Attention-based approaches for Text Analytics in Social Media and Automatic Summarization

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

José Ángel González Barba

Department of Computer Systems and Computation
Valencian Research Institute for Artificial Intelligence

This dissertation is submitted for the PhD in

Computer Science

Supervised by
Dr. Lluís Felip Hurtado Oliver
Dr. Emilio Sanchis Arnal

Valencia July 2021

A mi familia, especialmente a mis padres y a Ana, por haberme aguantado incondicionalmente todos estos años. A mis directores, Lluís y Emilio y a todos los miembros del grupo que me han acogido y ayudado a crecer durante esta andadura: Ferran, Encarna, Fernando, Salva y María José.

A los que estabais y estaréis, gracias a todos.

This thesis has been developed under a FPI scholarship (PAID-01-17) granted by the Universitat Politècnica de València, and in the framework of the projects ASLP-MULAN: Audio, Speech and Language Processing for MULtimedia ANalytics (TIN2014-54288-C4-3-R) and AMIC: Affective Multimedia analytics with Inclusive and natural Communication (TIN2017-85854-C4-2-R) funded with MINECO and FEDER funds.

Abstract

Nowadays, society has access, and the possibility to contribute, to large amounts of the content present on the internet, such as social networks, online newspapers, forums, blogs, or multimedia content platforms. These platforms have had, during the last years, an overwhelming impact on the daily life of individuals and organizations, becoming the predominant ways for sharing, discussing, and analyzing online content. Therefore, it is very interesting to work with these platforms, from different points of view, under the umbrella of Natural Language Processing. In this thesis, we focus on two broad areas inside this field, applied to analyze online content: text analytics in social media and automatic summarization. Neural networks are also a central topic in this thesis, where all the experimentation has been performed by using deep learning approaches, mainly based on attention mechanisms. Besides, we mostly work with the Spanish language, due to it is an interesting and underexplored language with a great interest in the research projects we participated in.

On the one hand, for text analytics in social media, we focused on affective analysis tasks, including sentiment analysis and emotion detection, along with the analysis of the irony. In this regard, an approach based on Transformer Encoders, based on contextualizing pretrained Spanish word embeddings from Twitter, to address sentiment analysis and irony detection tasks, is presented. We also propose the use of evaluation metrics as loss functions, in order to train neural networks for reducing the impact of the class imbalance in multi-class and multi-label emotion detection tasks. Additionally, a specialization of BERT both for the Spanish language and the Twitter domain, that takes into account inter-sentence coherence in Twitter conversation flows, is presented. The performance of all these approaches has been tested with different corpora, from several reference evaluation benchmarks, showing very competitive results in all the tasks addressed.

On the other hand, we focused on extractive summarization of news articles and TV talk shows. Regarding the summarization of news articles, a theoretical framework for extractive summarization, based on siamese hierarchical networks with attention mechanisms, is presented. Also, we present two instantiations of this framework: Siamese Hierarchical Attention Networks and Siamese Hierarchical Transformer Encoders. These systems were

evaluated on the CNN/DailyMail and the NewsRoom corpora, obtaining competitive results in comparison to other contemporary extractive approaches. Concerning the TV talk shows, we proposed a text summarization task, for summarizing the transcribed interventions of the speakers, about a given topic, in the Spanish TV talk shows of the “La Noche en 24 Horas" program. In addition, a corpus of news articles, collected from several Spanish online newspapers, is proposed, in order to study the domain transferability of siamese hierarchical approaches, between news articles and interventions of debate participants. This approach shows better results than other extractive techniques, along with a very promising domain transferability.

Resumen

Hoy en día, la sociedad tiene acceso y posibilidad de contribuir a grandes cantidades de contenidos presentes en Internet, como redes sociales, periódicos online, foros, blogs o plataformas de contenido multimedia. Todo este tipo de medios han tenido, durante los últimos años, un impacto abrumador en el día a día de individuos y organizaciones, siendo actualmente medios predominantes para compartir, debatir y analizar contenidos online. Por este motivo, resulta de interés trabajar sobre este tipo de plataformas, desde diferentes puntos de vista, bajo el paraguas del Procesamiento del Lenguaje Natural. En esta tesis nos centramos en dos áreas amplias dentro de este campo, aplicadas al análisis de contenido en línea: análisis de texto en redes sociales y resumen automático. En paralelo, las redes neuronales también son un tema central de esta tesis, donde toda la experimentación se ha realizado utilizando enfoques de aprendizaje profundo, principalmente basados en mecanismos de atención. Además, trabajamos mayoritariamente con el idioma español, por ser un idioma poco explorado y de gran interés para los proyectos de investigación en los que participamos.

