in the official evaluation campaign dataset.

		Accuracy	
		Development	Test
English	Gender	53.49%	63.38%
	Age	55.29%	59.80%
	Agreeable	23.7%	17.50%
	Conscientious	20.8%	18.10%
	Extroverted	20.85%	17.70%
	Open	24.78%	20.70%
	Stable	17.81%	27.80%
Spanish	Gender	56.9 %	62.5 %
	Age	46.58 %	56.82 %

	Agreeable	40.44 %	17.29%
	Conscientious	32.84 %	18.53 %
	Extroverted	36.98%	20.97%
	Open	39.55 %	16.17%
	Stable	29.05 %	24.40%
Italian	Gender	61.63%	69.4%
	Agreeable	43.28%	16.20%
	Conscientious	52.67%	12.40%
	Extroverted	45.6%	13.90%
	Open	42.20%	20.20%
	Stable	46.15%	25.30%
Dutch	Gender	57.49%	71.88%
	Agreeable	42.33 %	17.05%
	Conscientious	49.82 %	13.92%
	Extroverted	46.37%	18.29%
	Open	43.69 %	13.23%
	Stable	38%	17.85%

Table 6.5: Results obtained during development time and againts the evaluation campaign test set using the official.

Table 6.5 shows the results in terms of accuracy and RMSE obtained. Accuracy was the official metric employed for predicting age and gender, whilst RMSE was used for the personality traits. The first column of each subtable presents either the accuracy or the RMSE obtained after tunning our system during the development phase. The tunning process was carried out following a ten-fold cross-validation sweep. Meanwhile, the second column shows the results we got testing our system against the evaluation campaign test set. Noteworthy, the semantics of these two metrics it does not help the comparison of the results. Models with higher accuracies are better; meanwhile, systems with lower RMSE are better.

The organisers also calculated a global metric, aggregating all the demographic characteristics and computing the arithmetic mean of the previous ranks. Twenty-two research labs participated in the evaluation campaign. Overall we obtained 68.57% of accuracy, achieving the 13th position. Interestingly, all the participants achieved better results in Dutch, which is the dataset with fewer samples, and worse in English, the dataset with more examples. This phenomenon might indicate that the models were overfitted. In addition, in the work of F. Rangel, Rosso, et al., they highlighted than:

> Concerning personality recognition, one can see that the best results were obtained for Italian and Dutch. This fact can be explained by the smaller number of authors for these languages, which both reduces the variance in the data and raises the analysable data size per author. A similar phenomenon has been reported for personality prediction given Facebook profile pictures (Celli, Bruni, and Lepri, 2014).

Our work support this thesis. We saw an improvement in the performance of our systems when we added stylistic features after manually analysing the training dataset. However, these handcrafted features present several problems. Firstly, it encodes the biases of the practitioner, including stylistic features correlated with a trait in a particular dataset that does not imply causation. Secondly, the task is tedious and arduous to scale. Moreover, it leads to overfitted systems that had memorised certain artefacts from a particular dataset. Finally, since the handcrafted features are thoroughly tailor-made for a dataset is hard to use the model trained in the wild.

6.3.2 A deep learning approach

Pondering all the previous considerations that hinder the performance of an ML system trained with handcrafted features to predict the personality traits of the author of a text, in this section, we propose a novel deep learning model trained uniquely with the text itself. Although, in the previous section, we discussed all the demographic features that the organiser of the public evaluation campaign require the models to predict, namely age, gender and personality traits. Hereafter, we focus our work again on the PR subtask.

We introduced the use of a Convolutional Neural Network (CNN) for predicting the Big Five personality traits. The key contributions that we introduced with this proposed model were:

- Pre-trained word embeddings were used as features to develop a PR model.
- It was the first time than a deep learning model –concretely a CNN– trained with word embeddings to recognise the personality traits of an author.
- The architecture proposed achieved akin state-of-the-art results comparing it against the systems participating in the public evaluation campaign, including the one that we presented previously in this Chapter, without the limitations that models trained with handcrafted features present.

Following, we discuss the model we proposed for the Personality Recognition task. CNNs have been widely used in computer vision, as we explained in 2.3.1. In the same manner, they can be used in NLP. The intuition behind this model is that a word can be seen as a row in an image and, by extension, the whole tweet can be represented as an image. Therefore, if we organise the words in a text as an image, CNNs can be applied.

Provided that our objective was to train a model only with the text from the dataset, we used pre-trained word embeddings. Word embeddings as we discussed in 2.1.4 encode semantical meaning. Therefore, we employed this representation to have a richer input space from where CNN can learn the personality traits. Several pre-trained word embeddings can be found. We selected the embeddings introduced by Pennington, Socher, and Manning (2014) to represent the text in our models[6]. The authors have released several word embeddings; the difference among them is the origin of the data used to train the vectorise representation in an unsupervised fashion. Thus, since our data was extracted from Twitter, we used the embeddings trained from a Twitter corpus. This resource contains 1.2M words, each one represented as an n-dimensional real-valued vector. For computational simplicity, we evaluated our model using the 25-dimensional and the 50-dimensional word representations. Hence, a word (wr) is represented in our model following equation 6.2:

$$wr = \begin{cases} f(w) & \text{if } w \in \mathcal{G} \\ \vec{0} & \text{otherwise} \end{cases} \tag{6.2}$$

being $\mathcal{G}$ the Glove dictionary, the function $f(w)$ returns the Glove pre-trained vector $\in \mathbb{R}^n$ and $\vec{0} \in \mathbb{R}^n$, being $n \in [25; 50]$. We opt for including a vector of zeros for those words which are not present in the GloVe dictionary because this approximation allows us to model the relationship between seen and unseen words in the dictionary. If we would not include these vectors of zeros, the matrix that represents the tweet only will contain those words seen in the GloVe dictionary, and the order of the words would be lost. CNNs, in general, require a fixed size input matrix. Therefore, the input data must be padded. To that end, as proposed in the work of Kim (2014), we will include as many vectors of $\vec{0}$ as needed at the end of a sentence to pad the data to the maximum sequence length seen in training; during the testing phase, if a sentence is longer than this limit, it will be trimmed.

[6] These word vector representations are available at the following URL: `http://nlp.stanford.edu/projects/glove/` Accessed on 05-09-2020

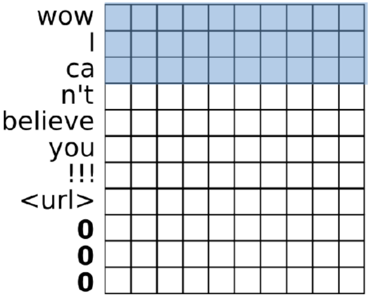


Figure 6.3: Input matrix padded to feed a CNN. Shadowed rows represent the area covered by a kernel of height 3.

In Figure 6.3, an example of text representation can be found. Each tweet will be represented using a matrix $\vec{X} \in \mathbb{R}^{m \times n}$, where m is the maximum number of words found in the training dataset, and n is the dimension of word embeddings employed. This matrix is composed of the concatenation of the word embeddings from a tweet following the column axis.

We continue the description of the model proposed explaining how we used CNNs in an NLP setup. In the literature, as we previously described in 2.3.1, several works (Kim, 2014; Y. Zhang and Wallace, 2015) advise that architectures with a convolutional layer should be followed by one or more fully connected layers in order to achieve good results in NLP tasks. The convolution is able to learn an internal representation of the text by means of which the personality traits can be learned. Therefore, we have explored architectures that contain a convolution layer and multiple fully connected layers.

Our CNN model shares some characteristics with the model proposed by Collobert et al. (2011). However, unlike the previously proposed models, our model addresses a regression problem. Consequently, the output layer will be a set of neurons with a linear activation function, and it uses the Root Mean Square Error as a cost function. The use of the RMSE as a loss function guarantees that our model is adjusted for the official metric. We respect the evaluation campaign set up to compare our model fairly against the task participants' models.

As part of the CNN architecture definition, the kernel size must be set. A kernel, as described in section 2.3.1 is a multidimensional array $k \in \mathbb{R}^{w \times n}$, applied to a window of size $w \times h$ being w the size of the GloVe embedding and h the height of the kernel. A row represents a word; therefore, the width of the kernel is determined by the word embedding size chosen. Conversely, the height of the kernel can be set to different values considering how many words we want the kernel to see. Therefore, we have experimented with the effect of different values for the height of the kernel.

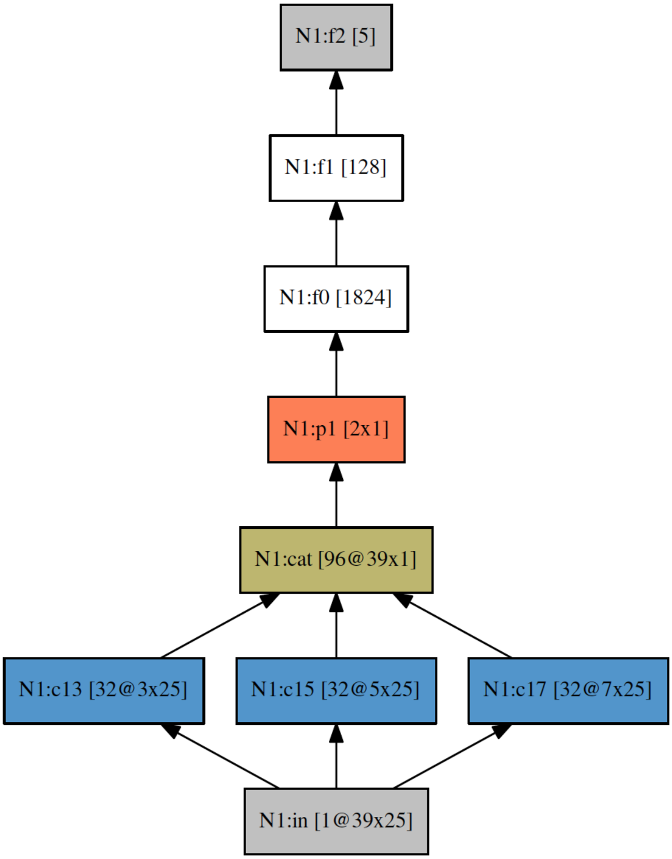


Figure 6.4: CNN architecture of our best performing system.

The convolutional layers employed Rectified Linear Units (ReLU) (Nair and G. E. Hinton, 2010) as an activation function. Multiple filters can be used to learn complementary features from the same regions. Thus, we have trained CNNs with 32 and 64 filters. Figure 6.4 shows the architecture of our best performing model. Thereafter, we will refer to this figure to describe our proposed architecture. The nodes labelled as *c13, c15, c17* represent the convolution layers using kernels of height 3, 5, and 7, respectively.

Noteworthy, we have included a concatenation layer, the node labelled as *cat* in Figure 6.4, which concatenates the feature maps obtained after applying the kernels of different heights in the convolution layers. This union allows CNN to learn contrasting representative features considering each kernel size. After that, a pooling layer may be applied over the feature maps in order to reduce the size of the feature maps and to obtain local invariance to small variations of the position of the words in a tweet. We have tried models with and without max-pooling layers. However, the best performing models always had pooling layers. In Figure 6.4, the node *p1* is a max-pooling layer. Next, this pooling layer output is fed to a fully connected layer with one or more hidden units with a ReLU activation function. In the case shown in Figure 6.4, this model has a reshape layer *f0* and one hidden layer *f1* with 128 neurons. The architectures evaluated with two hidden layers had 64 neurons in the second layer. Finally, the output layer *f2* is composed of five neurons –one for each personality trait– with a linear activation function. Another remarkable feature of this architecture is that it allows us to train all the personality traits, simultaneously speeding up the training process.

After describing the architecture, we proceed to present the evaluation and the results achieved with this model. As we previously introduced, we used the PAN-AP-2015 corpus, which was employed in the 3rd author profiling evaluation campaign. The decision of using this dataset was motivated by the availability of the data and, more importantly, because it allows comparing this novel model against state-of-the-art models under the same circumstances. Provided that we described the dataset in 6.3, here we outline its key features. This work has been focused on the English dataset. Each one of the personality traits ranges between -0.5 and 0.5. The training dataset consisted of 14,166 tweets written by 152 authors, whereas the test dataset consisted of 13,172 tweets written by 142 different authors. The evaluation of the models that participated in the evaluation campaign was carried out using the Root Mean Square Error (RMSE) for each trait and, the overall RMSE was calculated as the arithmetic means of each RMSE trait. To compare our results against the ones found in the literature, we followed the same evaluation strategy.

Previously, we have not preprocessed this dataset to get the word representations. However, in this case, since words were represented using GloVe vector representations, we tried to mimic their preprocess in order to maximise the number of words with a vector representation that can be found pre-trained in this resource.

Firstly, we tokenised the dataset using the PTBTokenizer Stanford Tokeniser[7]; the same tokeniser used to train GloVe. Then, we deleted those characters repeated more than three times in a token. This phenomenon is commonly known in the NLP literature as flooding characters, and they are used to express emphasis or emotion but rarely belong to the English vocabulary; the usual rules of English spelling outlaw triple letters. Also, URLs, user mentions, and retweets were replaced by its corresponding GloVe token. After this simple preprocess, we proceeded to create the input matrix for our Convolutional Neural Network, as we described previously.

Considering this text representation, we instantiate and train particular CNN architectures with the features that we just described. These CNN will vary in the following hyperparameters:

- The dimension of the word embeddings (25, 50).
- The number of filters (32, 64).
- The size of the filter (1, 3, 5, 7, 9, 12).
- The number of hidden layers. (128 neurons in the first hidden layer, and 64 neurons in the second hidden layer when there are two hidden layers).
- Whether or not there is a concatenation layer. When there is no concatenation layer, only one type of CNN kernel can be applied.
- Whether or not there is a pooling layer.
- Whether or not we applied batch normalization.
- Whether or not we applied dropout as means of regularization.

In the approach used for this deep learning model, each tweet from each author constitutes an independent training instance. If we try to concatenate all the tweets in a single matrix, the number of parameters to fit would make it unfeasible to train the model with the resources that we had available. Therefore, we have trained our model with 14,166 training instances, and it was tested with 13,172 cases. However, only a prediction for each author is required. Hence, the prediction for the author (a_i), and the trait (t_j) is the mean of the predictions obtained, defined following equation 6.3:

[7] `http://nlp.stanford.edu/software/tokenizer.shtml` Accessed on 6-09-2020

Model Identifier	CNN Architecture	WE	RMSE
1	K=3, N=32, H=1, BN, MP	25	**0.1650**
2	K=7, N=32, H=1, BN, MP	25	**0.1647**
3	K=357, N=32, H=1, BN, MP	25	**0.1625**
4	K=357, N=32, H=1, BN	50	**0.1633**
5	K=3, N=64, H=2, DR	50	**0.1692**

Model Identifier	E	S	A	C	O
1	0.1538	0.2214	0.1535	0.1457	0.1499
2	0.1584	0.2174	0.1525	0.1461	0.1491
3	<u>0.1582</u>	<u>0.2123</u>	<u>0.1498</u>	<u>0.1438</u>	<u>0.1482</u>
4	0.1564	0.2159	0.1517	0.1452	0.1473
5	0.1587	0.2268	0.1560	0.1475	0.1569

Table 6.6: Evaluation results of our best CNN models. RMSE: the value achieved using the Root Mean Square Error, the metric used in the competition. O: Openness, C: Conscientiousness, E: Extroversion, A: Agreeableness, S: Stability, WE: Dimension of the word embedding. MP: if there is a max-pooling layer. K: height of the kernels used. N: Number of kernels used. H: the number of hidden units. DR: if dropout was applied. BN: whereas batch normalization was applied.

$$prediction(a_i, t_j) = \frac{\sum_{s=0}^{N-1} p_{a_i,t_j,s}}{N_i} \tag{6.3}$$

Where N_i is the number of tweets for the i-th author.

Table 6.6 present the results of our best five models. The column in bold highlights the official aggregated metric of the competitions, while the underline row emphasises our best model. As can be seen, our best models were those that concatenated different convolutional filters, performed batch normalization and included max-pooling layers. Moreover, the results show that some traits accumulate a larger error consistently, e.g. the trait stability.

Noteworthy that we have trained our system only with word embeddings. In contrast, state-of-the-art systems used a combination of style-based and content-based features and n-gram models and other handcrafted features that depend heavily on the domain. Consequently, selecting these features require a comprehensive study from experts, making it harder to generalise to new data points.

Particularly, the best result in the evaluation campaign was achieved by the team *alvarezcarmona15*(Alvarez-Carmona et al., 2015). They attained 0.1442 RMSE. They have used stylistic features such as frequency of words, contractions, words with hyphens, stop-words, punctuation marks, function words, determiners, and a set of common emoticons; combined using second-order attributes to build word vectors and document vectors in the space of profiles. In addition, they used the 100 most common concepts as thematic information gathered exploiting Latent Semantic Analysis. Each document is finally represented as the union of all these features. Subsequently, they trained a LibLINEAR classier. Moreover, the authors tailor-made the classification problem for this dataset, considering only the values of the traits seen in the training dataset as discrete classes and ignoring the rest of the possible values. Despite that our approach is agnostic of the idiosyncrasies of the corpus when comparing our results against those presented by *alvarezcarmona15* using the dependent t-test, we found that the p-value obtained considering each the RMSE for each trait is 0.47. Therefore, we can retain the null hypothesis since there is no statistical significance between our results and the best performing system.

6.4 Conclusions

We can conclude that personality recognition is one of the most challenging tasks in NLP. Collecting labelled datasets with personality traits is a laborious task. Therefore, the corpora available is relatively small, and the literature focused on handcrafted approaches targeted for a concrete dataset.

Undoubtedly, when we are tackling a problem like detecting the personality on source code that we described in 6.2, the options to address this problem that we have available are limited. However, we have managed to proposed a model that solved this task satisfactorily and participated in an evaluation campaign where the results were tested independently.

Since this narrow description of the problem has restricted our possibilities to infer a person's personality traits from an NLP perspective, we participated in an evaluation campaign where the objective was to predict demographic features from tweets. This has allowed us to gain access to a well-tested dataset that we have used to validate the hypothesis that a deep learning approach can solve this problem without requiring handcrafted features, achieving results comparable with the state-of-the-art. Two approaches have been presented in this Chapter exploiting this dataset. On the one hand, we described our participation in the evaluation campaign, where we tackled this problem using classic ML algorithms and handcrafted features. On the other hand, we proposed a novel model using CNNs trained only with word embeddings. Noteworthy that the results of this simple deep learning model achieved comparable results to the best performing model on the evaluation campaign. Models that rely on handcrafted features are closely dependent on the domain and on the availability of resources. In contrast, our model is not bound by any external resource, and it is domain-independent.

In summary, PR is a challenging NLP task that could benefit from the advances in NLP techniques. Considering the impact of solving this problem in multiple domains, it is essential that we simultaneously improve the corpora available and that we applied novel NLP techniques.

Chapter 7

Study of Convolutional Neural Networks for Natural Language Processing

In this chapter, we present two contributions to the study of Convolutional Neural Networks (CNN) applied to Natural Language Processing (NLP) tasks.

On the one hand, as we introduced previously in 2.3.1, CNNs were developed within the computer vision field. In order to be able to apply CNNs effectively to NLP problems, the text of a sentence should be treated by the network similarly as it would treat an image. Embeddings, vectors of a fixed size, representing either words or n-grams[1] of characters are used for creating a matrix describing a sentence. Thus, a CNN can be trained in solving an NLP task utilising this matrix of embeddings. However, there are caveats when transferring techniques from one area of study to another. Hereafter, we studied one of the most prominent problems that one need to face when handling text as an image: the padding. Unlike images that have a fixed size or can be transformed to a fixed size, a sentence has a variable number of words. Hence, practitioners need to establish a single input size in order to be able to apply CNNs. This problem is neglected,

[1] A n-gram is a contiguous sequence of elements from a text. These elements can be characters or words from a text. See 2.1 for more information.

and the sentence is padded with a vector of zeros. Our working hypothesis was that this space that we allocate could be used more efficiently. Therefore, we considered padding the sentences with meaningful semantic vectors. We proposed a technique to fix the input size, alleviating the drawbacks of the standard zero-padding, and we proved that this methodology improves the performance of the neural network model. Moreover, this proposed methodology did not adversely affect the complexity of the training or penalise the convergence of the neural network.

On the other hand, in this chapter, we presented an investigation on how to help interpret Deep Learning models in the context of an NLP task. DL networks have allowed improving the performance of a wide variety of tasks radically. Notably, in NLP, deep learning approaches are demoting other approximations, and the improvements had led to implementing commercial applications that made the technology available for a broad public (Deng and Y. Liu, 2018). However, network interpretability is one of the main concerns that arose. Practitioner bias, ethics and the possibility of explaining the performance of these systems raised the interest of the community, which is reflected in a growth of literature on the topic (Gilpin et al., 2018; Linzen, Chrupała, and Alishahi, 2018). Therefore, we introduced a fine-grained definition of the interpretability of each filter in a Convolutional Neural Network trained for Sentiment Analysis. The purpose of this work is to discover the interpretability of each filter, namely, if a filter is learning some concepts for its internal representation. In order to do so, we studied the n-grams that got more activated in a filter and study what characteristics share these top activated n-grams in a filter. To this aim, we propose an iterative algorithm that selected the most influential n-grams for the classification of a sentence in a given class. In addition, we generalise the concept of purity presented in Jacovi, Shalom, and Goldberg, 2018, which separates n-grams into informative and uninformative for the classification.

Therefore, as can be seen, this chapter presents two distinct efforts that share the same purpose, to study how CNNs are applied in NLP tasks and the differences that this domain introduces into the DL technique.

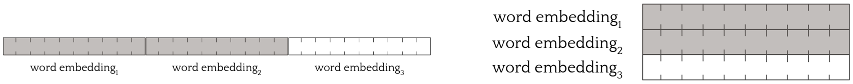
7.1 Semantic-based padding in Convolutional Neural Networks

The primary goal of the methodology proposed in this section is improving how techniques developed in Computer Vision or Speech Recognition can be adequately used in Natural Language Understanding, taking into account all the nuances that text presents.

Firstly, one must consider the different approaches for representing text in CNNs. As we introduced in section 2.1, there are several strategies for representing text. Nevertheless, in the last years, the most extensively used is text embeddings, particularly in DL models. Embeddings are a distributed representation of either words or characters in a multidimensional space as we defined in section 2.1.4. Here, we propose a methodology for word embeddings. Consequently, from now on, we will use the term embeddings as an equivalent term for word embeddings.

When applying CNNs to Computer Vision, it has been proved by W. Zhang et al. (1990) and W. Zhang (1988) that the network presents local invariance, i.e. can find relevant features in a different position of the image and is able to learn how to compose features from shallow to deeper levels. Transferring these properties to NLP tasks, a fundamental assumption is that the local invariance allows a CNN to learn which words are the more relevant to the classification task, and the compositional property allows the network to learn how to combine words embeddings to understand the meaning of a sentence. However, since the size of the filter is frequently limited between two and seven n-grams, CNNs are not suited to learn phrases that change the meaning of a sentence entirely if the distance between words is larger than the size of the kernel (W. Yin et al., 2017). Figurative language, like the language used in sentences that contain humour, irony or metaphors, present these linguistic constructions, where part of the sentence conveys a sentiment and the other part the opposite, as we showed in section 5. These long-distance relationships are hard to capture by CNNs.

Another challenging issue is that CNNs commonly required a fixed input size which is easy to obtain in images, but sentences have variable lengths. In the literature, this problem has been commonly addressed, selecting a maximum size and padding shorter sentences with zeros. Notwithstanding, other approaches have been proposed to address the fixed size restrictions in CNNs. Pal and Sudeep (2016) proposed to transform the input in order to standardise all the samples to the same size; however, this process was only found pertinent when applying CNNs to image classification tasks because a sentence cropped might not retain the original meaning. Another approach could be to develop a com-



(a) Convolution 1D.

(b) Convolution 2D.

Figure 7.1: Diagram depicting how to prepare text to train a CNN. The shadowed area corresponds to a filter that learns bi-grams. In 7.1a word embeddings are stacked horizontally. The height of a filter is fixed, and the width is a multiple of the size of the embedding vectors; hence, a convolution in one dimension is applied. In 7.1b word embeddings are stacked vertically. The width of a filter is fixed to the size of the embedding vectors, and the height vary depending on the size of the n-gram to learn; hence, a convolution in two dimensions is applied.

plex architecture of multiple sized CNNs and use the better-suited architecture in the ensemble. However, the training complexity of this approach makes it unfeasible. Finally, one could change the model completely using another approach like Recursive Neural Networks (RNNs), but the problem in CNNs is not addressed with this solution. Besides, as we described previously in 2.3.1, hybrid architectures where CNNs and RNNs are combined have achieved satisfactory results. Therefore, improving an architecture has an impact on the overall performance of these models.

Hereafter, a methodology for applying semantic-based padding in CNNS for NLP tasks is proposed. Semantic-based padding takes advantage of the unused space required for having a fixed-size input matrix in a Convolutional Network effectively, using words present in the sentence. In the next section, the methodology is proposed and has been evaluated intensively in Sentiment Analysis tasks using a variety of word embeddings. Next, the results and conclusions of the experimentation are presented. In all the experimentation carried out, the proposed semantic-based padding improved the results achieved when no padding strategy is applied.

7.1.1 Methodology

In the literature, there are two approaches to build a CNN for text classification. On the one hand, word embeddings can be concatenated and then apply a one-dimension convolution with a stride of the size of the embeddings. On the other hand, embeddings can be stacked vertically and compute a convolution where the width of the convolution has been fixed to the size of the embeddings. In Figure 7.1 a diagram of both approaches is presented.

However, in both cases, the input size of the convolution must be constant throughout the whole dataset.

The maximum size of the matrix is determined by the sentence of the maximum length in the training dataset. Sentences shorter than this value were padded, and sentences longer in the test dataset, were be trimmed. Therefore, the size of the embeddings matrix is: $\mathbb{R}^{m \times e}$ (7.1) being m the size of the longest sentence, and e the size of the embeddings used.

This restriction leads to the inclusion of padding. Padding guarantees that all the input sentences will have the exact dimensions, and the convolution can be computed efficiently. This padding is filled with a vector of a particular token, usually a vector of zeros.

Padding with a vector of zeros introduces noise in the input sentence that will influence the training of the CNN model.

To address the problems presented, we proposed to pad the sentence with embeddings of words present in the text. Therefore, when the kernel is sliding through the matrix of words towards the end of the sentence, it learns relationships between the end of the sentence and the beginning, forcing the network to learn long-distance relationships between words. In addition, this proposed semantic-based padding neither increases the size of the input of the CNN nor represents a bottleneck preprocessing the input data set because this padding process was already being carried out, and our proposed method does not constitute a computational overload.

To summarising, adding semantical-based padding did not increase the computational cost of training a CNN. The memory allocated for the matrix did not vary, including the semantical-based padding, since the new embedding vector occupied the space of the padding zero vector. In addition, the time needed for training a CNN with the enhancing semantical-based padding was not penalised. The convolutional operation is applied at the kernel level using the parallel properties of GPUs; therefore, the standard padding with a vector of zeros will not make the training faster.

Three padding methodologies were proposed.

- **Random**: This is the most basic way to fill the embeddings matrix. Each one of the empty rows is filled with embeddings from random words present in the sentence. This padding does not keep any semantic meaning, but we include it as a baseline.

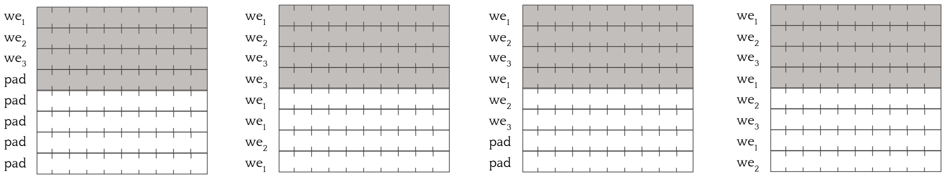


Figure 7.2: Diagrams with the different paddings compared in this work. Here we define we_i as the word embedding of the i-th word in the sentence. The first matrix corresponds to the standard *None* padding, following the *Random*, *Roll*, and *Loop* padding. The shadowed area is the area of interest of one of the four-grams learned kernels. Each kernel will slide through the sentence learning the feature maps.

- **Loop**: This is our first proposed semantic-based padding. After including each word of a sentence, the same sentence is included repeatedly, similarly as in the memory tape of a Turing machine, until the end of the matrix is filled. Hence, we are forcing the CNN architecture to learn long-distance relationships between phrases.
- **Roll**: This last padding is a variant of the previous one. In this case, we repeat each word of the sentence once in the embeddings matrix. Thus, at maximum, the sentence would appear two times. If the length of the sentence is two times shorter than m —from Equation 7.1—, the size of the embeddings matrix, zero-padding is still added at the end. We included this variant under the assumption that including zero-padding might be helpful for capturing as a feature of the model the length of the sentence while still capturing long-distance relationships between words. In systems with hand-crafted features, the length of the sentence is ubiquitous, and it is a feature that most of the models include (Saif M Mohammad, Kiritchenko, and X. Zhu, 2013; Giménez, Hernández, and Pla, 2015).

Besides, we included a fourth padding methodology in all our experiments, a classic zero-padding, that we denominated in our experiments as **None** padding.

A diagram illustrating the different paddings proposed can be found in Figure 7.2.

Considering this example sentence that could be found in a dataset *'I love this movie'*; and assuming that the maximum length of sentences seen in training is ten. Each one of the paddings used transformed the sentence as follows:

- **None:** $< I >< love >< this >< movie >< END >< END >< END >< END >$
- **Random:** $< I >< love >< this >< movie >< I >< movie >< love >< love >< movie >< this >$
- **Loop:** $< I >< love >< this >< movie >< I >< love >< this >< movie >< I >< love >$
- **Roll:** $< I >< love >< this >< movie >< I >< love >< this >< movie >< END >< END >$

Brackets separate tokens. The token $< END >$ is used for padding the end of a sentence, and it will be replaced by a vector of zeros.

All the experiments were executed with the four types of paddings proposed. Our objective is to prove that semantic-based paddings can improve the performance of the CNN architecture, regardless of the variations that training deep learning methods may present. Accordingly, we have trained and evaluated CNN models with the padding used in the state-of-the-art systems and the paddings we proposed. This allowed us to evaluate fairly the improvements of the methodology.

No other preprocessing of the sentences were carried out. Two reasons motivated this decision. The first one is that deep learning networks would learn an internal representation that best suits the task at hand, and the second one it is because we wanted to isolate the effect of the padding from other conditioning factors.

7.1.2 Experimental Setup and Results

In this section, the experiments conducted for validating the performance of semantic-based padding are described. These experiments proved that the performance of the systems where semantic-padding has been applied was improved. We have selected a classic NLP task: sentence-level sentiment analysis on various well-known datasets.

Following the description of the experimental setup is described. Also, we will discuss the results achieved.

In order to test the performance of the proposed paddings, we decided to employ a state-of-the-art architecture. The architecture proposed by Kim (2014) has been used. In Figure 7.3, a diagram of this architecture is depicted. This CNN architecture is a manageable deep learning model which allows us to evaluate the performance of our proposed semantic-based padding with different embeddings

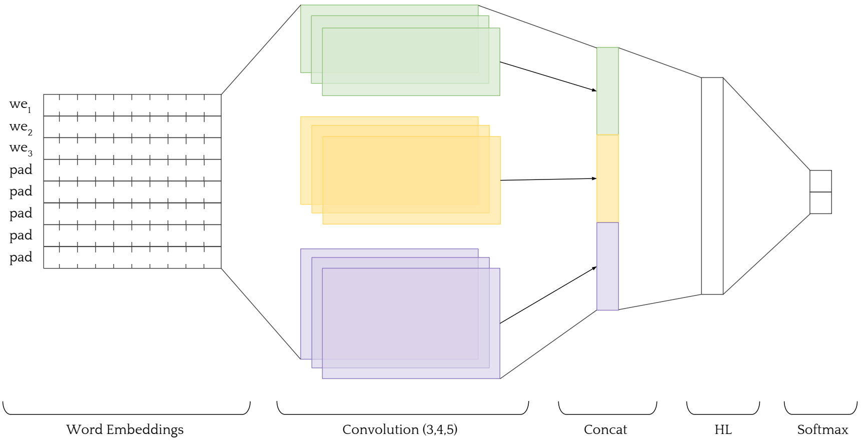


Figure 7.3: Model architecture without any kind of padding (None padding).

and datasets extensively. Besides, this architecture has been validated in the literature (Barnes, Klinger, and Walde (2017), Medhat, Ahmed Hassan, and Korashy (2014), and W. Yin et al. (2017)).

We have reimplemented the single-channel architecture proposed using the toolkit Tensorflow developed by Martín Abadi et al. (2015) and selected the same hyperparameters described in the literature. This is, we used three types of CNN kernels size $\mathbb{R}^{3\times e}$, $\mathbb{R}^{4\times e}$, and $\mathbb{R}^{5\times e}$ respectively, being e the size of the embeddings studied; each type of convolution will learn 100 feature maps; we used rectified linear (RELU) units after applying the convolution operation; the dropout rate was 0.5, and the fully connected layer consisted on 150 neurons. We have trained the model using mini-batches of size 100 through Adam gradient descent. We opted for reimplementing the system because this allows validating the performance of the padding without the variances that might appear due to the particularities of the toolkit or even the hardware used to train the reported state-of-the-art systems.

Sentiment Analysis Datasets

Sentiment Analysis, as we discussed previously in 3.1.1, is one of the most studied tasks in NLP. This is the reason that led us to select this task for this work. It allowed us to compare our proposed method to a well-known task and datasets.

The datasets used are:

- **SST-5**: Stanford Sentiment Treebank (Socher et al. (2013)), also found in the literature as *SST-fine*, is a dataset consisting of movie reviews that were annotated for five levels of sentiment: strong negative, negative, neutral, positive and strong positive. This dataset is annotated both phrase-level and sentence-level. Only the label for the whole sentence was considered as the golden truth in this work.
- **SST-2**: Stanford Sentiment Treebank binary (Socher et al. (2013)). The sentences of this dataset are the same sentences that composed the previous dataset. However, in this case, only two classes are considered: positive and negative. Sentences with strong positive and positive labels were merged in the positive class. Similarly, sentences with strong negative and negative labels were joined as negative sentences. Finally, neutral sentences were discarded.

	Train	Dev.	Test	Vocabulary	$\overline{Length}$	Min	Max	$\overline{Pad}$
SST-2	6920	872	1821	17539	19.3	2	52	32.45
SST-5	8544	1101	2210	19500	19.14	2	52	32.85

Table 7.1: Statistics of the number of words and padding needed in the datasets.

Table 7.1 reflects the statistics of the datasets studied. Noteworthy, the input matrix has size $\mathbb{R}^{52\times e}$ being e the size of the embeddings studied. Thus, on average, 32 positions of the input matrix are filled with padding ($\overline{Pad}$). Therefore, it is reasonable to argue that how we pad the input sentence will have an impact on the training of our deep learning model.

In addition, the number of sentences that require a considerable amount of padding corresponds to the head of a truncated Gaussian in both datasets, as it can be visualised in Figures 7.4 and 7.5. However, the tail of these Gaussians, i.e. the longer sentences, determine the size of the input matrix. Besides, this problem will increase in tasks where the average length of the sentences is smaller than in

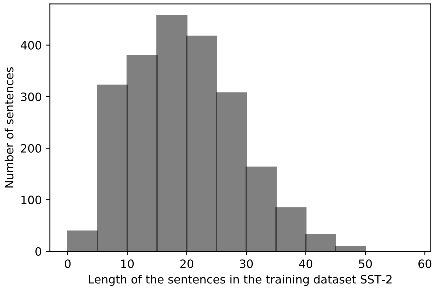


Figure 7.4: Distribution of the sentences length in the dataset SST-2.

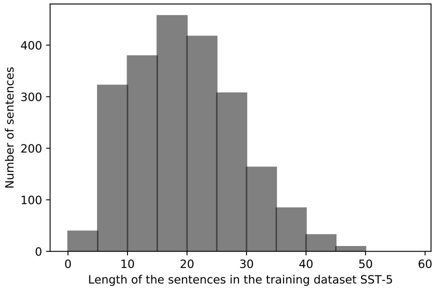


Figure 7.5: Distribution of the length of the sentences in the dataset SST-5.

these two datasets studied and in datasets where sentences present more variability in their lengths.

Word Embeddings Studied

In this section, we describe the word embeddings studied. We decided to use different word embeddings to validate the ability of the semantic-based padding to improve the performance of a model regardless of the methodology used for training these embeddings.

- **Random**: In this case, we assigned a vector of size $\mathbb{R}^{100}$ to each word. These vectors were initialised with random noise. Values were drawn from a uniform distribution. These word embeddings are the only ones used that were not pre-trained. Hence, they have no previous knowledge. Consequently, we needed to train word embeddings while we were learning the classification task.

- **NNLM-50**: These word embeddings were trained following the Neural Network Language model proposed by Bengio, Ducharme, et al. (2003). It maps each word into a 50-dimensional embedding vector. The Neural Network that learned these embeddings was trained on English Google News 200B corpus. Besides, it has a pre-built out-of-vocabulary (OOV) method that maps words that were not seen in the vocabulary of the training dataset with hash buckets. Each hash bucket is initialised using the remaining embedding vectors that hash to the same bucket.

- **NNLM-50-norm**: Similar to the previous one, but in this case, word embeddings were normalised.

- **NNLM-128**: These words embeddings were trained in the same fashion as *NNLM-50.* However, each word is mapped into a 128-dimensional embedding vector.

- **NNLM-128-norm**: Analogously, these words embeddings were a normalized version of *NNLM-128.* Each one of the NNLM models was proposed by Bengio, Ducharme, et al. (2003)

- **W2V-250** Word embeddings based on skipgram version of word2vec proposed by Mikolov, K. Chen, et al. (2013). They were trained using the English Wikipedia corpus. The algorithm used for training was hierarchical softmax, and sub-sampling was set to 1e-5. All the words OOV were mapped into one bucket, which was initialised with zeros.