Por un lado, para el análisis de texto en redes sociales, nos enfocamos en tareas de análisis afectivo, incluyendo análisis de sentimientos y detección de emociones, junto con el análisis de la ironía. En este sentido, se presenta un enfoque basado en Transformer Encoders, que consiste en contextualizar *word embeddings* pre-entrenados con tweets en español, para abordar tareas de análisis de sentimiento y detección de ironía. También proponemos el uso de métricas de evaluación como funciones de pérdida, con el fin de entrenar redes neuronales, para reducir el impacto del desequilibrio de clases en tareas *multi-class* y *multi-label* de detección de emociones. Adicionalmente, se presenta una especialización de BERT tanto para el idioma español como para el dominio de Twitter, que tiene en cuenta la coherencia entre tweets en conversaciones de Twitter. El desempeño de todos estos enfoques ha sido probado con diferentes corpus, a partir de varios *benchmarks* de referencia, mostrando resultados muy competitivos en todas las tareas abordadas.

Por otro lado, nos centramos en el resumen extractivo de artículos periodísticos y de programas televisivos de debate. Con respecto al resumen de artículos, se presenta un marco teórico para el resumen extractivo, basado en redes jerárquicas siamesas con mecanismos de atención. También presentamos dos instancias de este marco: *Siamese Hierarchical*

Attention Networks y *Siamese Hierarchical Transformer Encoders*. Estos sistemas han sido evaluados en los corpora CNN/DailyMail y NewsRoom, obteniendo resultados competitivos en comparación con otros enfoques extractivos coetáneos. Con respecto a los programas de debate, se ha propuesto una tarea que consiste en resumir las intervenciones transcritas de los ponentes, sobre un tema determinado, en el programa "La Noche en 24 Horas". Además, se propone un corpus de artículos periodísticos, recogidos de varios periódicos españoles en línea, con el fin de estudiar la transferibilidad de los enfoques propuestos, entre artículos e intervenciones de los participantes en los debates. Este enfoque muestra mejores resultados que otras técnicas extractivas, junto con una transferibilidad de dominio muy prometedora.

Resum

Avui en dia, la societat té accés i possibilitat de contribuir a grans quantitats de continguts presents a Internet, com xarxes socials, diaris online, fòrums, blocs o plataformes de contingut multimèdia. Tot aquest tipus de mitjans han tingut, durant els darrers anys, un impacte aclaparador en el dia a dia d'individus i organitzacions, sent actualment mitjans predominants per compartir, debatre i analitzar continguts en línia. Per aquest motiu, resulta d'interès treballar sobre aquest tipus de plataformes, des de diferents punts de vista, sota el paraigua de l'Processament de el Llenguatge Natural. En aquesta tesi ens centrem en dues àrees àmplies dins d'aquest camp, aplicades a l'anàlisi de contingut en línia: anàlisi de text en xarxes socials i resum automàtic. En paral·lel, les xarxes neuronals també són un tema central d'aquesta tesi, on tota l'experimentació s'ha realitzat utilitzant enfocaments d'aprenentatge profund, principalment basats en mecanismes d'atenció. A més, treballem majoritàriament amb l'idioma espanyol, per ser un idioma poc explorat i de gran interès per als projectes de recerca en els que participem.

D'una banda, per a l'anàlisi de text en xarxes socials, ens enfoquem en tasques d'anàlisi afectiu, incloent anàlisi de sentiments i detecció d'emocions, juntament amb l'anàlisi de la ironia. En aquest sentit, es presenta una aproximació basada en Transformer Encoders, que consisteix en contextualitzar *word embeddings* pre-entrenats amb tweets en espanyol, per abordar tasques d'anàlisi de sentiment i detecció d'ironia. També proposem l'ús de mètriques d'avaluació com a funcions de pèrdua, per tal d'entrenar xarxes neuronals, per reduir l'impacte de l'desequilibri de classes en tasques *multi-class* i *multi-label* de detecció d'emocions. Addicionalment, es presenta una especialització de BERT tant per l'idioma espanyol com per al domini de Twitter, que té en compte la coherència entre tweets en converses de Twitter. El comportament de tots aquests enfocaments s'ha provat amb diferents corpus, a partir de diversos *benchmarks* de referència, mostrant resultats molt competitius en totes les tasques abordades.

D'altra banda, ens centrem en el resum extractiu d'articles periodístics i de programes televisius de debat. Pel que fa a l'resum d'articles, es presenta un marc teòric per al resum extractiu, basat en xarxes jeràrquiques siameses amb mecanismes d'atenció. També presentem dues instàncies d'aquest marc: *Siamese Hierarchical Attention Networks* i *Siamese Hierar-*