- **W2V-250-norm**: These embeddings were a normalized version of *W2V-250.* These last two embeddings were proposed by Mikolov, K. Chen, et al. (2013).

All word embeddings modules studied, except for the *Random* word embeddings, were downloaded from the online library Tensorflow Hub [2].

Moreover, since word embeddings can be trained for improving the task at hand, we have included experiments where the word embedding input will be updated while training the sentiment analysis classifier. Experiments carried out with the embeddings being updated during training are labelled in the results presented hereunder as *training.*

[2] For more information, please visit the following URL `https://www.tensorflow.org/hub/`

Results

In this section, the results of our experimentation are exposed. As we described previously, we have implemented the model presented in 7.1.2 to validate the impact of the semantic-based padding in Sentiment Analysis tasks.

In this section, tables with the results found are presented. All the Tables in the section follow the same structure. The first two columns present the mean and the standard deviation of the ten experiments that we carried out. The third column reports the absolute difference between the semantic-based padding studied and the zero-based padding, i.e. *None padding*. Finally, the statistical significance of the results is reported in the last two columns. The null hypothesis indicates that the semantic-based padding is not contributing to the improvement of the performance. To determine the p-value or significance, we have calculated the T-test (De Winter, 2013) between *None-padding* and the semantic-based padding, i.e. *Roll-padding* and *Loop-padding*, to demonstrate that it exists a significant difference between the state-of-the-art padding and the semantic-based padding. Also, we have computed the T-test between the two proposed semantic-based paddings to find out if there are differences among these two methodologies for padding sentences.

The results of the experimentation with dataset SST-2 can be found in Tables 7.2, 7.4, 7.6, and 7.8. Similarly, the results of the experimentation with dataset SST-5 can be found in Tables 7.3, 7.5, 7.7, and 7.9. For each dataset we reported the results using embedding with and without normalisation. Also, we experimented training the word embedding simultaneously and with its values frozen.

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-v. None	p-v. Roll
Embeddings	Random	None	74.83 (0.74)	-	-	-
		Random	72.96 (1.09)	-	-	-
		Roll	75.18 (1.1)	0.34	$P < .001$	-
		Loop	77.31 (0.16)	2.47	$P < .001$	$P < .001$

Table 7.2: Results obtained experimenting with Random Word Embeddings on the SST-2 dataset. The $\overline{Accuracy}$ column reports the mean and the standard deviation of the experiments carried out. $|\Delta|$ column reports the absolute difference between the semantic-padding and the zero-padding. The column p-value None shows the statistical significance between the semantic-based padding and the zero-based padding. Finally, p-value Roll column shows the statistical significance between the two semantic-based padding: Roll and Loop

In several experiments, we have improved the results presented in the state-of-the-art papers (Kim, 2014; Barnes, Klinger, and Walde, 2017). The results of these experiments were highlighted in bold. Noteworthy, if the model with *None*

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-v. None	p-v. Roll
Embeddings	Random	None	33.91 (1.50)	-	-	-
		Random	33.51 (0.96)	-	-	-
		Roll	34.42 (1.54)	0.50	$P < .001$	-
		Loop	35.03 (1.51)	1.52	$P < .001$	$P < .001$

Table 7.3: Results obtained experimenting with Random Word Embeddings on the SST-5 dataset.

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-value None	p-value Roll
Embeddings	NNLM-50	None	75.17 (1.42)	-	-	-
		Random	76.00 (1.06)	-	-	-
		Roll	76.44 (0.91)	1.26	.015	-
		Loop	77.50 (0.71)	1.50	$P < .001$	$P < .001$
	NNLM-50 training	None	76.30 (1.06)	-	-	-
		Random	76.58 (2.31)	-	-	-
		Roll	77.20 (0.85)	0.89	$P < .001$	-
		Loop	76.82 (0.69)	0.51	.03	$P < .001$
	NNLM-50 norm.	None	76.49 (0.73)	-	-	-
		Random	75.81 (0.90)	-	-	-
		Roll	77.47 (0.54)	0.97	$P < .001$	-
		Loop	77.92 (1.11)	1.42	$P < .001$	.17
	NNLM-50 norm. training	None	76.48 (0.64)	-	-	-
		Random	75.37 (0.92)	-	-	-
		Roll	76.86 (0.91)	0.38	$P < .001$	-
		Loop	77.70 (0.61)	1.21	$P < .001$	$P < .001$

Table 7.4: Results obtained experimenting with NNLM-50 Word Embeddings on the SST-2 dataset. NNLM-50 used the pretrained weights; NNLM-50 training included a mechanism for training the embeddings simultaneously with the SA task; analogously NNLM-50 norm. and NNLM-50 norm. training are the normalised version of NNLM-50 and NNLM-50 training. The padding column describes the padding strategy followed.

		Padding	$\overline{Accuracy}$	$\|\Delta\|$	p-value None	p-value Roll
Embeddings	NNLM-50	None	35.41 (1.24)	-	-	-
		Random	35.71 (1.28)	-	-	-
		Roll	36.01 (0.64)	0.59	$P < .005$	-
		Loop	36.20 (0.54)	0.78	$P < .001$	.64
	NNLM-50 training	None	35.30 (0.83)	-	-	-
		Random	35.05 (0.94)	-	-	-
		Roll	37.40 (0.70)	2.10	$P < .001$	-
		Loop	37.08 (1.45)	1.78	$P < .001$	.62
	NNLM-50 norm.	None	34.19 (0.97)	-	-	-
		Random	34.73 (1.41)	-	-	-
		Roll	36.32 (1.64)	2.13	$P < .001$	-
		Loop	36.35 (1.22)	2.16	$P < .001$	.54
	NNLM-50 norm. training	None	35.15 (0.87)	-	-	-
		Random	35.30 (1.69)	-	-	-
		Roll	36.79 (0.72)	0.38	$P < .001$	-
		Loop	37.58 (0.65)	1.21	$P < .001$	$P < .001$

Table 7.5: Results obtained experimenting with NNLM-50 Word Embeddings on the SST-5 dataset. All the columns were described previously.

		Padding	$\overline{Accuracy}$	$\|\Delta\|$	p-value None	p-value Roll
Embeddings	NNLM-128	None	81.34 (1.3)	-	-	-
		Random	81.31 (0.41)	-	-	-
		Roll	**82.45** (0.45)	1.09	$P < .001$	-
		Loop	**82.36** (0.67)	1.02	$P < .001$	.53
	NNLM-128 training	None	75.04 (3.23)	-	-	-
		Random	73.75 (0.81)	-	-	-
		Roll	75.76 (1.17)	0.72	.067	-
		Loop	75.57 (1.50)	0.53	.14	.65
	NNLM-128 norm. training	None	**81.63** (0.56)	-	-	-
		Random	80.08 (1.06)	-	-	-
		Roll	**82.25** (0.75)	0.62	$P < .001$	-
		Loop	**82.00** (0.75)	0.37	$P < .001$	.43
	NNLM-128 norm. training	None	75.09 (1.81	-	-	-
		Random	74.58 (0.75)	-	-	-
		Roll	77.55 (0.53)	2.46	$P < .001$	-
		Loop	76.57 (0.25)	1.48	$P < .005$	$P < .001$

Table 7.6: Results obtained experimenting with NNLM-128 Word Embeddings on the SST-2 dataset. All the columns were described in Table 7.4.

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-value None	p-value Roll
Embeddings	NNLM-128	None	38.59 (1.82)	-	-	-
		Random	38.60 (0.41)	-	-	-
		Roll	**40.93** (1.45)	2.3	$P < .001$	-
		Loop	40.07 (0.65)	1.47	$P < .001$	.41
	NNLM-128 training	None	34.95 (1.20	-	-	-
		Random	33.72 (1.91)	-	-	-
		Roll	39.47 (1.12)	4.52	$P < .001$	-
		Loop	39.57 (0.57)	4.62	$P < .001$	.84
	NNLM-128 norm.	None	38.35 (0.95)	-	-	-
		Random	38.32 (0.70)	-	-	-
		Roll	39.56 (1.43)	1.21	$P < .001$	-
		Loop	40.07 (0.98)	1.72	$P < .001$	.76
	NNLM-128 norm. training	None	34.12 (2.53)	-	-	-
		Random	33.35 (0.98)	-	-	-
		Roll	35.46 (0.78)	1.34	$P < .005$	-
		Loop	36.62 (0.98)	2.50	$P < .001$	.78

Table 7.7: Results obtained experimenting with NNLM-128 Word Embeddings on the SST-5 dataset. All the columns were described previously.

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-value None	p-value Roll
Embeddings	W2V-250	None	80.38 (3.24)	-	-	-
		Random	80.49 (0.21)	-	-	-
		Roll	**81.88** (0.81)	1.5	$P < .001$	-
		Loop	**82.14** (0.94)	1.76	$P < .001$	.50
	W2V-250 training	None	73.67(5.27)	-	-	-
		Random	73.99 (1.21)	-	-	-
		Roll	76.27 (0.98)	2.60	$P < .001$	-
		Loop	75.83 (1.57)	2.16	$P < .001$	.08
	W2V-250 norm.	None	77.18 (0.68)	-	-	-
		Random	75.11 (2.12)	-	-	-
		Roll	77.97 (0.74)	0.79	.03	-
		Loop	78.08 (0.61)	0.90	$P < .005$	.32
	W2V-250 norm. training	None	75.37 (1.66)	-	-	-
		Random	75.03 (0.45)	-	-	-
		Roll	76.26 (0.67)	0.89	$P < .005$	-
		Loop	77.25 (1.58)	1.88	$P < .001$	$P < .005$

Table 7.8: Results obtained experimenting with W2V-250 Word Embeddings on the SST-2 dataset. All the columns were described previously.

		Padding	$\overline{Accuracy}$	$\lvert\Delta\rvert$	p-value None	p-value Roll
Embeddings	W2V-250	None	33.75 (0.90)	-	-	-
		Random	32.20 (0.79)	-	-	-
		Roll	35.71 (1.28)	1.96	$P < .001$	-
		Loop	35.26 (1.04)	1.50	0	
	W2V-250 training	None	32.67 (0.19)	-	-	-
		Random	30.28 (1.80)	-	-	-
		Roll	34.34 (0.87)	1.67	$P < .001$	-
		Loop	33.18 (1.99)	0.51	.058	$P < .001$
	W2V-250 norm.	None	34.85 (1.32)	-	-	-
		Random	33.90 (1.21)	-	-	-
		Roll	35.44 (0.33)	0.59	$P < .005$	-
		Loop	35.55 (1.24)	0.70	$P < .005$	.72
	W2V-250 norm. training	None	33.03 (2.37)	-	-	-
		Random	32.40 (2.83)	-	-	-
		Roll	35.26 (1.29)	2.23	$P < .001$	-
		Loop	34.11 (2.18)	1.08	$P < .005$	.008

Table 7.9: Results obtained experimenting with W2V-250 Word Embeddings on the SST-5 dataset. All the columns were described previously.

padding, the one used in the literature, achieved the state-of-the-art performance, our proposed semantic-based padding improved the accuracy. Moreover, in all the cases, the semantic-based padding outperforms the zero-padding approach.

Discussion and Conclusions

In this section, the results presented are being evaluated and discussed. Firstly, the most obvious conclusion we can draw from the experiments assessed is that semantic-based padding improved the performance of the SA task. Moreover, it did it in a statistically significant way, as the p-value calculated between the results achieved without padding and the ones obtained with both semantic-based paddings were below $< .05$ in almost every case. However, when we were training the word embeddings simultaneously with the SA task, the performance of the semantic-based padding did not have an impact as relevant as when embeddings were not trained. Updating the weights of the embedding input during training helps to increase the performance of the task. Therefore, the margin for improvement with the semantic-padding might be narrower.

Secondly, *Random padding* always performed worst than any other padding. This is because the model learned relationships between words that have no relationship between them in the sentence, despite the individual impact of some words in determining the sentiment of a sentence. This experiment also proved that CNNs were learning how words interact with each other, and when random connections appeared, the model performed worst.

In the results presented, both semantic-based paddings boosted the performance achieved by the zero padding. The improvement found between both systems was very similar, which is validated by the p-value obtained. In most of the experiments, the difference between the two proposed semantic-based paddings was not statistically significant. On the one hand, the end of sentence tokens present on the *Roll* semantic-based padding encoded the length of the sentence, which is a typical feature in models trained with hand-crafted features. On the other hand, the *Loop* semantic-based padding allowed CNN to learn the same pattern of words in more positions of the embedding input matrix, introducing more redundancy. Although both proposed semantic-based paddings achieved a very similar performance, outperforming the state-of-the-art that used zero padding, the features learned by the model differ. This work presents multiple follow-up lines of research that will be discussed in Chapter 8.

To summarise this work, we have proposed a random padding and two semantic-based paddings for NLP tasks and validated its performance with two Sentiment Analysis datasets and seven different word embeddings. For each experiment performed, the semantic-based padding proved to improve the performance of the system. Moreover, when the model without padding achieved state-of-the-art performance, the semantic-based padding outdoes the accuracy reported in state of the art.

7.2 Fine-grained interpretability of Convolutional Neural Networks

As we introduced previously, the objective of this section is to present a fine-grained interpretability framework to explain the decisions taken by a Convolutional Neural Network when classifying a sentence.

The general hypothesis is that a filter learns to detect specialised concepts, and the neural network will use this internal representation for classifying the input in between different categories. This hypothesis has been proven for image classification in several works (Cruz-Roa et al., 2013; Q.-s. Zhang and S.-C. Zhu, 2018). Notably, the work of Bau et al. (2017) proves that the first layers of a convolutional neural network detect abstract concepts like edge detectors, and deeper layers get activated with more complex concepts like objects.

Conversely, instead of establishing a fixed threshold to interpret the saliency maps learned by the models, we proposed an iterative algorithm for selecting g the interpretability threshold of filters iteratively in a CNN. Moreover, a novel fine-grained interpretability method to explain the most relevant n-grams learned by a CNN and a methodology for retrieving the ratio of relevant n-grams required for classifying a sentence correctly is introduced.

Our hypothesis and algorithms were tested in a Sentiment Analysis task. Finally, the results and conclusions learned are presented.

7.2.1 Methodology

In this section, we describe the methodology we propose for shedding light on what a network is learning. Provided that the internal representation learned by a CNN maximises the objective function for which it has been trained, full interpretability might not be attainable. However, the hypothesis is that certain linguistics concepts are encoded in this representation. Besides, determining which kernels add relevant information is useful for concluding if the design of the neural network is oversized for the problem, more extensive networks can overfit to the training samples memorising them, and they require resource-intensive training.

The key objectives of the methodology proposed for interpretability are:

- The interpretability methodology should not impact the performance of the model. A filter is discarded when it does not add relevant information. Hence, the performance of the classification task must be preserved even when we are studying the interpretability of the model.
- Do not modify the architecture of the model. Including new layers such as an upsampling layer, B. Zhou et al., 2016 or transforming the architecture into a Network in NetworkLin, Q. Chen, and Yan, 2013 will lead to learning how to interpret similar architectures. The objective that we seek is to be able to interpret a production model without modifications.
- Following the same philosophy described in the previous point, this methodology will not modify the learned weights of the model. We will not try to saturate a neuron to discover what is learning because this might introduce

a practitioner bias. Therefore, we will only rely on the forward pass to interpret our network.

The methodology proposed has two granularity levels. The first one, described in section 7.2.1, focuses on studying what the filters of the convolutional layers are learning. Meanwhile, the second one, defined in section 7.2.1, seeks to discover what the whole neural network is learning, combining the most relevant features learned in the filters of the CNN using fully connected layers.

Discovering the interpretability of Convolutional Network Kernels

To interpret what the filters of the convolutional neural network are learning, firstly, we propose an iterative algorithm that will select the more relevant neurons without penalising the performance of the system. Then, we propose several metrics to determine the interpretability of what the CNN filters learned.

In the work of Jacovi, Shalom, and Goldberg (2018), the authors studied the contribution of each filter f_l to a class, assuming the case of non-linear fully connected layers. This assumption is a simplification of the state-of-the-art Convolutional Neural Network architecture but allowed explaining the n-grams learned simply. Contrastly, our objective is to interpret each filter following a fine-grained approach and taking into account the effect of the non-linear functions used after the convolution operations to increase the expressibility of the functions learned.

Furthermore, the iterative algorithm proposed defines a formal methodology for selecting a threshold for a model, eliminating the uncertainty of selecting a threshold empirically for each task and model. We use the inner tensors of the convolutional layers for studying the interpretability of what trained CNN learned and evaluated the results using the saliency maps obtained. This allowed us to study how each neuron in a filter contributes differently to predict the class of a sentence.

A Convolutional Neural Network is composed by convolutions with different kernel sizes. In order to take into consideration the different behaviours of each filter with the same kernel configuration, we computed the top percentile for each layer $l \in L$. This process relies on selecting the top quantile level as Bau et al., 2017 proposed. In this paper, the authors suggested to filter the more relevant units considering the saliency maps $A_k(x)$ obtained for each image in the dataset. Afterwards, for each unit k the top quantile level T_k is defined as $P(a_k > T_k) = 0.005$ and the units below this threshold are set to $\vec{O}$.

Conversely, we propose to use the weights and biases directly from each convolutional neural layer instead of the activation maps. In this way, we simplified the computation since the top percentile is computed only once for each layer and computing the activation maps for selecting the top percentile is not required. Most importantly, this method computed a threshold independently of the dataset considered for the evaluation. Moreover, we propose to refine this threshold selection making it iterative. Our algorithm iteratively selects the thresholds considering the performance of the model. It starts with a small conservative threshold and increases it each iteration as described in the algorithm 1. It is important to stress that this process will select the most relevant units at the convolutional layer level. Afterwards, the max-pooling layer will filter these relevant units, keeping only the units with maximum value. Studying the output of the convolutional layer allows us to gather information about what the model is learning.

```
Data:
  • A develoment dataset $d = (x_j, y_j)| \forall j x_j \in \vec{X} d, y_j \in \vec{Y} d$
  • A CNN model trained for the task at hand.
Result: The threshold to discard n-grams in every layer of the CNN model
        without compromising the performance of the model.
top_percentile = 0.99;
$F'_{1dev}, F_{1,dev} = F_{1,dev}, F_{1,dev}$;
while $F'_{1,dev} \geq F_{1,dev} + \epsilon$ do
    forall $l \in L$: layer in the CNN model do
        threshold_weights = get_top_percentile_layer($\vec{W_l}$, top_percentile);
        threshold_bias = get_top_percentile_layer($\vec{b_l}$, top_percentile);
    end
    $F'_{1,dev}$ = Compute $F_{1,dev}$ with the masked convolution setting to $\vec{0}$
      the positions where the weights of the convolution are below the threshold;
      similarly set to $\vec{0}$ the positions where the biases are below the threshold;
    if $F'_{1,dev} \geq F_{1,dev} + \epsilon$ then
        top_percentile -= 0.01;
    end
end
```

Algorithm 1: Iterative algorithm for finding the interpretability threshold.

The algorithm has been designed and implemented to select the threshold after training the model. Nevertheless, it could be implemented during training using the development split. In that case, we would evaluate the interpretability of the model, and the training schedule could be adjusted accordingly. However, for simplicity, we implemented the search only once after the training is completed.

At this point, we can define the proposed method for evaluating the interpretability of each one of the feature maps learned at a level of the CNN.

Given a labelled dataset $\mathbb{D}$ with M tuples (x_i, y_i), where x_i is a text in a known language and y_i the class from the ground truth assigned to this text, we try to validate or discard the hypothesis that a CNN learns semantic concepts. To that end, we automatically annotated each word with semantic concepts such as part of speech tags, obtaining a vector $An(\vec{x_i})$ with the annotation labels[3] found in a sentence. Moreover, $R_k(\vec{x_i})$ is the set of relevant n-grams from sentence i that was labeled as class $\mathrm{y_i}$ in the convolutional unit k. The relevance vector is obtained by selecting the n-grams with the value over the threshold computed applying the iterative algorithm described in 1. Therefore, Equation 7.2 define the interpretability of each filter in each layer of a convolutional network considering only those examples where the CNN predicted the class correctly. This Equation is analogous to the purity function proposed in Jacovi, Shalom, and Goldberg, 2018, but here we consider the annotation concepts explicitly and the non-linear function applied after the convolutional layer.

$$interpretability_{k,c,t} = \frac{\sum_{i=0}^{M} |\{R_k(\vec{s_i})|y_i == c\} \bigcap An_t(\vec{s_i})|}{\sum_{i=0}^{M} |\{R_k(\vec{s_i})|y_i == c\} \bigcup An_t(\vec{s_i})|} \tag{7.2}$$

Meanwhile, Equation 7.3 considers what the filter learns in each sentence independently of the correctness of the prediction. This interpretability measure gives insights into global concepts learned in a feature map from a convolutional layer, regardless of how the following layers combine this information to predict the label of the sentence.

$$interpretability_{k,t} = \frac{\sum_{i=0}^{M} |\{R_k(\vec{s_i})\} \bigcap An_t(\vec{s_i})|}{\sum_{i=0}^{M} |\{R_k(\vec{s_i})\} \bigcup An_t(\vec{s_i})|} \tag{7.3}$$

Similarly, Equation 7.4 measures the ability of each convolutional unit to predict the class of the sentence correctly. In this case, we aim to uncover units specialised in identifying a class disregarding any semantic concepts.

$$interpretability_{k,c} = \frac{\sum_{i=0}^{M} |R_k(\vec{s_i})| \quad y_i == c|}{\sum_{i=0}^{M} |R_k(\vec{s_i})|} \tag{7.4}$$

[3] The annotation classes considered are discussed in Section 7.2.2

From the previous definitions, we can derive general metrics such as accuracy, precision, recall and the F_1 score for each unit in a layer.

$$TP_k = \sum_{i=0}^{M} |R_k(\vec{s_i})| \ y_i == c| \qquad TN_k = \sum_{i=0}^{M} |NR_k(\vec{s_i})| \ y_i! = c|$$
$$FP_k = \sum_{i=0}^{M} |R_k(\vec{s_i})| \ y_i! = c| \qquad FN_k = \sum_{i=0}^{M} |NR_k(\vec{s_i})| \ y_i == c|$$

Being $NR_k(\vec{x_i})$ the set of not relevant n-grams, this is the inverse of $R_k(\vec{x_i})$ previously presented.

$$accuracy_k = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.6a}$$
$$precision_k = \frac{TP}{TP + FP} \tag{7.6b}$$
$$recall_k = \frac{TP}{TP + FN} \tag{7.6c}$$
$$F_{1,k} = 2\frac{precision \cdot recall}{\text{precision} + \text{recall}} \tag{7.6d}$$

Figure 7.6 shows an schema of the framework proposed. The top of the Figure depicts a CNN model with two convolutional layers with four filters, each layer and a fully connected layer. The shadowed areas represent the neurons above the interpretability threshold. Considering the receptive field of these neurons, the lower part of the Figure illustrates the words where the model was paying attention to classify a sentence with a polarity label. If a filter does not have any neuron over the threshold, it means that this filter was not relevant for the classification. In the Figure 7.6, the last filter from the $layer_0$ and the second one from $layer_1$ exemplifies this case.

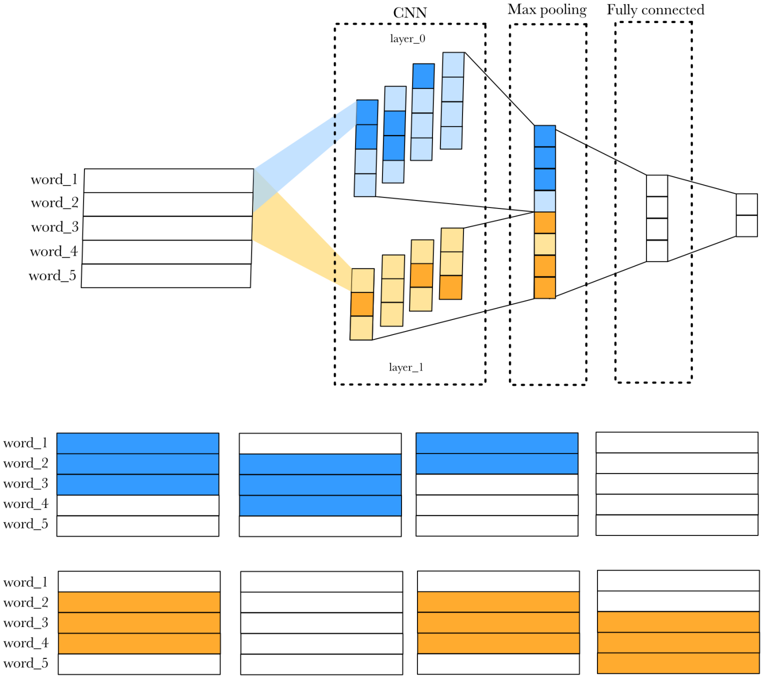


Figure 7.6: Schema of the interpretability framework proposed.

Network Interpretability

Previously, we described a methodology for interpreting convolutional layers. Though, to predict the class of a sentence, the neural network feeds the internal representation learned in the convolutional layers to a fully connected layer. Hereafter, we extend the methodology presented for interpreting what the final layers are learning.

After the convolutional layer, a max-pooling layer selects the neuron with higher activation for each filter, creating a vector $\vec{M}p(s_i)$ and a set of fully connected layers will predict the label y_i of the i-th sentence. In the simplest architecture, one will have a layer that will compute unnormalised log probabilities of the classes:

$$\vec{z} = \vec{W}^\top \vec{M}p + \vec{b} \tag{7.7}$$

where $\vec{w}$ a vector of learned weights that will classify the sentence, and $z_i = \log \tilde{P}(y = i|\vec{M}p)$.

The algorithm described in 1 computes the threshold value for the relevant weights and biases in the kernel in the convolutional layer. Hence, we propose to evaluate during inference how many of the relevant n-grams needed to perform a decision as described in Equation 7.8.

$$relevance\_fc = \frac{\sum_{i=0}^{M} \sum_{k=0}^{K} |\vec{W}^\top \vec{Mp}_k(\vec{s_i})|}{\sum_{i=0}^{M} \sum_{k=0}^{K} |\vec{W}^\top \vec{Mp}(\vec{s_i})|} \tag{7.8}$$

Being Mp_k the max-pooling achieved after filtering the least relevant positions from the thresholded convolutional layers, and Mp_k the max-pooling learned without thresholding.

7.2.2 Experimental Setup and Results

Hereafter, we presented the evaluation of the interpretability framework proposed. Firstly, we describe the task and the details of the model that we want to interpret. Subsequently, we present the results of the experimental phase.

Likewise, as in the semantic-based padding study, Sentiment Analysis was the NLP task selected to evaluate this framework. As we introduced previously, SA is a well-studied task that allowed us to focus on the analysis of the interpretability of the model. Provided that in the literature, there is evidence that suggests that longer texts are challenging for CNN models(W. Yin et al., 2017), we have selected a task that aims to predict the sentiment of a movie review. Sentences with different sizes were studied, allowing us to analyse the effect of this parameter in the learned inner representation.

The dataset used is commonly known as IMDB because it is composed of movie reviews extracted from the popular website[4]. Maas et al. (2011) curated this dataset, establishing a popular benchmark for evaluating SA models. This corpus is composed of 50.000 reviews, divided into two balanced splits, one for training and one for testing. Each split has 25000 reviews, and in each split, there are 12.500 positive reviews and 12.500 negative reviews. A review is considered negative, and therefore, labelled with the tag 0 if the score is less than four. Whereas positive reviews, labelled with the tag 1, are those with a score greater than seven. Therefore, neutral rated reviews are not included in this corpus.

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

In Figure 7.7, the length of the samples found in this dataset are shown. On average, both the train and the development splits have relatively short samples. The mean is 238.7 words per review in the training dataset and 230.8 in the development split. However, the outliers are numerous and significative longer, pushing the standard deviation to 176.5 and 169.2, respectively. As we discussed previously, this requires padding the sentences to ensure that all the inputs have the same size. Noteworthy, padding the sentences of this corpus presents similar difficulties to the problem that we tackled in 7.1. Nevertheless, for simplicity here, we have padded the sentences with a vector of zeros. The maximum length was a hyperparameter set to 200.

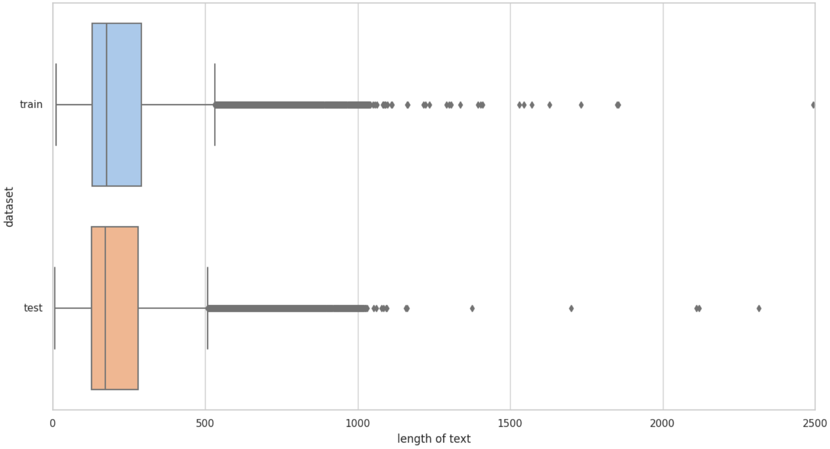


Figure 7.7: Distribution of the sentences length in the dataset imdb.

The model that we have proposed to visualise in this work to validate the framework introduced replicates the architecture of a well-known model in text classification proposed by Kim, 2014 that we presented earlier in this chapter. A diagram of this model can be found in 7.3. This CNN architecture has a convolutional layer with 100 filters of size $\mathbb{R}^{3\times e}$, $\mathbb{R}^{4\times e}$, and $\mathbb{R}^{5\times e}$ respectively, being e the size of the embeddings with rectified linear (RELU) units, it is regularised by means of a dropout layer with a probability of 0.5, and the fully connected layer consisted on 150 neurons. The model was also trained using mini-batches of size 100 through Adam gradient descent. Custom convolutional and the max-pooling layers were implemented using Keras (Chollet et al., 2015) and TensorFlow (Martín Abadi et al., 2015) to allow saving and analysing the activation maps.

The word embeddings used were pre-trained embeddings based on the skipgram version of word2vec proposed by Mikolov, K. Chen, et al. (2013). In this case, we applied embeddings trained using Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases [5] All the words OOV were assigned a random vector sampled from a Gaussian of mean 0 and standard deviation of 0.25.

Results

The objective of the first experiment we carried out was to select the interpretability threshold. Initially, the architecture described was trained until the model achieved a satisfactory F_1 of 0.864. Using this trained model, we applied the iterative algorithm described previously.

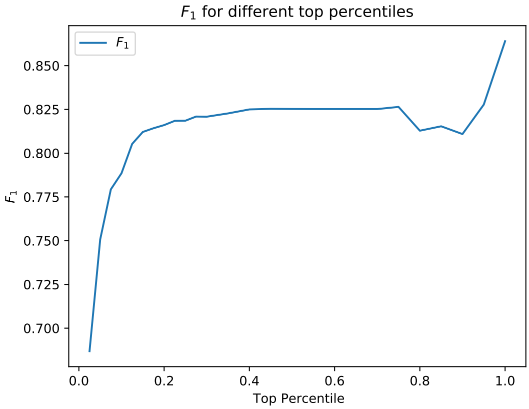
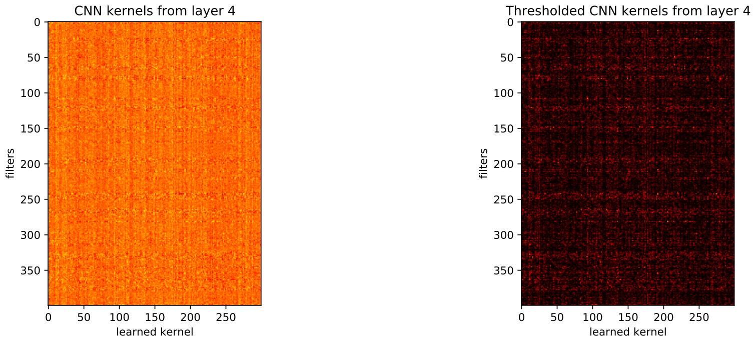


Figure 7.8: Evolution of the F_1 selecting different thresholds.

Figure 7.8 shows the decay in the achieved F_1 as we decreased the top percentile of neurons that are kept active. Interestingly, the F_1 plummeted when we masked a small percentile of filters. However, the F_1 plateau until only less than 20 % of the filters is still active. From that point, going forward, rapid decay is appreciated. This behaviour could be explained because the CNN architecture used has a high capacity for modelling the problem and, therefore, high tolerance for missing connections. Besides, it is noteworthy to highlight that shorter sentences will have more positions filled with padding; therefore, masking these positions will

[5] This pre-trained embeddings can be downloaded from `https://code.google.com/archive/p/word2vec/`



(a) Learned weighths.

(b) Thresholded weighths.

Figure 7.9: Weigths from the convolutional layer with a kernel of size four before and after thresholding the top percentile.

not have any impact on the performance of the model. Nevertheless, a more in-depth study of this phenomenon is required.

We set the top percentile to 0.3. Masking the positions below the threshold of this top percentile, the model achieved an 82.08 %. Therefore, the thresholded model is performing only a 5 % worse than the original model. Notwithstanding, this loss in performance can be justified because the thresholded model is more easily interpretable. The weights of one of the convolutional layers before and after thresholding is applied can be seen in Figure 7.9. Unlike in computer vision, interpreting these filters is not straightforward. Hence, the analytical framework proposed will allow investigating what the model learned systematically.

To illustrate an example of what the convolutional neural layers are learning, Figure 7.10 shows a truncated example of the activation map for one of the sentences in the corpora. The rows correspond to the n-gram used by the filter, and the columns are each of the 100 filters for a layer. Brighter positions in the heatmap indicate the relevance of the n-gram in the filter. In the two highlighted areas, where the filters were slightly more active, sentiment loaded words can be found. This is an anecdotal example. Nevertheless, it shows certain redundancy across the filters in a layer and that the filters are paying attention to particular n-grams.

Notably, shorter sentences imply having most of the input matrix padded, and therefore the activation map presents more deactivated positions. As discussed previously in this chapter, using that region with semantic-based padding could

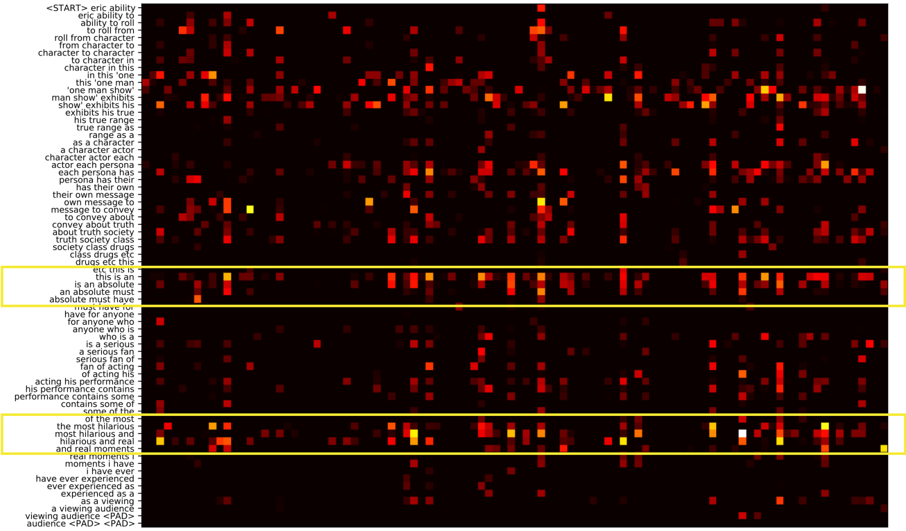


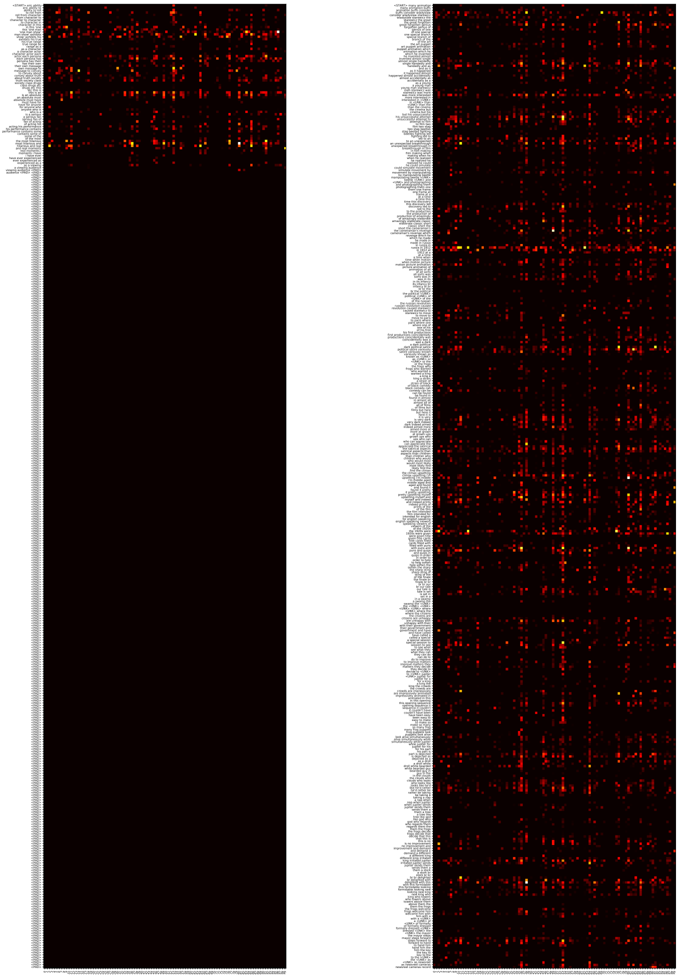
Figure 7.10: Evolution of the F_1 selecting different thresholds.

boost the performance of the model. Figure 7.11 presents the activation map of two contrasting sentences, a short and a long sentences. As expected, most of the activation map is deactivated –therefore black– in the example of the shorter sentence.