chical Transformer Encoders. Aquests sistemes s'han avaluat en els corpora CNN/DailyMail i Newsroom, obtenint resultats competitius en comparació amb altres enfocaments extractius coetanis. Pel que fa als programes de debat, s'ha proposat una tasca que consisteix a resumir les intervencions transcrites dels ponents, sobre un tema determinat, al programa "La Noche en 24 Horas". A més, es proposa un corpus d'articles periodístics, recollits de diversos diaris espanyols en línia, per tal d'estudiar la transferibilitat dels enfocaments proposats, entre articles i intervencions dels participants en els debats. Aquesta aproximació mostra millors resultats que altres tècniques extractives, juntament amb una transferibilitat de domini molt prometedora.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Artificial Intelligence and more concretely, Machine Learning, is one of the most promising areas of Computer Science. It covers a lot of different research lines, being the most interesting and difficult to formalize those which are focused on studying human inherent abilities such as the vision or the spoken and written language understanding and generation. The basis of these skills is developed during the first years of human life. For example, the human vision starts recognizing small objects and finishes tracking complex objects, allowing eye-body coordination. The case of language acquisition is more intriguing, we know that language appears in all the neurotypical children inside very similar time frames. However, language continues evolving throughout human life, influenced by environmental, cultural, and socioeconomic aspects. Language is, likely, one of the psychological functions whose reality is closest to us, intervening in most of our activities.

From all these human inherent abilities, the understanding of the language is one of the most controversial and object of many theories intended to explain how the language is acquired and developed [3–7]. Although most of them differ in aspects like the innateness or the neural structures and interactions/stimulus required to learn the language, they all agree that language decisively influences the development of the mind. The language precedes the thoughts, and modifies their nature: levels of intellectual functioning depend on more abstract language. Due to the language, we are capable of communicating and understanding a potentially infinite number of messages, in many situations, by means of lexical, syntactic, semantic, and pragmatic components, with many communicative intents. Thus, language is one of the capabilities that most highlights the concept of intelligence, allowing to compose and to communicate ideas among humans with the aim of learning, understanding, reasoning, and taking decisions to compose ideas about their reality. An aspect closely linked to language is emotional intelligence [8], i.e. the essential ability of people to attend and perceive feelings appropriately and accurately. Proper management of feelings allows us to assimilate them,

understand them properly and improve the ability to modify, through language, our own state of mind and, to a certain extent, also that of others.

Language capacities are not useful if there is no notion of society. Communication is the support of life in societies; no group could survive without a continuous exchange of communicative elements. It is in societies where these abilities bloom and take relevance in their evolution. Since the origins of the language, until several years ago, the interactions among individuals were limited to hundreds or thousands of individuals, however, the development of the internet and particularly with the arrival of social networks, the possibility of interacting with any other individual of the world is a reality. These platforms have had, during the last years, an overwhelming impact on the daily life of individuals and organizations, becoming the predominant platforms for sharing and discussing content mainly by means of language. Individuals tend to express their opinions on these media platforms, in a straightforward/spontaneous way, and this opened the door to study social problems by focusing on the population that habit in the media platforms. Regarding the way users communicate in social networks, there are some aspects of the language that appear recurrently such as: extreme sentiment polarization about issues like politics or sports, knowledge and use of specific jargon of virtual social environments such as hashtags, emojis and abbreviations, and the use of figurative language like the irony in order to favor social interactions, evoking humor, diminishing or enhancing criticisms and getting the attention of the users by means of the creativity. The case of irony is very relevant in the context of affective analysis as it is closely connected with the expression of a feeling, emotion, and attitudes, acting as an implicit valence shifter.

Nowadays, society has access, and the possibility to contribute, to a big quantity of textual content present on the internet, such as social networks (Twitter, Facebook, etc.), online newspapers, forums, blogs, or multimedia content platforms such as YouTube. So, the need for automatic summarization systems has grown proportionally to the expansion, in terms of quantity and complexity, of these digital resources. Furthermore, the number of people with internet access also has experimented an exponential growth, so, presumably, not all of them have the same cognitive capabilities (special needs, intellectual disabilities, etc.) or the same background knowledge. In this way, the automatic summarization problem also covers a social dimension, posing as an effective solution for this type of users to understand the key content of the resources. Furthermore, these resources may be in different formats like video, audio, text, or even multimodal combinations among them, which favors the implantation of summarization systems in many different technologies.

As the reader has noticed, it is very difficult to formalize all the aspects of the language we discussed in this introduction in order to be automatically applied in practical situations such

as understanding affective aspects of a tweet that can be potentially communicated creatively by using figurative language; or to summarize newspapers or interventions of speakers in TV talk shows about controversial issues. Thus, it is also difficult to imitate them by means of computational approaches. The field intended to address these language-related problems by means of computational approaches is known as Natural Language Processing (NLP), and all the work of this thesis falls under its umbrella. Fortunately, despite the modeling difficulties, Deep Learning poses a powerful and flexible mathematical framework that allows us to implement models capable of learning linguistic knowledge and perform complex reasoning on top of linguistically-informed representations of raw data. However, it is worth noting that all these systems are based on statistical correlations, and for this reason, some language aspects are, nowadays, out of their scope [9].

Furthermore, in spite that the Spanish language is the world's second-most spoken native language (being the official language in 21 countries and existing different Spanish variants mainly in Latin American countries) and the third language most used by the users on the internet, the NLP research for the Spanish language is, by far, not as extensive as for the English language, and it is typically limited to following in the wake of advances in the English language. For this reason, we considered the Spanish language as the central language in the experimentations of this thesis, to contribute and motivate the study of computational approaches for addressing NLP problems in this widely spread and understudied language.

The close relationship between language and the human being, the difficulty of modeling it computationally, the opportunities social media platforms offer to study the language, the benefits that these technologies can bring to society and companies, and the study of computational approaches for the Spanish language, are the main challenges that have motivated this thesis and that will continue to motivate us to work in this field.

1.1 Objectives

The objectives pursued by this thesis aim towards developing new attention-based deep learning approaches and resources for text analytics in social media and summarization tasks, focusing on the Spanish language and transferring the developed technologies to research projects. In more detail, these objectives are defined as follows:

(O1) To develop state-of-the-art technologies for social media text analytics tasks. The goal is to focus on affective analysis tasks, including sentiment analysis and emotion detection, along with the analysis of the irony.

(O2) To propose new ideas, corpora, and models for extractive summarization.

(O3) To explore Deep Learning models as a central topic to build these technologies, with special emphasis on attention-based models and on techniques to improve the performance of the models on the addressed tasks.

(O4) To interpret Deep Learning models with the aim of study the linguistic knowledge that models capture, and observe its influence in the addressed tasks.

(O5) To apply all the developed technologies in research projects.

(O6) To work with the Spanish language as the main language in the experimentations, that is an interesting and underexplored language with a great interest for the research projects we participated in.

1.2 Contributions

The contributions of this thesis are in line with the objectives we addressed. Regarding the first objective (O1), we propose a system for contextualizing pre-trained word embeddings by means of Transformer encoders, both for sentiment analysis and irony detection on several Spanish corpora. This system was the first or second-ranked on all the experimentations, compared to the rest of the systems in the evaluation workshops. Nevertheless, we analyzed the behavior of the system by hypothesizing some properties that the model must learn to competitively address the tasks, and studying how these properties arise in the attention mechanisms of the models (O4). Also, for emotion detection, the number of emotions to be detected can be large and typically there are extreme class imbalance problems in the corpora. To this aim, we propose differentiable approximations of evaluation metrics common in text classification, that take into account the class imbalance, to be used as loss functions in order to guide the parameter estimation process (O1).

For the second objective (O2), we propose an attentional framework for extractive summarization, based on siamese hierarchical networks with attention mechanisms. It allows to develop models that dispense with extractive oracles and Reinforcement Learning techniques based on ROUGE to address the task as a sequential binary classification problem. Under this framework, we propose two different models based on different attentional encoders, Siamese Hierarchical Attention Networks, and Siamese Hierarchical Transformer Encoders. Furthermore, we apply some of the developed systems for summarizing the interventions of the speakers about given topics on Spanish TV talk shows. To this aim, we built a corpus from online Spanish newspapers, to pretrain the models, and we manually generated another corpus with reference summaries for the interventions. In this way, we explored the domain transferability using our proposal.

For the third objective (O3), we used attention-based models in all the experimentations, both integrated into recurrent architectures, and as sequence modelers. Concretely, we used attentional Long Short-Term Memories [10] as a robust baseline in the sentiment analysis and irony detection experiments, and as encoder in the attentional framework proposed. However, most of the modeling work of this thesis have been done with Transformers [11]. In this regard, we proposed the hierarchical transformers, and we used them in the attentional extractive summarization framework (O2). We also proposed the use of Transformer encoders for contextualized pre-trained word embeddings (O1). Besides, we developed an adaptation of BERT to address text classification tasks in Spanish Twitter (TWilBERT), which obtains significant improvements on several text classification tasks of international workshops. To this aim, we adapted the next sentence prediction signal for learning coherence between pairs of tweets inside Twitter conversations.

All this work was done under the scope of two research projects during the realization of this thesis (O5). The first one was ASLP-MULAN: Audio Speech and Language Processing for Multimedia Analytics. In this project, we intended to work in the direction of generating the right mixture of audio, speech, and language technologies, in order to offer it to companies that work daily with such multimedia content, like TV broadcasting companies, or to analytics companies interested in this information, improving their capacity to offer their services with increased quality, accuracy and usability of their reports. The second one is AMIC: Affective Multimedia analytics with Inclusive and natural Communication, which is intended to advance, develop and improve speech and language technologies as well as image and video technologies in the analysis of multimedia content adding to this analysis the extraction of affective information.

Regarding the last objective (O6), most of the current research is intended for the English language. Thus, there are lots of languages underexplored in the NLP field. Two especial languages are those that have more native speakers than the English language: the Standard Chinese and the Spanish languages. So, to develop technologies with potentially broad implantation, with a great interest for the research projects we participated, and to give our two cents to the Spanish language, we work with Spanish in all the addressed tasks.