To resume the analytical study, we computed the interpretability metrics presented in 7.2.1. The annotation tags considered in this experiment were part of speech tags (POS). The library NLTK was used to label the words in the corpus (Bird, Klein, and Loper, 2009).

Firstly, we explored the interpretability of the filters where the class of the examples was predicted correctly, as defined in 7.2. Forty POS tags were observed in the corpus. The least relevant POS tags were symbols (SYM), proper nouns in the plural (NNPS), interjections (UH), possessive interrogatives pronouns (WP$) and the padding positions. Contrastly, the most relevant POS tags found were nouns in the singular (NN), adjectives (JJ), preposition conjunction (IN), determiner (DT), and adverbs (RB). Table 7.10 presents the mean and the standard deviation of the ten most relevant POS tags aggregated by the class predicted correctly.

Noteworthy, the filters were more activated for words that identify nouns and sentiment charged words like adjectives and adverbs. Nevertheless, it is unexpected



(a) Short sentence.

(b) Long sentence.

Figure 7.11: Activation maps of two sentences in the convolutional layer with a kernel of size three.

POS tag	Class	$Interpretability_{k,c,t}$
NN	Negative	23.88 % (3.99)
	Positive	24.81 % (4.23)
JJ	Negative	10.85% (1.40)
	Positive	11.34 % (1.49)
IN	Negative	10.39 % (2.19)
	Positive	10.66 % (2.11)
DT	Negative	8.97 % (2.17)
	Positive	8.49 % 2.19
RB	Negative	5.69 % (1.52)
	Positive	5.44 % (1.39)
NNS	Negative	5.16 % (0.93)
	Positive	5.33 % (1.39)
VBZ	Negative	4.43 % (1.06)
	Positive	4.96 % (1.12)
CC	Negative	4.66 % (3.45)
	Positive	4.89 % (3.49)
VBD	Negative	3.79 % (0.92)
	Positive	3.30 % (0.77)
VB	Negative	2.97 % (1.56)
	Positive	2.51 % (1.33)

Table 7.10: Interpretability of the top ten POS tags considered examples correctly predicted. The abbreviations of the POS tags are: nouns in singular (NN), adjectives (JJ), prepositions (IN), determiners (DT), adverbs (RB), nouns in plural (NNS), conjunctions (CC), verbs in present tense and in third person singular (VBZ), verbs in past tense (VBD), and verbs in base form (VB).

that stop words such as determinants or prepositions also appear among the most relevant words in the correct classification.

Figure 7.12 plots the quartiles of the filters in each layer for considering the class correctly predicted and the POS tag. For simplicity, we narrow the visualisation to the top five POS tags. As can be seen, there are no significant differences between the positive and negative reviews. Besides, this boxplot shows that there are not any relevant outliers among the filters in any layer. Therefore, we have not found a filter that specialises in detecting a particular POS tag.

Figure 7.13 shows the activation map of each filter aggregated as we did in Figure 7.12. This plot confirms what we have previously seen in the preceding Figure

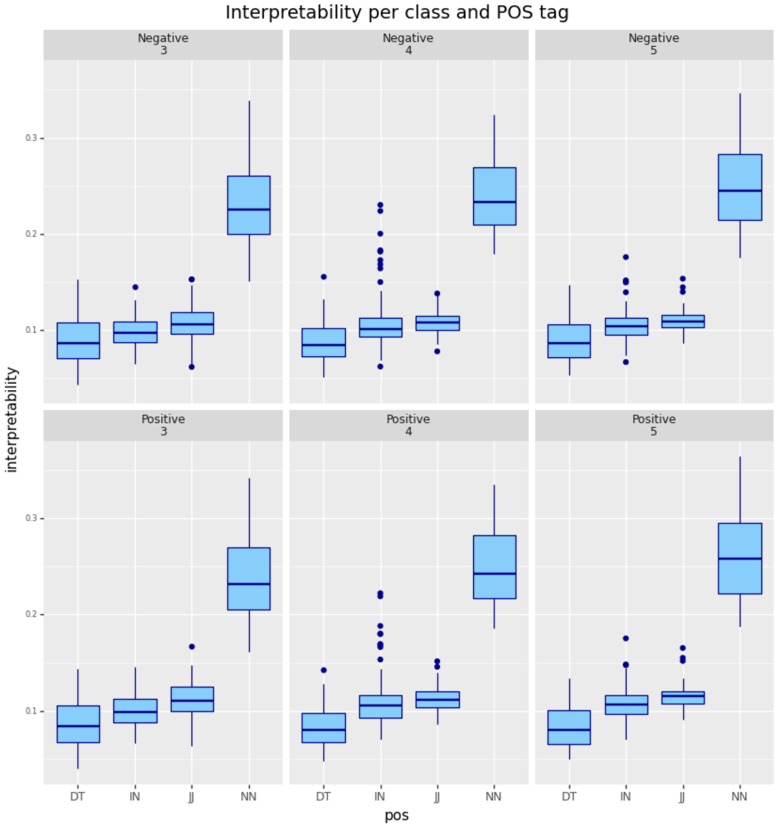


Figure 7.12: Distribution of the interpretability aggregated per tag and class.

7.13. Nonetheless, this plot shows the disaggregated interpretability of each filter. It can be appreciated how the most interpretable POS tag is NN across all the filters and all the classes. Interestingly, some filters are more specialised in prepositions (IN), particularly in the layer with kernel size four.

Following, we analyse the results of computing the interpretability considering the POS tag independently if the classification was correct or not, as defined in Equation 7.3.

Table 7.11 presents the mean and the standard deviation interpretability of the filters aggregated by the POS tags. Noteworthy, the ten most relevant POS tags found when the class was predicted correctly – shown in Table 7.10– are the same POS tags found in this metric, indicating that the internal representation of a

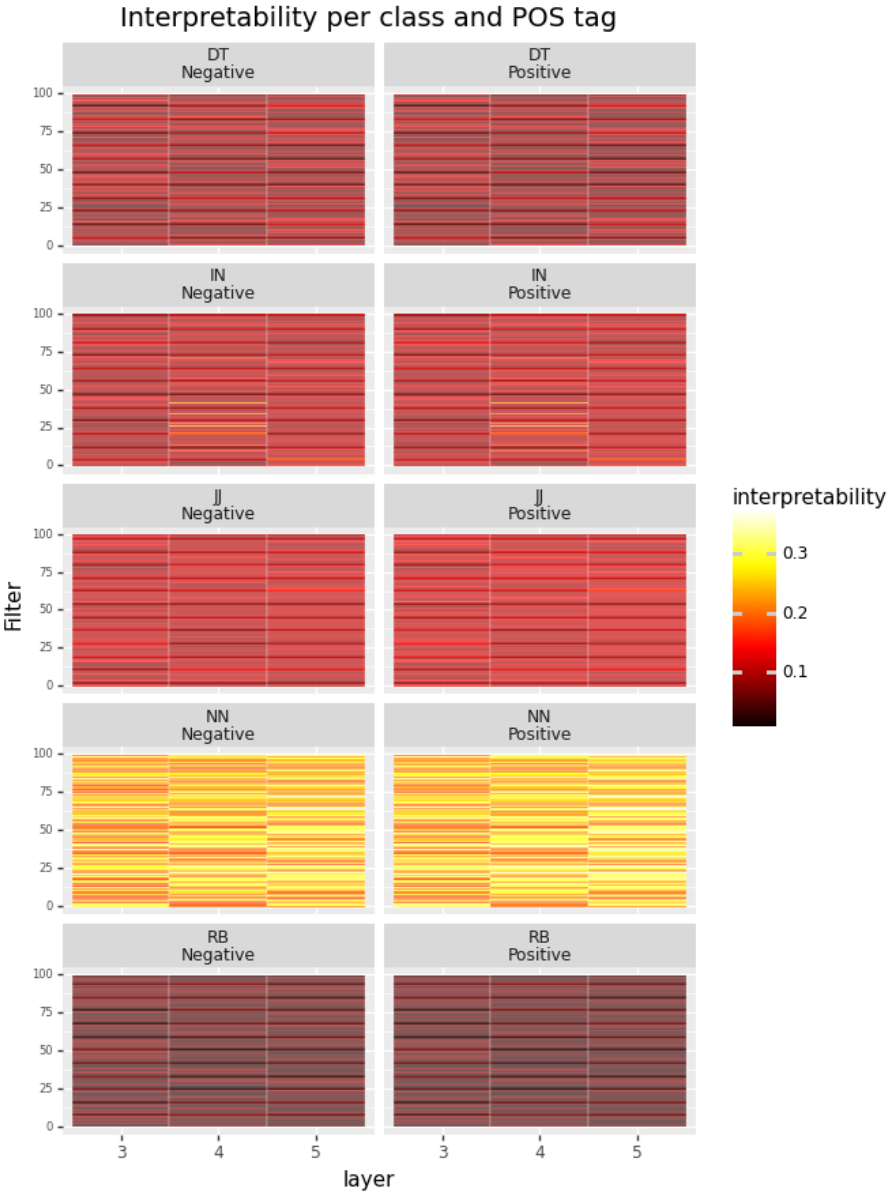


Figure 7.13: Interpretability of each filter in the CNN aggregated per tag and class.

sentence in the convolutional layers are attending to the same POS tags regardless of the sentiment of a sentence.

POS tag	$Interpretability_{k,t}$
NN	24.34 % (4.13)
JJ	11.1 % (1.46)
IN	10.55 % (2.13)
DT	8.69 % (2.17)
RB	5.55 % (1.44)
NNS	5.22 % (0.88)
CC	4.75 % (3.43)
VBZ	4.70 % (1.08)
VBD	3.54 % (0.84)
VB	2.73 % (1.43)

Table 7.11: Interpretability of the top ten POS tags.

The relevance of the POS tags in the internal representation of the convolutional layer can also be seen in Figure 7.14, where again the nouns in the singular are the most relevant and some filters are slightly more interpretable for prepositions.

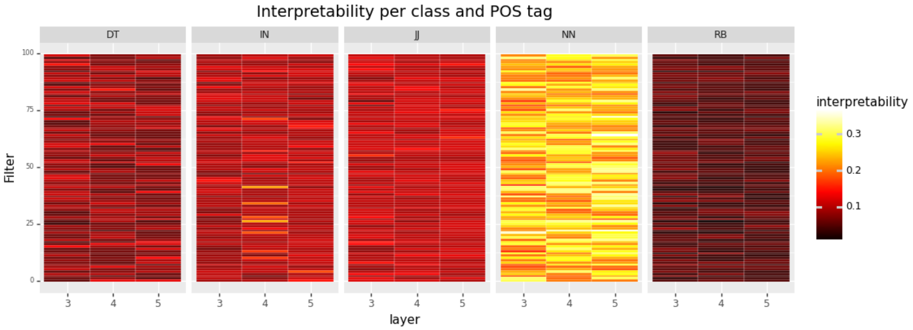


Figure 7.14: Interpretability of each filter in the CNN aggregated per tag.

Continuing our analysis to interpret what the internal layers of the convolutional neural network learned, we examined the capacity of each filter to predict the class of the sentence correctly as defined in Equation 7.4.

As we did previously, Table 7.12 shows the mean and the standard deviation of the interpretability of the filters that correctly predicted the class of the movie review. Meanwhile, Figure 7.15, presents the individual interpretability of each filter considering the class predicted.

Class	$Interpretability_{k,t}$
Negative	44.21 % (3.11)
Positive	40.54 % (3.35)

Table 7.12: Interpretability of the filters considering examples correctly predicted.

From these two visualisations, it could be extrapolated that filters that predicted correctly positive reviews were more activated than those that predicted correctly negative reviews. In particular, it can be observed that filters with kernel size four and five, these are filters attending to 4-grams and 5-grams, were more activated in the positive class. Contrastingly, filters that predicted negative reviews had the 3-gram filters more activated, revealing a preference for shorter sequences.

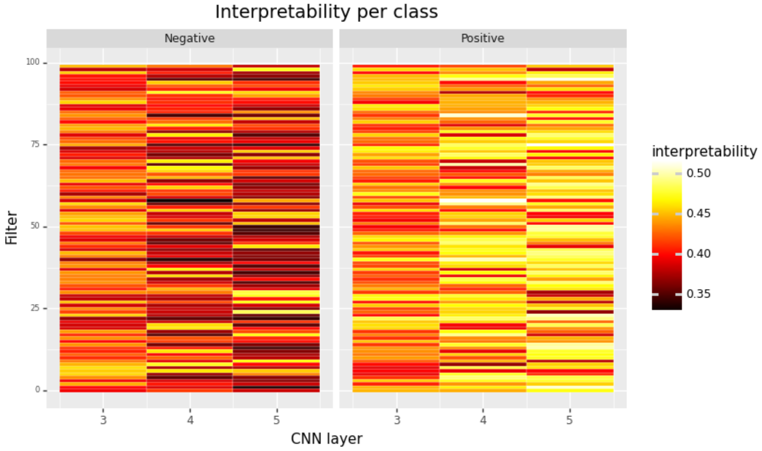


Figure 7.15: Interpretability of each filter in the CNN aggregated per class.

To conclude our study of the interpretability of the internal representation of the convolutional neural networks, the computation of the performance metrics for the fully connected layers, as defined in 7.6, was carried out.

Table 7.13 presents the mean and the standard of these metrics. The filters have high precision but low recall, which affects the final F_1, as shown in Figure 7.16. It should be noted that each layer of the convolutional network presents different F_1, being the convolutional layer with a kernel of size three the layer that achieves

Accuracy	Precision	Recall	F_1
17.62 % (4.75)	84.76 % (0.66)	5.52 % (6.63)	9.61 % (10.56)

Table 7.13: Interpretability metrics of the convolutional layers.

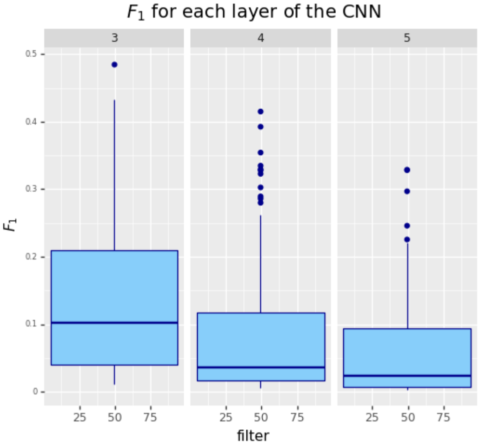


Figure 7.16: F_1 of each filter in the CNN aggregated per layer.

the best performance. Moreover, in each layer, outliers filters are found. Studying these outliers filters can lead to interesting findings.

Finally, we studied the relevance of the classification of the fully connected layers. In this model, two fully connected layers can be found. The first one has 150 neurons, and the second one has two correspondings to the two classes that needed to be predicted. We have studied the first dense layer, as proposed in Equation 7.8. We observed that only 11.11 % of the neurons were active after filtering the least active kernels in the convolution. This result reinforces the idea that the network was over-dimensioned for the problem at hand because it only needed a small fraction of its neurons active to predict the result correctly.

An interesting side-effect of pruning the least relevant learned weights in the convolutional neural layer is that the number of non-zero features in the convolutional layers increases after deactivating the weights and biases below the threshold. The non-zero thresholded convolutional neurons constituted 5.18 % of the total, whilst the non-zero convolutional neurons before the thresholding constituted 5.6 %. The variation is slight, but it is a side-effect worth studying in future work that might be able to explain the artefact seen in Figure 7.8. This pattern is propagated through the network, and the first dense layer has 0.34 % more neurons active after filtering the least relevant neurons.

Discussion and Conclusions

In summary, we have proposed a new interpretability framework that seeks to shed light on what the internal layers of a convolutional neural network are learning in an NLP task.

The framework has been validated using a sentiment analysis task. Among the most relevant findings is the resilience of the network, which was evidenced in the iterative algorithm that selected the interpretability threshold –most of the weights needed to be disabled in order to see the performance plummeted.

Another relevant finding was the validation that certain POS tags such as nouns, adjectives and verbs were relevant in the internal representation of the convolutional layers. Intuitively, this finding could have been foreseen, but we have validated it methodologically. Contrastingly, categories of stop words were also relevant. Future work should indagate fine-grained categories of words or study the effect of each word directly and explain the relevance of stop words that are commonly filtered out. Moreover, there were no clear outliers in any category studied. Filters in each layer had similar values; this will reinforce the notion that the network trained had ample redundancy. Nevertheless, this also signals a lack of specialisation, discarding the hypothesis that certain filters learn specific concepts.

7.3 Conclusions

To conclude, in this chapter, we have presented two studies of Convolutional Neural Networks for NLP tasks. Mainly, we have focused our efforts on Sentiment Analysis Tasks.

In the first part, we have proposed semantic-based padding and validated its boost in the performance of SA models. In this second part, we have carried out a study to interpret the internal representation of CNN models and analyse what the model was learning. Both efforts had the same objective, to scrutinise DL models, concretely CNN, which was the focus of this thesis.

Being able to use critically and reason about the models that we develop is critical, and the research should be focused not only on advancing the state-of-the-art but also on making this advance fair and robust. Architectures are tailor-made for a problem and then used in another domain should be applied critically considering the nuances of the new domain.

In the last years, we have seen DL applications ubiquitously. Sorrowfully, we have seen plenty of cases where the models have failed to provide unbiased, ethical, and fair results. Therefore, all the effort to study and interpret models that impact so many aspects of our lives is crucial in the field's future development.

Chapter 8

Conclusions and Future Work

In this final chapter, we summarise the contributions achieved and the main discussions that arose from the work presented. Besides, we listed the publications derived from the research that was carried out during this thesis. Finally, future lines of work that could extend from the contributions introduced here are described.

8.1 Conclusions of the work presented

As we introduced at the beginning, our motivation was broad. We wanted to explore how NLP systems could evolve and perform in the new paradigm of deep learning that was getting very popular in other areas of study like computer vision. Therefore, we implemented end-to-end deep learning models focusing on two particular tasks Sentiment Analysis and Personality Recognition. Moreover, we proposed a study on how text could be structured for improving the performance of Convolutional Neural Networks used in NLP tasks. Finally, we deepen our study of CNN applied in NLP tasks proposing a new framework for interpretability.

In this section, the contributions presented in this work are summarised.

Natural Language Processing Evaluation Task. In this work, we curated and labelled a dataset for topic classification in the context of the 2015 electoral cycle and organised a shared evaluation task where seventeen teams participated in proposing their different models. As a result of this effort, a new body of work was developed for topic classification in Spanish. In addition, the corpus annotated automatically using the best performing models has been used in political research (Baviera, 2017; Baviera, 2018; Baviera Puig, Calvo, and Llorca-Abad, 2019).

Sentiment Analysis in Social Media. Our contributions to sentiment analysis were the development of machine learning models, concretely Support Vector Machine models trained with adhoc features manually selected. This research shows that a handcrafted model requires adjusting the features and resources selected for each new task which limits the ability to generalise to new domains. Nevertheless, these less intricate models studied in this part of the work allowed us to investigate how different language devices, such as irony, hinders the ability of a model to predict the sentiment conveyed in a text. Furthermore, we presented a study on how resources that encode emotions helps to classify the sentiment present in a text correctly.