Furthermore, the source code of all the experiments is publicly available. The models for sentiment analysis and irony detection are released as a transferable result through the Office for the Promotion of Research, Innovation and Technology Transfer (UPV), under the software SENTAT, ES-IRONIC and EN-IRONIC. The works with our attentional extractive summarization framework are available on three Github repositories: `AES`, `SHA-NN` and `SHTE`. The source code of the differentiable evaluation metrics for text classification can be accessed from `DEVM-TC`. Finally, we provided a framework for training, evaluating, and fine-tuning

BERT models, that also implements several improvements on Transformer models recently published in the literature. With this framework, we pretrained the TWilBERT models, whose weights are publicly available together with the source code of the framework in `TWilBERT`.

All the work of this three years has been reflected in 27 publications on several international journals, conferences and workshops:

Sentiment Analysis:

- José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Self-attention for twitter sentiment analysis in spanish. *Journal of Intelligent & Fuzzy Systems*, 39:2165–2175, 2020
- Rosario Sanchis-Font, Maria Jose Castro-Bleda, José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. Cross-domain polarity models to evaluate user experience in e-learning. *Neural Processing Letters*, May 2020
- Rosario Sanchis-Font, Maria Jose Castro-Bleda, and José-Ángel González. Applying sentiment analysis with cross-domain models to evaluate user experience in virtual learning environments. In Ignacio Rojas, Gonzalo Joya, and Andreu Catala, editors, *Advances in Computational Intelligence*, pages 609–620, Cham, 2019. Springer International Publishing
- José-Ángel González, José Arias Moncho, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at TASS 2020: TWilBERT for Sentiment Analysis and Emotion Detection in Spanish Tweets. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 179–186. CEUR-WS.org, 2020
- José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at TASS 2019: Transformer Encoders for Twitter Sentiment Analysis in Spanish. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, pages 571–578, 2019
- José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at TASS 2018: Sentiment Analysis in Twitter based on Deep Learning. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 37–44, 2018

- José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at TASS 2017: Sentiment Analysis in Twitter based on Deep Learning. In *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2017, co-located with 33th SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 37–44, 2017
- José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at SemEval-2017 task 4: Sentiment analysis using deep learning. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 723–727, Vancouver, Canada, August 2017. Association for Computational Linguistics

Emotion Detection:

- Lluís-F Hurtado, José-Ángel González, and Ferran Pla. Choosing the right loss function for multi-label emotion classification. *Journal of Intelligent & Fuzzy Systems*, 36(5):4697–4708, 2019
- José-Ángel González, Lluís-F. Hurtado, and Ferran Pla. ELiRF-UPV at SemEval-2019 task 3: Snapshot ensemble of hierarchical convolutional neural networks for contextual emotion detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 195–199, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics
- José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV en TASS 2018: Categorización emocional de noticias (ELiRF-UPV at TASS 2018: Emotional Categorization of News Articles). In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 103–109, 2018

Irony Detection:

- José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. *Information Processing & Management*, 57(4):102262, 2020
- José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at IroSvA: Transformer Encoders for Spanish Irony Detection. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, pages 278–284, 2019

- José-Ángel González, Lluís-F. Hurtado, and Ferran Pla. ELiRF-UPV at SemEval-2018 tasks 1 and 3: Affect and irony detection in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 565–569, New Orleans, Louisiana, June 2018. Association for Computational Linguistics

Summarization:

- José Ángel González, Encarna Segarra, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Extractive summarization using siamese hierarchical transformer encoders. *Journal of Intelligent & Fuzzy Systems*, 39:2409–2419, 2020. 2
- José-Ángel González, Lluís-Felip Hurtado, Encarna Segarra, Fernando Garcia-Granada, and Emilio Sanchis. Summarization of Spanish Talk Shows with Siamese Hierarchical Attention Networks. *Applied Sciences*, 9(18), 2019
- José-Ángel González, Segarra Encarna, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Siamese hierarchical attention networks for extractive summarization. *Journal of Intelligent and Fuzzy Systems*, 36(5):4599–4607, 2019
- Emilio Sanchis Fernando García-Granada José Ángel González, Julien Delonca and Encarna Segarra. Applying Siamese Hierarchical Attention Neural Networks for multi-document summarization. *Procesamiento del Lenguaje Natural*, 63(0):111–118, 2019

Others:

- José Ángel González, Lluís-F. Hurtado, and Ferran Pla. TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter. *Neurocomputing*, 426:58 – 69, 2021
- José-Ángel González, Lluís-F. Hurtado, Encarna Segarra, and Ferran Pla. ELiRF-UPV at SemEval-2018 task 10: Capturing discriminative attributes with knowledge graphs and Wikipedia. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 968–971, New Orleans, Louisiana, June 2018. Association for Computational Linguistics
- José-Ángel González, Lluís-F. Hurtado, Encarna Segarra, and Ferran Pla. ELiRF-UPV at SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 1034–1037, 2018

- José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at MultiStanceCat 2018. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 173–179, 2018
- José-Ángel González, Ferran Pla, and Lluís-Felip Hurtado. ELiRF-UPV at IberEval 2017: Stance and Gender Detection in Tweets. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 193–198, 2017
- José-Ángel González, Ferran Pla, and Lluís-Felip Hurtado. ELiRF-UPV at IberEval 2017: Classification Of Spanish Election Tweets (COSET). In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 55–60, 2017
- Lluís-F. Hurtado, Encarna Segarra, Ferran Pla, Pascual Carrasco, and José-Ángel González. ELiRF-UPV at SemEval-2017 task 7: Pun detection and interpretation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 440–443, Vancouver, Canada, August 2017. Association for Computational Linguistics
- Victor Nina-Alcocer, José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. Aggressiveness detection through deep learning approaches. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 544–549. CEUR-WS.org, 2019
- Fernando García-Granada, Emilio Sanchis, María José Castro Bleda, José-Ángel González, and Lluís-F. Hurtado. Word discovering in low-resources languages through cross-lingual phonemes. In Albert Ali Salah, Alexey Karpov, and Rodmonga Potapova, editors, *Speech and Computer - 21st International Conference, SPECOM 2019, Istanbul, Turkey, August 20-25, 2019, Proceedings*, volume 11658 of *Lecture Notes in Computer Science*, pages 133–141. Springer, 2019

1.3 Thesis Outline

In this section, we briefly describe the outline of this document, summarizing the contents that can be found in the four different chapters in which we organized it:

Chapter 2. Deep Learning: the purpose of this chapter is merely introductory to the Deep Learning models we used from an NLP perspective. We discuss the basic aspects of feedforward networks and some practical strategies we used in our experiments. We introduce the sequence modeling problem and briefly introduce a typology of tasks and models intended for this aim. A formalization of the models we used in the experiments is also provided in this chapter: Convolutional Neural Networks, Recurrent Neural Networks, attention mechanisms, and Transformers. Furthermore, we discuss the success of Deep Learning for text representation learning, focusing on non-contextual and contextualized embeddings.

Chapter 3. Text Analytics in Social Media: in this chapter we discuss the sentiment analysis problem and our proposal based on transformer encoders, along with an extensive experimental study. It also contains our work on the emotion detection field, where we propose a novel way to optimize evaluation metrics with multi-label emotional classes. We also present in this chapter our work in irony detection, where we proposed a transformer encoder model and we deeply study its connection with sentiment analysis and the ironic features learned by the system. Part of the research shown in this chapter was published in three papers by the author [12, 20, 23].

Chapter 4. Pre-trained Deep Bidirectional Transformers for Spanish Twitter: in this chapter we present our adaptation of BERT to the Twitter domain and the Spanish language (TWilBERT), which shows large improvements in comparison to multilingual versions of BERT for sentiment analysis, emotion detection, irony detection, and other text classification problems. Part of the research shown in this chapter was published in one paper by the author [30].

Chapter 5. Automatic Summarization: the purpose of this chapter is to present our proposals for automatic summarization. Concretely, a formalization of the theoretical framework for extractive summarization, two instantiations of this framework with encoders based on attentional recurrent networks and Transformers, and their application to the summarization of news articles and Spanish TV talk shows are presented. Regarding the summarization of Spanish TV talk shows, this chapter also presents the corpora we built in order to pretrain summarization systems on Spanish news articles

and to evaluate their domain transferability on the TV talk shows domain. Part of the research shown in this chapter was published in three papers by the author [26–28].

Conclusions and Future Works: in this chapter, we summarize the work performed in this thesis. We present both the conclusions derived from each specific work and holistic conclusions to discuss all the work as a whole. Finally, future lines of works and a discussion of the extensions, in which we are working currently, are presented.

Chapter 2

Deep Learning

Deep learning is a central topic in this thesis that has been used across all the experimentations for sentiment analysis, emotion detection, irony detection, and automatic summarization of newspapers and TV talk shows. In order to automatically understand what a tweet means, its emotional content, and the presence of figurative language, or to compress all the information of a document in a brief high-quality summary, it is required to model the mental processes and the knowledge that humans use to this aim. In contrast to other tasks where the data is almost perfectly defined in terms of a fixed set of features (e.g., identify the family of the iris flowers in terms of petal/sepal length and widths), the processing of natural languages has to deal with a cognitive problem: understanding a potentially infinite number of messages composed in terms of a finite set of tokens, syntactical rules to combine them and semantic functions that assign meaning to the tokens and their combinations. Therefore, it is required to know about lexical, syntactical, and semantic aspects in order to process the form of a text message and to understand its meaning.

It is very difficult to formalize human being language abilities, that process immense amounts of world knowledge, to imitate them through computational approaches. Fortunately, machine learning poses a framework for extracting abstract patterns from raw data. However, classical machine learning approaches heavily depend on high-quality representations in order to perform well on NLP tasks, and this feature designing step is especially difficult. This central problem is solved by Deep Learning models, that can perform representation learning. Deep Learning models are particularly flexible and powerful due to their ability to represent, in terms of correlations, the input data as a nested hierarchy of concepts, modeling complex abstract concepts in terms of simple ones such as semantic aspects defined in terms of syntactical and surface-level patterns.