Personality Recognition. Unlike sentiment analysis which had a large body of study, personality recognition is a less investigated field. Here we have contributed to enlarge the literature on the topic addressing two different setups. On the one hand, we proposed a model for the very challenging task of inferring the personality of a developer given their source code. On the other hand, we proposed two approaches for predicting the personality of a user considering the texts they share on social media. A classical ML approach and a DL approach were studied and compared. The CNN model that we proposed performed similarly as the best performing models that rely on handcrafted features which are closely dependent on the domain and the availability of resources for a given domain and language. Therefore, we have proven that even in this challenging setup, deep learning models can generalise without the need for handcrafted resources customised for a particular task.

Study of Convolutional Neural Networks for Natural Language Processing. Finally, we studied in depth how convolutional neural networks were applied in natural language processing tasks. As a result of this research, we have proposed semantic-based paddings for NLP tasks and validated its performance against state-of-the-art models. For each experiment performed, the semantic-based padding proved to improve the performance of the system. Moreover, when the model without padding achieved state-of-the-art performance, the semantic-based padding outdoes the accuracy reported in state of the art. To conclude, we tackled one of the most common critiques to deep learning models; this is the lack of interpretability. In order to address this concern, an interpretability framework was proposed. This framework allowed us to understand both the convolutional neural layers and the fully connected layer of a state-of-the-art model. Moreover, we proposed an iterative algorithm to analyse which parts of the network were relevant for the classification.

8.2 Derived publications

As a result of the research here presented, the following publications were derived. We aggregate the publications considering where they were published.

Publications in indexed journals JCR-SCI:

- Maite Giménez, Javier Palanca, and Vicent Botti (2020). "Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis". In: *Neurocomputing* 378, pp. 315–323

  In this paper we presented the semantic-based padding described in section 7.1.

- M$^{\text{a}}$. Amparo Escortell Pérez, Maite Giménez Fayos, and Paolo Rosso (2017). "El impacto de las emociones en el análisis de la polaridad en textos con lenguaje figurado en Twitter". In: *Procesamiento del Lenguaje Natural. N. 58 (2017)*

  This work investigates the impact of emotions in a Sentiment Analysis task. We exposed this work in section 5.2.

Publications in ranked conferences CORE:

- Maite Giménez, Roberto Paredes, and Paolo Rosso (2017). "Personality Recognition Using Convolutional Neural Networks". In: *International Conference on Computational Linguistics and Intelligent Text Processing.* Springer, pp. 313–323

  Here we introduced the use of CNNs for solving the personality recognition task. The findings from this work were presented in 6.3.2.

- Maite Giménez, Delia Irazú Hernández, and Ferran Pla (2015). “Segmenting Target Audiences: Automatic Author Profiling Using Tweets.” In: *Proceedings of CLEF*

  This work explored the task of personality recognition from source code, and it was explained in 6.2.

Publications in non ranked conferences:

- Maite Giménez, Tomás Baviera, et al. (2017). “Overview of the 1st classification of spanish election tweets task at ibereval 2017”. In: *Notebook Papers of 2nd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL), Murcia, Spain, September.* Vol. 19

  In this publication, the shared evaluation campaign that we organised and the results achieved by participants are summarised. We discussed this work in chapter 4.

- Maite Giménez and Roberto Paredes (2016). “PRHLT at PR-SOCO: A Regression Model for Predicting Personality Traits from Source Code.” In: *FIRE (Working Notes)*, pp. 38–42

  This work explored solutions for the personality recognition task from source code, and the research carried on was presented in 6.2.

- Maite Giménez, Ferran Pla, and Lluís-F Hurtado (2015). “Elirf: a SVM approach for SA tasks in twitter at SemEval-2015”. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 574–581

  Here we proposed a classical machine learning model trained with handcrafted features to infer the sentiment, and it summarised our participation in a public evaluation campaign. This work was addressed in 5.1.

Pending

- Maite Giménez, Raül Fabra, et al. (Status: Under Review). “A fine-grained study of interpretability of Convolutional Neural Networks for text classification”. In: *Knowledge-Based Systems*

  Finally, in this work, we proposed an interpretability framework to help to understand what convolutional neural networks are learning. This research was presented in 7.2.

Moreover, during the development of this thesis, we collaborated with other researchers, which lead to the following publications, which are nevertheless out of the scope of the work presented here. Therefore, we only list them here for completeness.

- Maite Giménez, Jaume Jordán, et al. (2018). "Rassel: Robot Assistant for the Elderly". In: *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, pp. 5–9
- Marius Andrei Ciurez et al. (2019). "Automatic Categorization of Educational Videos According to Learning Styles". In: *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, pp. 1–6

8.3 Future work

The time available to work on a thesis is limited, and of course, there were lines of research that we could not explore thoroughly and new lines that were opened recently. In the last years, we have seen a massive development in NLP propelled by the availability of extensive corpora that could be processed thanks to the popularisation of faster and cheaper hardware such as GPUs and TPUs. Future researchers in NLP will have access to all these advances and many more that we cannot even fathom. Nevertheless, during the years that we worked on this thesis, we have accumulated some wisdom, and there are some lines of research that are worth exploring.

In this final section, we list the future lines of work that we could not explore but which might be useful to the reader who pursues to improve NLP models. The future work is listed following the same order as the chapters in this document.

Firstly, we would like to encourage researchers to openly publish and participate in public campaigns when this possibility is available. Shared evaluations allow comparing fairly different approaches. Moreover, in public evaluation campaigns, negative results are not discouraged from being published. Hence, public campaigns extend the body of the research in a field both with positive and negative results. In addition, the development of a new corpus to tackle a particular topic of research or in a less served language enables the progress of new lines of research that might impact larger communities. As we mentioned previously, the dataset that we curated ignited lines of research in the social science field. Therefore, developing and making available datasets in collaboration with other researchers, particularly in minority languages, might lead to unlocking better models and solving new problems.

Interestingly, our study of sentiment analysis revealed how different language devices, such as irony or humour, affects the performance of models. Provided that the capacity of new language models have increased massively since we carried out those experiments, a relevant study nowadays will be to evaluate if these new models such as BERT(Devlin et al., 2018) or GPT-3(Brown et al., 2020) are robust enough to predict the sentiment of a text that presented figurative language. Yadav and Vishwakarma (2020) presented an interesting survey on deep learning models applied to sentiment analysis. The work done in this area is extensive. However, beyond the figurative language, most of the research is focused on the most spoken languages. Therefore, it is worth exploring models that address sentiment analysis in different languages.

Regarding personality analysis, it remains a relatively understudied field. One of the reasons for this is the ethical dilemmas that this area of study represent (Mukherjee and Kumar, 2016). Therefore, all the research done in this area should be thoroughly studied by researchers with different backgrounds that should include social sciences, psychology and philosophy. Besides, mitigations should be put in place, preventing the misuse of the models developed. Considering all these premises, there are multiple lines of research to explore in this field. Practitioners could evaluate the behaviour of the latest deep learning models in this task, provided that richer datasets were collected. Moreover, this task is mainly tackled in English datasets. Hence, once again, we must compel researchers to study different languages. The impact that this task has in numerous domains worth exploring which models could solve it. However, it is worth emphasising the need to innovate responsibly.

As a final point in these lines of research to explore in the future, we discuss the work that can be done to have a better understanding of DL models. Noteworthy, the lack of interpretability is one of the most commonly cited problems that DL presents. Being able to audit and understand the decisions of the models that we put in production and, in the end, takes decisions about our lives is crucial. In addition, if we understand the models better, we could improve their performance, introducing advances that tackle known limitations. We have focused our efforts to study the nuances of convolutional neural networks implemented to NLP. However, it will be interesting to investigate if the semantic-based padding overperforms considering other models from the state-of-the-art that also require a fixed input length. Similarly, being able to interpret the internal layers of other models will be noteworthy. The iterative algorithm, presented in section 1, could be easily adapted to other architectures as well as the interpretability metrics proposed. Furthermore, we have focused our research on sentiment analysis tasks. We selected this task because, as we mentioned several times already, SA is one of the most well-known tasks in NLP, with a comprehensive list of resources that allowed us to focus on developing the correct study. Nevertheless, deepen the study considering in other domains or even other tasks might lead to new advances in the development of NLP models.

In conclusion, the advances seen in NLP are a guide to explore the development of the field, pursuing new and well-established tasks. Users interact using natural language constantly, and we should aim to provide better responsible models in their languages.

Bibliography

Abdaoui, Amine et al. (2017). “Feel: a french expanded emotion lexicon”. In: *Language Resources and Evaluation* 51.3, pp. 833–855 (cit. on p. 24).

Aggarwal, Charu C and ChengXiang Zhai (2012). *Mining text data.* Springer Science & Business Media. Chap. 6 (cit. on p. 29).

Agirre, Eneko et al. (2014). “Semeval-2014 task 10: Multilingual semantic textual similarity”. In: *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 81–91 (cit. on p. 33).

Alba, Carlos Diez and Jesús Vieco Pérez (2017). “IberEval 2017, COSET Task: A Basic Approach.” In: *IberEval SEPLN*, pp. 61–67 (cit. on pp. 45, 46).

Alghamdi, Rubayyi and Khalid Alfalqi (2015). “A survey of topic modeling in text mining”. In: *Int. J. Adv. Comput. Sci. Appl.(IJACSA)* 6.1 (cit. on p. 30).

Almeida, Felipe and Geraldo Xexéo (2019). “Word embeddings: A survey”. In: *arXiv preprint arXiv:1901.09069* (cit. on pp. 10, 11).

Alvarez-Carmona, Miguel A et al. (2015). “INAOE’s participation at PAN’15: Author profiling task”. In: *Working Notes Papers of the CLEF* (cit. on p. 103).

Ambrosini, L. and G Nicoló (2017). “Comparative study of neural models for the COSET shared task at IberEval 2017”. In: (cit. on pp. 46, 47, 50).

Amigó, Enrique et al. (2017). “Evall: Open access evaluation for information access systems”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1301–1304 (cit. on pp. 44, 49).

Analytics, Pear (2009). *Twitter study.* URL: `https://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf` (visited on 12/02/2012) (cit. on p. 56).

Araque, Oscar et al. (2017). “Enhancing deep learning sentiment analysis with ensemble techniques in social applications”. In: *Expert Systems with Applications* 77, pp. 236–246 (cit. on p. 16).

Argamon, Shlomo, Sushant Dhawle, et al. (2005). “Lexical predictors of personality type”. In: *Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America*, pp. 1–16 (cit. on pp. 28, 85, 86).

Argamon, Shlomo, Moshe Koppel, Jonathan Fine, et al. (2003). “Gender, genre, and writing style in formal written texts”. In: *Text & Talk* 23.3, pp. 321–346 (cit. on p. 28).

Argamon, Shlomo, Moshe Koppel, James W Pennebaker, et al. (2007). “Mining the blogosphere: Age, gender and the varieties of self-expression”. In: *First Monday* (cit. on p. 28).

Arrieta, Alejandro Barredo et al. (2020). “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information Fusion* 58, pp. 82–115 (cit. on p. 36).

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). “Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.” In: *Lrec.* Vol. 10. 2010, pp. 2200–2204 (cit. on p. 64).

Bachrach, Yoram et al. (2012). “Personality and patterns of Facebook usage”. In: *Proceedings of the 4th annual ACM web science conference*, pp. 24–32 (cit. on p. 28).

Balahur, Alexandra and Marco Turchi (2012). “Multilingual sentiment analysis using machine translation?” In: *Proceedings of the 3rd workshop in compu-*

tational approaches to subjectivity and sentiment analysis. Association for Computational Linguistics, pp. 52–60 (cit. on p. 24).

Balahur, Alexandra and Marco Turchi (2014). “Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis”. In: *Computer Speech & Language* 28.1, pp. 56–75 (cit. on p. 24).

Barbieri, Francesco et al. (2016). “Overview of the evalita 2016 sentiment polarity classification task”. In: (cit. on p. 26).

Barbosa, Luciano and Junlan Feng (2010). “Robust sentiment detection on twitter from biased and noisy data”. In: *Proceedings of the 23rd international conference on computational linguistics: posters.* Association for Computational Linguistics, pp. 36–44 (cit. on p. 23).

Barnes, Jeremy, Roman Klinger, and Sabine Schulte im Walde (2017). “Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets”. In: *arXiv preprint arXiv:1709.04219* (cit. on pp. 112, 116).

Basile, Pierpaolo et al. (2018). “Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA)”. In: *EVALITA Evaluation of NLP and Speech Tools for Italian* 12, p. 10 (cit. on p. 26).

Basile, Valerio et al. (2014). “Overview of the evalita 2014 sentiment polarity classification task”. In: (cit. on p. 26).

Basu, Moumita, Saptarshi Ghosh, and Kripabandhu Ghosh (2018). “Overview of the fire 2018 track: Information retrieval from microblogs during disasters (irmidis)”. In: *Proceedings of the 10th annual meeting of the Forum for Information Retrieval Evaluation*, pp. 1–5 (cit. on p. 34).

Bau, David et al. (2017). “Network dissection: Quantifying interpretability of deep visual representations”. In: *arXiv preprint arXiv:1704.05796* (cit. on pp. 36, 121, 123).

Baviera, Tomás (2017). “Técnicas para el Análisis de Sentimiento en Twitter: Aprendizaje Automático Supervisado y SentiStrength”. In: *Revista Dígitos* 1.3, pp. 33–50 (cit. on p. 144).

Baviera, Tomás (2018). “Influence in the political Twitter sphere: Authority and retransmission in the 2015 and 2016 Spanish General Elections”. In: *European Journal of Communication* 33.3, pp. 321–337 (cit. on p. 144).

Baviera, Tomás, Dafne Calvo, and Germán Llorca-Abad (2019). “Mediatisation in Twitter: an exploratory analysis of the 2015 Spanish general election”. In: *The Journal of International Communication* 25.2, pp. 275–300 (cit. on pp. 42, 52).

Baviera Puig, Tomás, Dafne Calvo, and Germán Llorca-Abad (2019). “Twitter Dataset-2015 Spanish General Election”. In: (cit. on p. 144).

Baziotis, Christos, Nikos Pelekis, and Christos Doulkeridis (2017). “Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 747–754 (cit. on p. 27).

Bengio, Yoshua, Réjean Ducharme, et al. (2003). “A neural probabilistic language model”. In: *Journal of machine learning research* 3.Feb, pp. 1137–1155 (cit. on pp. 10, 115).

Bengio, Yoshua, Jean-Sébastien Senécal, et al. (2003). “Quick Training of Probabilistic Neural Nets by Importance Sampling.” In: *AISTATS*, pp. 1–9 (cit. on p. 10).

Bengio, Yoshua, Patrice Simard, Paolo Frasconi, et al. (1994). “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE transactions on neural networks* 5.2, pp. 157–166 (cit. on p. 16).

Bernath, C. (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: ConradCR”. In: *IberEval SEPLN* (cit. on pp. 45, 46).

Berscheid, Ellen (1980). “Emotion”. In: *Psyccritiques* 25.10, pp. 779–780 (cit. on pp. 72, 73).

Bestgen, Yves et al. (2008). “Building Affective Lexicons from Specific Corpora for Automatic Sentiment Analysis.” In: *LREC* (cit. on p. 24).

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc." (cit. on pp. 74, 132).

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152 (cit. on p. 14).

Boyle, Gregory J (1995). "Myers-Briggs type indicator (MBTI): some psychometric limitations". In: *Australian Psychologist* 30.1, pp. 71–74 (cit. on p. 27).

Boyle, Gregory J, Gerald Matthews, and Donald H Saklofske (2008a). *Handbook of Personality Theory and Assessment: Vol. 2: Personality Measurement and Assessment.* SAGE Publications, Limited (cit. on p. 82).

Boyle, Gregory J, Gerald Matthews, and Donald H Saklofske (2008b). *The SAGE Handbook of Personality Theory and Assessment: Personality Measurement and Testing (Volume 2).* Vol. 2. Sage (cit. on p. 27).

Breiman, Leo et al. (1984). *Classification and regression trees.* CRC press (cit. on p. 13).

Broman, Karl et al. (2017). "Recommendations to funding agencies for supporting reproducible research". In: *American statistical association.* Vol. 2 (cit. on p. 32).

Brown, Tom B et al. (2020). "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (cit. on p. 148).

Cambria, Erik et al. (2013). "New avenues in opinion mining and sentiment analysis". In: *IEEE Intelligent systems* 28.2, pp. 15–21 (cit. on p. 23).

Carvalho, Paula et al. (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.* ACM, pp. 53–56 (cit. on p. 25).

Castro, Santiago, Luis Chiruzzo, and Aiala Rosá (2018). "Overview of the HAHA Task: Humor Analysis Based on Human Annotation at IberEval 2018." In: *IberEval at SEPLN*, pp. 187–194 (cit. on p. 35).

Celli, Fabio, Elia Bruni, and Bruno Lepri (2014). “Automatic personality and interaction style recognition from facebook profile pictures”. In: *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1101–1104 (cit. on p. 96).

Celli, Fabio, Bruno Lepri, et al. (2014). “The workshop on computational personality recognition 2014”. In: *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1245–1246 (cit. on p. 28).

Celli, Fabio and Luca Rossi (2012). “The role of emotional stability in Twitter conversations”. In: *Proceedings of the workshop on semantic analysis in social media*, pp. 10–17 (cit. on p. 28).

Chadwick, Andrew (2017). *The hybrid media system: Politics and power*. Oxford University Press (cit. on p. 40).

Chen, Tao et al. (2017). “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN”. In: *Expert Systems with Applications* 72, pp. 221–230 (cit. on p. 16).

Chen, Yan et al. (2012). “A semi-supervised bayesian network model for microblog topic classification”. In: *Proceedings of COLING 2012*, pp. 561–576 (cit. on p. 31).

Chen, Yanqing and Steven Skiena (2014). “Building sentiment lexicons for all major languages”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 383–389 (cit. on p. 24).

Chiruzzo, Luis et al. (2019). “Overview of HAHA at IberLEF 2019: Humor Analysis based on Human Annotation.” In: *IberLEF at SEPLN*, pp. 132–144 (cit. on p. 35).

Chollet, François et al. (2015). *Keras*. `https://github.com/fchollet/keras` (cit. on p. 129).

Chuliá, Luis Cebrián and Sergio Ferrer Sánchez (2017). “Classification Of Spanish Election Tweets (COSET) with Neural Networks.” In: *IberEval SEPLN*, pp. 43–48 (cit. on pp. 45, 46, 49, 50).

Ciurez, Marius Andrei et al. (2019). “Automatic Categorization of Educational Videos According to Learning Styles”. In: *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM).* IEEE, pp. 1–6 (cit. on p. 147).

Clark, Alexander, Chris Fox, and Shalom Lappin (2013). *The handbook of computational linguistics and natural language processing.* John Wiley & Sons (cit. on p. 48).

Clematide, Simon et al. (2010). “Evaluation and extension of a polarity lexicon for German”. In: (cit. on p. 24).

Collobert, Ronan et al. (2011). “Natural language processing (almost) from scratch”. In: *Journal of machine learning research* 12.ARTICLE, pp. 2493–2537 (cit. on pp. 10, 16, 98).