One of the main objectives of this thesis is to explore and develop new Deep Learning technologies for social media text analytics and automatic summarization. So, this chapter

is intended to cover some modeling aspects that are not studied deeply in Chapters 4 and 5, contextualizing and formalizing each one of the models and techniques we used in the experimentations. As most of the topics discussed in this section have been deeply discussed in the literature, the purpose of this chapter is merely introductory to the models we used from an NLP perspective. This chapter is structured as follows. First, in §2.1, we discuss the basic aspects of feedforward networks and some practical strategies we used in our experiments. In §2.2 we introduce the sequence modeling problem and briefly introduce a typology of tasks and models intended to this aim. In §2.2.1, §2.2.2, §2.2.3 and §2.2.4, we formalize the models we used in our experiments: Convolutional Neural Networks [39], Recurrent Neural Networks [10], attention mechanisms [40–43] and Transformers [11]. Finally, in §2.3 we discuss the success of Deep Learning for text representation learning, focusing on non-contextual §2.3.1 and contextualized embeddings §2.3.2.

2.1 Feedforward Networks

Feedforward networks are the most relevant model in the Deep Learning field for facing non-sequential data. They are composed of a bunch of computational units, called neurons, arranged as a stack of layers, where there is not feedback between the neurons of a layer and the previous layer's one. A feedforward network (and, typically, all artificial neural networks) consists of, at least, three different blocks of layers: input layers, hidden layers, and output layers. The input layers act as an entry point of the data to the network, thus allowing the flow of the input through the subsequent layers. The hidden layers define non-linear transformations of their inputs, which can be seen as a set of features that represent those inputs in terms of different abstractions. Finally, the output layers draw the connection between the transformations computed by the hidden layers and the desired output. By this way, feedforward networks are function approximators, that define a parameterized mapping, $y = f_\theta(x)$, to approximate some desired function $y^* = f^*(x)$. The number of hidden layers is known as the depth of the model, and by means of consecutive applications of hidden layers, the mapping defined by the feedforward networks can be seen as a function composition of the functions computed by the output layer and all the hidden layers, taking the input from the input layer. The number of neurons in the hidden layers determines the width of the model and the dimensionality of the output of each hidden layer.

The hidden layers play a very relevant role in this architecture and they are the key point of the Deep Learning success. Without hidden layers, feedforward networks implement linear functions. Furthermore, if there is at least one hidden layer, but all of them compute linear functions, then the function modeled by the feedforward networks is also linear. In this way,

to increase the capacity of this architecture, nonlinear activation functions are considered on top of, at least, the output of one hidden layer, thus being able to model non-linear functions. To better illustrate this fact, it is required to understand the behavior of the neurons inside a hidden layer. Although these neurons can be specialized, like in Convolutional Neural Networks (CNN) or Transformers to consider receptive fields or attentions respectively, the basic unit for feedforward networks computes affine transformations of its inputs, thus inherently computing linear functions. Nonlinear activation functions are applied on the affine transformation to address this lack of capacity, as shown in Figure 2.1. Neural networks made up of this kind of neurons, in which all neurons in layer i are connected to those in layer $i+1$, are called fully connected feedforward networks.

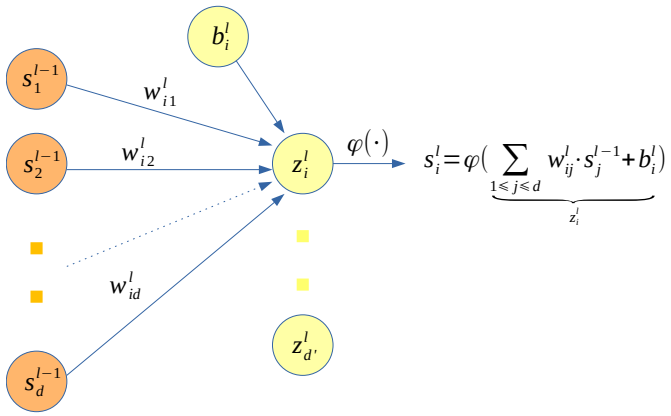

Fig. 2.1 Simple feed-forward network.

In Figure 2.1, w_{ij}^l is the weight that connects the neuron i of the layer l with the neuron j of the layer $l-1$, s_i^l is the output of the neuron i in the layer l that is computed by means of an activation function φ applied on the affine transformation that involves the neurons connected to i, b_i^l is the bias for the neuron i in the layer l, and d, d' are the number of neurons in layers $l-1$ and l respectively. It should be noted that this computation should be repeated for all the neurons in the layer l. However, a compact formulation in terms of matrix notation is more convenient, both for implementation and clarity aspects. This operation can be seen as a matrix product between the weight matrix $W^l \in \mathbb{R}^{d' \times d}$, that connects all the neurons in the layer $l-1$ with all the neurons in the layer l, and the output of the previous layer, s^{l-1}, adding the bias vector and the non-linear activation function, $s^l = \varphi(W^l s^{l-1} + b^l)$. Most of the formal model definitions used in this thesis employ matrix notations. The choice of the activation function is highly task-dependent, but some activation functions were historically used as *de facto* standards, like sigmoid (also known as logistic) or hyperbolic tangent. However, these activation functions saturate for large positive or large negative preactivation values and make it almost impractical for deep models. Nowadays, the most