Conover, Michael D et al. (2011). “Predicting the political alignment of twitter users”. In: *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing.* IEEE, pp. 192–199 (cit. on p. 31).

Conway, Bethany A, Kate Kenski, and Di Wang (2015). “The rise of Twitter in the political campaign: Searching for intermedia agenda-setting effects in the presidential primary”. In: *Journal of Computer-Mediated Communication* 20.4, pp. 363–380 (cit. on p. 30).

Costa Jr, Paul T and Robert R McCrae (2008). *The Revised NEO Personality Inventory (NEO-PI-R).* Sage Publications, Inc (cit. on p. 82).

Cruz-Roa, Angel Alfonso et al. (2013). “A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, pp. 403–410 (cit. on p. 121).

Cui, Anqi et al. (2011). “Emotion tokens: Bridging the gap among multilingual twitter sentiment analysis”. In: *Asia information retrieval symposium.* Springer, pp. 238–249 (cit. on p. 24).

Danilevsky, Marina et al. (2020). "A Survey of the State of Explainable AI for Natural Language Processing". In: *arXiv preprint arXiv:2010.00711* (cit. on p. 37).

De la Peña Sarracén, Gretel Liz (2017). "Ensembles of Methods for Tweet Topic Classification." In: *IberEval SEPLN*, pp. 15–19 (cit. on pp. 46, 47).

De Winter, Joost CF (2013). "Using the Student's t-test with extremely small sample sizes." In: *Practical Assessment, Research & Evaluation* 18.10 (cit. on p. 116).

Deng, Li and Yang Liu (2018). *Deep Learning in Natural Language Processing.* Springer (cit. on p. 106).

Deng, Li and Dong Yu (2014). "Deep learning: methods and applications". In: *Foundations and trends in signal processing* 7.3–4, pp. 197–387 (cit. on p. 15).

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (cit. on pp. 27, 148).

Dos Santos, Cicero and Maira Gatti (2014). "Deep convolutional neural networks for sentiment analysis of short texts". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78 (cit. on p. 26).

Drummond, Chris (2009). "Replicability is not reproducibility: nor is it good science". In: (cit. on p. 31).

Dubois, Elizabeth and Devin Gaffney (2014). "The multiple facets of influence: Identifying political influentials and opinion leaders on Twitter". In: *American behavioral scientist* 58.10, pp. 1260–1277 (cit. on p. 30).

Ekman, Paul et al. (1987). "Universals and cultural differences in the judgments of facial expressions of emotion." In: *Journal of personality and social psychology* 53.4, p. 712 (cit. on p. 73).

Escortell Pérez, M^{a}. Amparo, Maite Giménez Fayos, and Paolo Rosso (2017). "El impacto de las emociones en el análisis de la polaridad en textos con lenguaje figurado en Twitter". In: *Procesamiento del Lenguaje Natural. N. 58 (2017)* (cit. on p. 145).

Farias, DI Hernández and Paolo Rosso (2017). "Irony, sarcasm, and sentiment analysis". In: *Sentiment Analysis in Social Networks*. Elsevier, pp. 113–128 (cit. on pp. 71, 72).

Farías, Delia Irazú Hernández et al. (2015). "Valento: Sentiment analysis of figurative language tweets with irony and sarcasm". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 694–698 (cit. on p. 71).

Farnadi, Golnoosh et al. (2016). "Computational personality recognition in social media". In: *User modeling and user-adapted interaction* 26.2-3, pp. 109–142 (cit. on p. 29).

Fernandez Hernandez, A. and I. Segura Bedmar (2017). "Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: UC3M". In: *IberEval SEPLN* (cit. on pp. 45–47).

Fersini, Elisabetta, Paolo Rosso, and Maria Anzovino (2018). "Overview of the Task on Automatic Misogyny Identification at IberEval 2018." In: *IberEval at SEPLN* 2150, pp. 214–228 (cit. on p. 35).

Forman, George (2008). "BNS feature scaling: an improved representation over tf-idf for svm text classification". In: *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 263–270 (cit. on p. 10).

Gayo Avello, Daniel, Panagiotis T Metaxas, and Eni Mustafaraj (2011). "Limits of electoral predictions using twitter". In: *Proceedings of the fifth international AAAI conference on weblogs and social media.* Association for the Advancement of Artificial Intelligence (cit. on p. 31).

Gharavi, Erfaneh and Kayvan Bijari (2017). "Short Text Classification Using Deep Representation: A Case Study of Spanish Tweets in Coset Shared Task." In: *IberEval SEPLN*, pp. 28–35 (cit. on p. 46).

Ghosh, Aniruddha et al. (2015). "Semeval-2015 task 11: Sentiment analysis of figurative language in twitter". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 470–478 (cit. on pp. 56, 61).

Gill, Alastair J and Jon Oberlander (2002). “Taking care of the linguistic features of extraversion”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 24. 24 (cit. on p. 28).

Gill, Alastair James (2003). “Personality and language: The projection and perception of personality in computer-mediated communication”. PhD thesis. Citeseer (cit. on pp. 28, 81).

Gilpin, Leilani H et al. (2018). “Explaining Explanations: An Overview of Interpretability of Machine Learning”. In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, pp. 80–89 (cit. on pp. 36, 106).

Giménez, Maite, Tomás Baviera, et al. (2017). “Overview of the 1st classification of spanish election tweets task at ibereval 2017”. In: *Notebook Papers of 2nd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL), Murcia, Spain, September*. Vol. 19 (cit. on pp. 40, 146).

Giménez, Maite, Raül Fabra, et al. (Status: Under Review). “A fine-grained study of interpretability of Convolutional Neural Networks for text classification”. In: *Knowledge-Based Systems* (cit. on p. 146).

Giménez, Maite, Delia Irazú Hernández, and Ferran Pla (2015). “Segmenting Target Audiences: Automatic Author Profiling Using Tweets.” In: *Proceedings of CLEF* (cit. on pp. 110, 146).

Giménez, Maite, Jaume Jordán, et al. (2018). “Rassel: Robot Assistant for the Elderly”. In: *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, pp. 5–9 (cit. on p. 147).

Giménez, Maite, Javier Palanca, and Vicent Botti (2020). “Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis”. In: *Neurocomputing* 378, pp. 315–323 (cit. on p. 145).

Giménez, Maite and Roberto Paredes (2016). “PRHLT at PR-SOCO: A Regression Model for Predicting Personality Traits from Source Code.” In: *FIRE (Working Notes)*, pp. 38–42 (cit. on p. 146).

Giménez, Maite, Roberto Paredes, and Paolo Rosso (2017). "Personality Recognition Using Convolutional Neural Networks". In: *International Conference on Computational Linguistics and Intelligent Text Processing.* Springer, pp. 313–323 (cit. on p. 145).

Giménez, Maite, Ferran Pla, and Lluís-F Hurtado (2015). "Elirf: a SVM approach for SA tasks in twitter at SemEval-2015". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 574–581 (cit. on p. 146).

González, José-Ángel, Ferran Pla, and Lluís-Felip Hurtado (2017). "ELiRF-UPV at IberEval 2017: Classification Of Spanish Election Tweets (COSET)." In: *IberEval SEPLN*, pp. 55–60 (cit. on pp. 45, 46, 48, 50, 53).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press (cit. on p. 15).

Hammar, Kim et al. (2018). "Deep text mining of instagram data without strong supervision". In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI).* IEEE, pp. 158–165 (cit. on p. 26).

Hansen, Lars Kai et al. (2011). "Good friends, bad news-affect and virality in twitter". In: *Future information technology.* Springer, pp. 34–43 (cit. on pp. 63, 93).

Harris, Anne and Stacy Holman Jones (2016). "Words". In: *Writing for Performance.* Brill Sense, pp. 19–35 (cit. on p. 8).

Harris, Zellig S (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162 (cit. on p. 10).

Hassan, Abdalraouf and Ausif Mahmood (2017). "Deep learning approach for sentiment analysis of short texts". In: *2017 3rd international conference on control, automation and robotics (ICCAR).* IEEE, pp. 705–710 (cit. on p. 16).

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (cit. on p. 15).

Hillard, Dustin, Stephen Purpura, and John Wilkerson (2008). “Computer-assisted topic classification for mixed-methods social science research”. In: *Journal of Information Technology & Politics* 4.4, pp. 31–46 (cit. on p. 30).

Ho, Tin Kam (1995). “Random decision forests”. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282 (cit. on p. 13).

Hu, Minqing and Bing Liu (2004). “Mining and summarizing customer reviews”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 168–177 (cit. on pp. 63, 93).

Huang, Cheng-Hui, Jian Yin, and Fang Hou (2011). “A text similarity measurement combining word semantic information with TF-IDF method”. In: *Jisuanji Xuebao(Chinese Journal of Computers)* 34.5, pp. 856–864 (cit. on p. 10).

Hurtado, Lluís F and Ferran Pla (2014). “ELiRF-UPV en TASS 2014: Análisis de sentimientos, detección de tópicos y análisis de sentimientos de aspectos en twitter”. In: *Procesamiento del Lenguaje Natural*, pp. 1–7 (cit. on p. 62).

Hussein, Doaa Mohey El-Din Mohamed (2018). “A survey on sentiment analysis challenges”. In: *Journal of King Saud University-Engineering Sciences* 30.4, pp. 330–338 (cit. on p. 24).

Hutson, Matthew (2018). *Artificial intelligence faces reproducibility crisis* (cit. on p. 31).

Jacovi, Alon, Oren Sar Shalom, and Yoav Goldberg (2018). “Understanding Convolutional Neural Networks for Text Classification”. In: *arXiv preprint arXiv:1809.08037* (cit. on pp. 37, 106, 123, 125).

Jansen, Bernard J et al. (2009). “Twitter power: Tweets as electronic word of mouth”. In: *Journal of the American society for information science and technology* 60.11, pp. 2169–2188 (cit. on p. 23).

Juárez, G. and A. Peralta (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: Electa”. In: *IberEval SEPLN* (cit. on pp. 45, 46).

Jurgens, David, Mohammad Taher Pilehvar, and Roberto Navigli (2014). "SemEval-2014 Task 3: Cross-Level Semantic Similarity." In: *SemEval at COLING*, pp. 17–26 (cit. on p. 33).

Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom (2014). "A convolutional neural network for modelling sentences". In: *arXiv preprint arXiv:1404.2188* (cit. on p. 26).

Kao, Anne and Steve R Poteet (2007). *Natural language processing and text mining*. Springer Science & Business Media (cit. on p. 29).

Keogh, Eamonn and Abdullah Mueen (2011). "Encyclopedia of machine learning". In: *Curse of Dimensionality. Springer: Boston*, pp. 257–258 (cit. on p. 83).

Khandelwal, Ankush et al. (2017). "Classification Of Spanish Election Tweets (COSET) 2017: Classifying Tweets Using Character and Word Level Features." In: *IberEval SEPLN*, pp. 49–54 (cit. on pp. 45, 46, 50).

Kim, Yoon (2014). "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882* (cit. on pp. 16, 26, 97, 98, 111, 116, 129).

Krebs, Florian et al. (2017). "Social emotion mining techniques for Facebook posts reaction prediction". In: *arXiv preprint arXiv:1712.03249* (cit. on p. 26).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105 (cit. on p. 15).

Lafuente, Carlos Villar and Gonçal Garcés Díaz-Munío (2017). "Several Approaches for Tweet Topic Classification in COSET-IberEval 2017." In: *IberEval SEPLN*, pp. 36–42 (cit. on pp. 46, 47, 50).

Laleh, Asadzadeh and Rahimi Shahram (2017). "Analyzing Facebook activities for personality recognition". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 960–964 (cit. on p. 28).

Landecker, Will et al. (2013). "Interpreting individual classifications of hierarchical networks". In: *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, pp. 32–38 (cit. on p. 36).

Larsson, Anders Olof and Hallvard Moe (2012). “Studying political microblogging: Twitter users in the 2010 Swedish election campaign”. In: *New media & society* 14.5, pp. 729–747 (cit. on p. 31).

LeCun, Y. (1989). “Generalization and Network Design Strategies”. In: *Connectionism in Perspective*. Ed. by R. Pfeifer et al. Zurich, Switzerland: Elsevier (cit. on p. 15).

Lee, Kathy et al. (2011). “Twitter trending topic classification”. In: *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, pp. 251–258 (cit. on p. 31).

Lefever, Els and Veronique Hoste (2010). “Semeval-2010 task 3: Cross-lingual word sense disambiguation”. In: *5th International workshop on Semantic Evaluation (SemEval 2010)*. Association for Computational Linguistics (ACL), pp. 15–20 (cit. on p. 33).

Legendre, Adrien Marie (1805). *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot (cit. on p. 12).

Li, Pengfei and Kezhi Mao (2019). “Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts”. In: *Expert Systems with Applications* 115, pp. 512–523 (cit. on p. 16).

Li, Yang and Tao Yang (2018). “Word embedding for understanding natural language: a survey”. In: *Guide to Big Data Applications*. Springer, pp. 83–104 (cit. on p. 11).

Lichtenwalter, D. and T. Olősan (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: LichtenwalterOlsan”. In: *IberEval SEPLN* (cit. on p. 46).

Lin, Min, Qiang Chen, and Shuicheng Yan (2013). “Network in network”. In: *arXiv preprint arXiv:1312.4400* (cit. on p. 122).

Linzen, Tal Tal, Grzegorz Chrupała, and Afra Alishahi (2018). “Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (cit. on p. 106).

Litvinova, Tatiana et al. (2017). “Overview of the RUSProfiling PAN at FIRE Track on Cross-genre Gender Identification in Russian.” In: *FIRE (Working Notes)*, pp. 1–7 (cit. on p. 34).

Liu, Bing (2012). “Sentiment analysis and opinion mining”. In: *Synthesis lectures on human language technologies* 5.1, pp. 1–167 (cit. on pp. 22, 23).

Liu, Bing, Minqing Hu, and Junsheng Cheng (2005). “Opinion observer: analyzing and comparing opinions on the web”. In: *Proceedings of the 14th international conference on World Wide Web*. ACM, pp. 342–351 (cit. on p. 24).

Liu, Hui, Qingyu Yin, and William Yang Wang (2018). “Towards explainable NLP: A generative explanation framework for text classification”. In: *arXiv preprint arXiv:1811.00196* (cit. on p. 36).

López García, Guillermo and Lidia Valera Ordaz (2017). “Pantallas electorales: el discurso de partidos, medios y ciudadanos en la campaña de 2015”. In: *Pantallas electorales*, pp. 1–205 (cit. on p. 40).

Maas, Andrew L et al. (2011). “Learning word vectors for sentiment analysis”. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, pp. 142–150 (cit. on p. 128).

Mahendran, Aravindh and Andrea Vedaldi (2015). “Understanding deep image representations by inverting them”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196 (cit. on p. 36).

Mahiques Sifres, X. and V. Lyeuta Tykhovod (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: slovak”. In: *IberEval SEPLN* (cit. on pp. 45–47).

Mairesse, François et al. (2007). “Using linguistic cues for the automatic recognition of personality in conversation and text”. In: *Journal of artificial intelligence research* 30, pp. 457–500 (cit. on p. 28).

Maluenda Maez, F. and G.A. Garcìa Ferrando (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: Citripio”. In: *IberEval SEPLN* (cit. on p. 46).

Mandl, Thomas et al. (2019). “Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages”. In: *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pp. 14–17 (cit. on p. 34).

Manning, Christopher D, Christopher D Manning, and Hinrich Schütze (1999). *Foundations of statistical natural language processing.* MIT press (cit. on p. 59).

Martín Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org (cit. on pp. 112, 129).

Maynard, DG and Mark A Greenwood (2014). “Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis”. In: *LREC 2014 Proceedings.* ELRA (cit. on p. 25).

Mazzoleni, Gianpietro (2014). *La comunicación política.* Alianza Editorial (cit. on p. 30).

Medhat, Walaa, Ahmed Hassan, and Hoda Korashy (2014). “Sentiment analysis algorithms and applications: A survey”. In: *Ain Shams Engineering Journal* 5.4, pp. 1093–1113 (cit. on p. 112).

Mehta, Yash et al. (2019). “Recent trends in deep learning based personality detection”. In: *Artificial Intelligence Review*, pp. 1–27 (cit. on p. 29).

Menini, Stefano et al. (2017). “Topic-based agreement and disagreement in us electoral manifestos”. In: *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2938–2944 (cit. on p. 31).

Mìguez, O. and M. Valdiviezo (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: MiVal”. In: *IberEval SEPLN* (cit. on p. 46).

Mihaylova, Tsvetomila et al. (2019). “SemEval-2019 task 8: Fact checking in community question answering forums”. In: *arXiv preprint arXiv:1906.01727* (cit. on p. 33).

Mikolov, Tomas, Kai Chen, et al. (2013). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (cit. on pp. 11, 115, 130).

Mikolov, Tomas, Ilya Sutskever, et al. (2013). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119 (cit. on p. 11).

Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41 (cit. on p. 64).

Minsky, Marvin and Seymour Papert (1969). “An introduction to computational geometry”. In: *Cambridge tiass., HIT* (cit. on p. 12).

Mlinarić, Ana, Martina Horvat, and Vesna Šupak Smolčić (2017). “Dealing with the positive publication bias: Why you should really publish your negative results”. In: *Biochemia medica: Biochemia medica* 27.3, pp. 447–452 (cit. on p. 32).

Mnih, Andriy and Geoffrey Hinton (2007). “Three new graphical models for statistical language modelling”. In: *Proceedings of the 24th international conference on Machine learning*, pp. 641–648 (cit. on p. 10).

Mohammad, Saif M and Svetlana Kiritchenko (2015). “Using hashtags to capture fine emotion categories from tweets”. In: *Computational Intelligence* 31.2, pp. 301–326 (cit. on p. 93).

Mohammad, Saif M, Svetlana Kiritchenko, and Xiaodan Zhu (2013). “NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets”. In: *arXiv preprint arXiv:1308.6242* (cit. on pp. 93, 110).

Mohammad, Saif M and Peter D Turney (2010). “Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon”. In: *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, pp. 26–34 (cit. on p. 73).

Mohammad, Saif M. and Peter D. Turney (2013). “Crowdsourcing a Word-Emotion Association Lexicon”. In: 29.3, pp. 436–465 (cit. on p. 64).

Montoyo, Andrés, Patricio Martínez-Barco, and Alexandra Balahur (2012). *Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments* (cit. on p. 25).

Morin, Frederic and Yoshua Bengio (2005). "Hierarchical probabilistic neural network language model." In: *Aistats*. Vol. 5. Citeseer, pp. 246–252 (cit. on p. 10).

Moro, Andrea and Roberto Navigli (2015). "Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking". In: *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 288–297 (cit. on p. 33).

Mukherjee, Swati and Updesh Kumar (2016). "Ethical issues in personality assessment". In: *The Wiley Handbook of Personality Assessment*, pp. 415–426 (cit. on p. 148).

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *ICML* (cit. on p. 99).

Nakov, Preslav, Doris Hoogeveen, et al. (2017). "SemEval-2017 task 3: Community question answering". In: *arXiv preprint arXiv:1912.00730* (cit. on p. 33).

Nakov, Preslav, Lluís Màrquez, et al. (2015). "Semeval-2015 task 3: Answer selection in community question answering". In: *arXiv preprint arXiv:1911.11403* (cit. on p. 33).

Nakov, Preslav, Sara Rosenthal, et al. (2013). "Semeval-2013 task 2: Sentiment analysis in Twitter". In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 312–320 (cit. on p. 57).

Navigli, Roberto, David Jurgens, and Daniele Vannella (2013). "Semeval-2013 task 12: Multilingual word sense disambiguation". In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 222–231 (cit. on p. 33).

Norman, Warren T (1963). "Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings."

In: *The Journal of Abnormal and Social Psychology* 66.6, p. 574 (cit. on pp. 81, 82).

Nosek, Brian A, Jeffrey R Spies, and Matt Motyl (2012). "Scientific utopia: II. Restructuring incentives and practices to promote truth over publishability". In: *Perspectives on Psychological Science* 7.6, pp. 615–631 (cit. on p. 32).

O'Connor, Brendan, Ramnath Balasubramanyan, et al. (2010). "From tweets to polls: Linking text sentiment to public opinion time series". In: *Fourth International AAAI Conference on Weblogs and Social Media* (cit. on p. 23).

O'Connor, Brendan, Michel Krieger, and David Ahn (2010). "Tweetmotif: Exploratory search and topic summarization for twitter". In: *Fourth International AAAI Conference on Weblogs and Social Media* (cit. on p. 23).

Oberlander, Jon and Scott Nowson (2006). "Whose thumb is it anyway? Classifying author personality from weblog text". In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 627–634 (cit. on p. 28).

Ortiz-Ospina, Esteban (n.d.). *The rise of social media*. `https://ourworldindata.org/rise-of-social-media`. Accessed: 17-10-2020 (cit. on p. 1).

Pabón, Oscar Hernán Paruma et al. (2016). "Finding relationships between sociotechnical aspects and personality traits by mining developer e-mails". In: *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, pp. 8–14 (cit. on p. 79).

Pal, Kuntal Kumar and KS Sudeep (2016). "Preprocessing for image classification by convolutional neural networks". In: *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, pp. 1778–1781 (cit. on p. 107).

Pang, Bo, Lillian Lee, et al. (2008). "Opinion mining and sentiment analysis". In: *Foundations and Trends® in Information Retrieval* 2.1–2, pp. 1–135 (cit. on p. 23).

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-*

Volume 10. Association for Computational Linguistics, pp. 79–86 (cit. on pp. 23, 62).

Patra, Braja Gopal et al. (2015). "Shared task on sentiment analysis in indian languages (sail) tweets-an overview". In: *International Conference on Mining Intelligence and Knowledge Exploration*. Springer, pp. 650–655 (cit. on p. 26).

Patterson, Thomas E (1980). *The mass media election: How Americans choose their president*. Greenwood (cit. on p. 30).

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830 (cit. on pp. 10, 48, 66, 72, 74, 86, 92).

Pennebaker, James W, Martha E Francis, and Roger J Booth (2001). "Linguistic inquiry and word count: LIWC 2001". In: *Mahway: Lawrence Erlbaum Associates* 71.2001, p. 2001 (cit. on p. 73).

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543 (cit. on pp. 11, 97).

Perez-Rosas, Veronica, Carmen Banea, and Rada Mihalcea (2012). "Learning Sentiment Lexicons in Spanish." In: *LREC*. Vol. 12, p. 73 (cit. on p. 24).

Piryonesi, S Madeh and Tamer E El-Diraby (2020). "Data analytics in asset management: Cost-effective prediction of the pavement condition index". In: *Journal of Infrastructure Systems* 26.1, p. 04019036 (cit. on p. 13).

Pla, Ferran and Lluís-F Hurtado (2014). "Political tendency identification in twitter using sentiment analysis techniques". In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*, pp. 183–192 (cit. on p. 62).

Pla, Ferran and Lluıs-F Hurtado (2013). "ELiRF-UPV en TASS-2013: Análisis de sentimientos en Twitter". In: *XXIX Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN 2013). TASS*, pp. 220–227 (cit. on p. 62).

Plank, Barbara and Dirk Hovy (2015). “Personality traits on twitter-or-how to get 1.500 personality tests in a week”. In: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 92–98 (cit. on p. 28).

Plesser, Hans E (2018). “Reproducibility vs. replicability: a brief history of a confused terminology”. In: *Frontiers in neuroinformatics* 11, p. 76 (cit. on p. 31).

Pool, Chris and Malvina Nissim (2016). “Distant supervision for emotion detection using Facebook reactions”. In: *arXiv preprint arXiv:1611.02988* (cit. on p. 26).

Potthast, Martin et al. (2019). “A decade of shared tasks in digital text forensics at PAN”. In: *European Conference on Information Retrieval*. Springer, pp. 291–300 (cit. on p. 34).

Puigcerver, J. (2017). “Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: Puigcerver”. In: *IberEval SEPLN* (cit. on pp. 46, 47).

Quercia, Daniele, Harry Askham, and Jon Crowcroft (2012). “Tweetlda: supervised topic classification and link prediction in twitter”. In: *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 247–250 (cit. on p. 31).

Quercia, Daniele, Michal Kosinski, et al. (2011). “Our twitter profiles, our selves: Predicting personality with twitter”. In: *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, pp. 180–185 (cit. on p. 28).

Quinlau, R (1986). “Induction of decision trees”. In: *Machine learning* 1.1, S1–S106 (cit. on p. 13).

Rangel, Francisco, Fabio González, et al. (2016). “Pan at fire: overview of the pr-soco track on personality recognition in source code”. In: *Forum for Information Retrieval Evaluation*. Springer, pp. 1–19 (cit. on pp. 28, 79, 82).

Rangel, Francisco, Paolo Rosso, et al. (2015). “Overview of the 3rd Author Profiling Task at PAN 2015”. In: *CLEF*. sn, p. 2015 (cit. on pp. 80, 86, 90, 95).

Remus, Robert, Uwe Quasthoff, and Gerhard Heyer (2010). “SentiWS-A Publicly Available German-language Resource for Sentiment Analysis.” In: *LREC*. Citeseer (cit. on p. 24).

Reyes, Antonio, Paolo Rosso, and Davide Buscaldi (2012). “From humor recognition to irony detection: The figurative language of social media”. In: *Data & Knowledge Engineering* 74, pp. 1–12 (cit. on p. 24).

Reyes, Antonio, Paolo Rosso, and Tony Veale (2013). “A multidimensional approach for detecting irony in twitter”. In: *Language resources and evaluation* 47.1, pp. 239–268 (cit. on p. 24).

Robertson, Stephen (2004). “Understanding inverse document frequency: on theoretical arguments for IDF”. In: *Journal of documentation* (cit. on p. 10).

Roelleke, Thomas and Jun Wang (2008). “Tf-idf uncovered: a study of theories and probabilities”. In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 435–442 (cit. on p. 10).

Rosenberg, Daniel (2014). “Stop, Words”. In: *Representations* 127.1, pp. 83–92 (cit. on p. 58).

Rosenblatt, Frank (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory (cit. on p. 12).

Rosenthal, Sara et al. (2015). “Semeval-2015 task 10: Sentiment analysis in twitter”. In: *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 451–463 (cit. on pp. 55, 57).

Rosso, Paolo et al. (2016). “Overview of PAN’16”. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, pp. 332–350 (cit. on pp. 34, 82).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). “Learning representations by back-propagating errors”. In: *nature* 323.6088, pp. 533–536 (cit. on pp. 12, 16).

Samek, Wojciech, Thomas Wiegand, and Klaus-Robert Müller (2017). "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models". In: *arXiv preprint arXiv:1708.08296* (cit. on p. 36).

Sánchez, I. (2017). "Submission to the 1st classification of spanish election tweets task at ibereval 2017. Team: ivsanro1". In: *IberEval SEPLN* (cit. on pp. 45–47).

Schütze, Hinrich, Christopher D Manning, and Prabhakar Raghavan (2008). "Introduction to information retrieval". In: *Proceedings of the international communication of association for computing machinery conference*, p. 260 (cit. on p. 58).

Schwartz, H Andrew et al. (2013). "Personality, gender, and age in the language of social media: The open-vocabulary approach". In: *PloS one* 8.9, e73791 (cit. on p. 28).

Schwartz, Richard et al. (1997). "A maximum likelihood model for topic classification of broadcast news". In: *Fifth European Conference on Speech Communication and Technology* (cit. on p. 30).

Shen, Yelong et al. (2014). "Learning semantic representations using convolutional neural networks for web search". In: *Proceedings of the companion publication of the 23rd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 373–374 (cit. on p. 16).

Shirani-Mehr, Houshmand (2014). "Applications of deep learning to sentiment analysis of movie reviews". In: *Technical report*. Stanford University (cit. on p. 27).

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (cit. on p. 36).

Smedt, Tom De and Walter Daelemans (2012). "Pattern for python". In: *Journal of Machine Learning Research* 13.Jun, pp. 2063–2067 (cit. on p. 63).

Snidaro, Lauro, Giovanni Ferrin, and Gian Luca Foresti (2019). "Distributional memory explainable word embeddings in continuous space". In: *2019 22th*

International Conference on Information Fusion (FUSION). IEEE, pp. 1–7 (cit. on p. 36).

Socher, Richard et al. (2013). “Recursive deep models for semantic compositionality over a sentiment treebank”. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642 (cit. on p. 113).

Spark-Jones, K (1975). “Report on the need for and provision of an’ideal’information retrieval test collection”. In: *Computer Laboratory* (cit. on p. 51).

Sriram, Bharath et al. (2010). “Short text classification in twitter to improve information filtering”. In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 841–842 (cit. on p. 31).

Štajner, Sanja and Seren Yenikent (2020). “A Survey of Automatic Personality Detection from Texts”. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6284–6295 (cit. on pp. 27, 29).

Sterling, Theodore D, Wilf L Rosenbaum, and James J Weinkam (1995). “Publication decisions revisited: The effect of the outcome of statistical tests on the decision to publish and vice versa”. In: *The American Statistician* 49.1, pp. 108–112 (cit. on p. 32).

Strapparava, Carlo and Rada Mihalcea (2007). “Semeval-2007 task 14: Affective text”. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 70–74 (cit. on p. 25).

Sudhahar, Saatviga, Giuseppe A Veltri, and Nello Cristianini (2015). “Automated analysis of the US presidential elections using Big Data and network analysis”. In: *Big Data & Society* 2.1, p. 2053951715572916 (cit. on p. 31).

Sulis, Emilio et al. (2016). “Figurative messages and affect in Twitter: Differences between# irony,# sarcasm and# not”. In: *Knowledge-Based Systems* 108, pp. 132–143 (cit. on pp. 24, 72, 75).

Sun, Chi, Luyao Huang, and Xipeng Qiu (2019). “Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence”. In: *arXiv preprint arXiv:1903.09588* (cit. on p. 27).

Suttles, Jared and Nancy Ide (2013). "Distant supervision for emotion classification with discrete binary values". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 121–136 (cit. on p. 73).

Taboada, Maite et al. (2011). "Lexicon-based methods for sentiment analysis". In: *Computational linguistics* 37.2, pp. 267–307 (cit. on pp. 31, 63).

Tang, Duyu et al. (2015). "Effective LSTMs for target-dependent sentiment classification". In: *arXiv preprint arXiv:1512.01100* (cit. on p. 27).

Tarwani, Kanchan M and Swathi Edem (2017). "Survey on recurrent neural network in natural language processing". In: *Int. J. Eng. Trends Technol* 48, pp. 301–304 (cit. on p. 16).

Taulé, Mariona et al. (2017). "Overview of the task on stance and gender detection in tweets on Catalan independence at IberEval 2017". In: *2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017*. Vol. 1881. CEUR-WS, pp. 157–177 (cit. on p. 35).

Tumasjan, Andranik et al. (2010). "Predicting elections with twitter: What 140 characters reveal about political sentiment". In: *Fourth international AAAI conference on weblogs and social media*. Citeseer (cit. on p. 31).

Turney, Peter D (2002). "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 417–424 (cit. on p. 23).

Twitter Investor Relations (n.d.). `https://investor.twitterinc.com/home/default.aspx`. Accessed: 23-08-2020 (cit. on p. 80).

Vashishth, Shikhar et al. (2019). "Attention interpretability across nlp tasks". In: *arXiv preprint arXiv:1909.11218* (cit. on p. 37).

Veale, Tony and Yanfen Hao (2007). "Learning to understand figurative language: from similes to metaphors to irony". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 29. 29 (cit. on pp. 24, 25).

Villena Román, Julio et al. (2013). "Tass-workshop on sentiment analysis at sepln". In: (cit. on p. 26).

Villena-Román, Julio et al. (2014). "Tass2014-workshop on sentiment analysis at sepln-overview". In: *Proceedings of the TASS workshop at SEPLN* (cit. on p. 26).

Vinciarelli, Alessandro and Gelareh Mohammadi (2014). "A survey of personality computing". In: *IEEE Transactions on Affective Computing* 5.3, pp. 273–291 (cit. on p. 27).

Vinodhini, G and RM Chandrasekaran (2012). "Sentiment analysis and opinion mining: a survey". In: *International Journal* 2.6, pp. 282–292 (cit. on p. 24).

Waltinger, Ulli (2010). "GermanPolarityClues: A Lexical Resource for German Sentiment Analysis." In: *LREC*, pp. 1638–1642 (cit. on p. 24).

Wang, Hao et al. (2012). "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle". In: *Proceedings of the ACL 2012 system demonstrations*, pp. 115–120 (cit. on p. 31).

Wang, Jin et al. (2016). "Dimensional sentiment analysis using a regional CNN-LSTM model". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 225–230 (cit. on p. 16).

Wang, Xin et al. (2015). "Predicting polarities of tweets by composing word embeddings with 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)*. Vol. 1, pp. 1343–1353 (cit. on p. 27).

Wang, Xingyou, Weijie Jiang, and Zhiyong Luo (2016). "Combination of convolutional and recurrent neural network for sentiment analysis of short texts". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2428–2437 (cit. on p. 16).

Wang, Yequan, Minlie Huang, Li Zhao, et al. (2016). "Attention-based LSTM for aspect-level sentiment classification". In: *Proceedings of the 2016 conference* *on empirical methods in natural language processing*, pp. 606–615 (cit. on p. 27).

Weller, Katrin et al. (2014). *Twitter and society.* Vol. 89. Peter Lang (cit. on p. 23).

Wiegand, Michael et al. (2010). “A survey on the role of negation in sentiment analysis”. In: *Proceedings of the workshop on negation and speculation in natural language processing*, pp. 60–68 (cit. on p. 25).

Wikipedia contributors (2020a). *Accuracy and Precision — Wikipedia, The Free Encyclopedia.* [Online; accessed 08-10-2020] (cit. on p. 17).

Wikipedia contributors (2020b). *F1 score — Wikipedia, The Free Encyclopedia.* [Online; accessed 08-10-2020] (cit. on p. 18).

Wikipedia contributors (2020c). *Mean Squared Error — Wikipedia, The Free Encyclopedia.* [Online; accessed 05-10-2020] (cit. on p. 16).

Wikipedia contributors (2020d). *Pearson Correlation Coefficient — Wikipedia, The Free Encyclopedia.* [Online; accessed 05-10-2020] (cit. on p. 17).

Wikipedia contributors (2020e). *Regression Analysis — Wikipedia, The Free Encyclopedia.* [Online; accessed 27-09-2020] (cit. on p. 12).

Wikipedia contributors (2020f). *SemEval — Wikipedia, The Free Encyclopedia.* [Online; accessed 08-August-2020] (cit. on p. 33).

Wilson, Theresa et al. (2005). “OpinionFinder: A system for subjectivity analysis”. In: *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pp. 34–35 (cit. on p. 24).

Wu, Sun and Udi Manber (1992). “Fast text searching: allowing errors”. In: *Communications of the ACM* 35.10, pp. 83–91 (cit. on p. 51).

Xu, Wei, Chris Callison-Burch, and Bill Dolan (2015). “Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit)”. In: *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 1–11 (cit. on p. 33).

Xue, Di et al. (2018). “Deep learning-based personality recognition from text posts of online social networks”. In: *Applied Intelligence* 48.11, pp. 4232–4246 (cit. on p. 29).

Yadav, Ashima and Dinesh Kumar Vishwakarma (2020). “Sentiment analysis using deep learning architectures: a review”. In: *Artificial Intelligence Review* 53.6, pp. 4335–4385 (cit. on p. 148).

Yin, Wenpeng et al. (2017). “Comparative study of CNN and RNN for natural language processing”. In: *arXiv preprint arXiv:1702.01923* (cit. on pp. 16, 27, 107, 112, 128).

Youyou, Wu, Michal Kosinski, and David Stillwell (2015). “Computer-based personality judgments are more accurate than those made by humans”. In: *Proceedings of the National Academy of Sciences* 112.4, pp. 1036–1040 (cit. on p. 29).

Yu, Jianguo and Konstantin Markov (2017). “Deep learning based personality recognition from facebook status updates”. In: *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, pp. 383–387 (cit. on pp. 28, 29).

Yun-tao, Zhang, Gong Ling, and Wang Yong-cheng (2005). “An improved TF-IDF approach for text classification”. In: *Journal of Zhejiang University-Science A* 6.1, pp. 49–55 (cit. on p. 10).

Zeiler, Matthew D and Rob Fergus (2014). “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer, pp. 818–833 (cit. on p. 36).

Zhang, Lei, Shuai Wang, and Bing Liu (2018). “Deep learning for sentiment analysis: A survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4, e1253 (cit. on pp. 16, 26, 27).

Zhang, Quan-shi and Song-Chun Zhu (2018). “Visual interpretability for deep learning: a survey”. In: *Frontiers of Information Technology & Electronic Engineering* 19.1, pp. 27–39 (cit. on p. 121).

Zhang, Wei (1988). “Shift-invariant pattern recognition neural network and its optical architecture”. In: *Proceedings of annual conference of the Japan Society of Applied Physics* (cit. on p. 107).

Zhang, Wei et al. (1990). “Parallel distributed processing model with local space-invariant interconnections and its optical architecture”. In: *Applied optics* 29.32, pp. 4790–4797 (cit. on p. 107).

Zhang, Wei Emma et al. (2020). “Adversarial attacks on deep-learning models in natural language processing: A survey”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.3, pp. 1–41 (cit. on p. 34).

Zhang, Ye and Byron Wallace (2015). “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification”. In: *arXiv preprint arXiv:1510.03820* (cit. on pp. 16, 98).

Zhou, Bolei et al. (2014). “Object detectors emerge in deep scene cnns”. In: *arXiv preprint arXiv:1412.6856* (cit. on p. 36).

Zhou, Bolei et al. (2016). “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929 (cit. on p. 122).

Zhou, Chunting et al. (2015). “A C-LSTM neural network for text classification”. In: *arXiv preprint arXiv:1511.08630* (cit. on p. 27).

Zhu, Xiaodan, Svetlana Kiritchenko, and Saif Mohammad (2014). “Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets”. In: *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 443–447 (cit. on p. 23).

Zirn, Cäcilia et al. (2016). “Classifying topics and detecting topic shifts in political manifestos”. In: (cit. on p. 31).