widely used solutions to the saturation problem are based on Rectifier Linear Units (ReLU) and its variants like Exponential Linear Units (ELU) or Gaussian Error Linear Units (GELU) to alleviate the dying ReLU problem that arises for negative values. When it is required to estimate the conditional probability of some dependent variable given observations from the networks, the softmax activation function is typically used. The two most known contexts where softmax is used are the computation of posterior probabilities of classes given the last output of the networks in classification tasks, and in attention mechanisms to explicitly focus on the most salient parts of the observations. Figure 2.2 shows a visual representation and the definition of the activation functions used in this thesis, along with their derivatives.

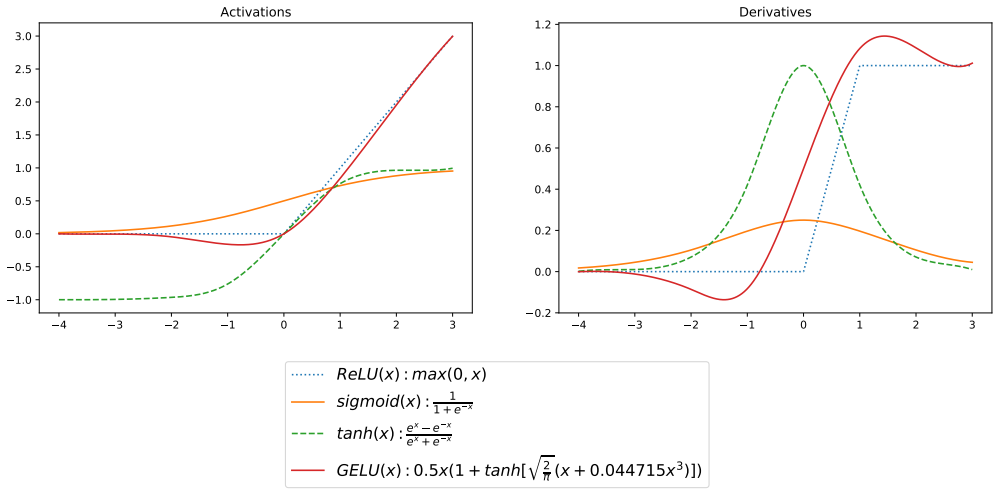

Fig. 2.2 Activation functions we used in this thesis, along with their derivatives.

As stated before, the number of hidden layers plays a very relevant role in the Deep Learning success. Deep Learning models are particularly flexible and powerful due to their ability to represent the input data as a nested hierarchy of concepts, modeling complex abstract concepts (that arise in late hidden layers) in terms of simple ones (that arise in early hidden layers). For example, in order to detect a car from a raw image, the earlier layers of the networks could capture surface levels patterns like edges or some simple geometric figures, that will be specialized in later layers in order to represent more complex patterns like wheels or headlights. Although it is easy to understand this in the context of computer vision, in NLP this is less intuitive and, nowadays, it is an active line of research, mainly on deep pretrained language models that we will discuss in §2.3. Also, a big part of Chapter 3 is intended to this aim. In this regard, it is convenient to recall a very interesting theoretical result [44, 45], which states that feedforward networks with a single non-linear layer are universal approximators (and also with more than one non-linear hidden layers), in the sense that they can approximate any function we want to learn, under specific conditions regarding

the network's size. So, why we need Deep Learning? this is mainly due to the network's size condition, as in the worst case, the single hidden layer requires an exponential number of units [45]. Deeper models can reduce the number of units required to represent the desired function with better generalization error [46].

Feedforward networks are trained by gradient descent, which requires designing some components such as a loss function, an optimization procedure, and a model. The model is defined by its parameters, that, in the case of neural networks, it is parameterized by the set of weights of all the layers, $\theta = \{w_{ij}^1, ..., w_{ij}^L\}$. Roughly, the loss function tells us how good are the network predictions compared to the expected outputs, and, as in most cases the function defined by neural models is a probability distribution $p_\theta(y|x)$, it measures the differences between two probability distributions (the predicted distribution and the training data distribution). In this way, neural models are usually trained under maximum likelihood approaches, in order to estimate the parameters of the network that minimize, in practice, the negative log-likelihood. The optimization procedure consists in estimating θ to minimize the loss function, by means of iterative updates that require the derivative of the loss function with respect to each weight inside the network. To compute these derivatives, the Backpropagation algorithm (BP) [47] that computes an exact analytical solution to this purpose, is used. Basically, what BP does is to take the desired output and the predicted output (after a forward pass on the network), to compute the gradient of the loss function with respect to the last weights in the network, and then propagate these errors to previous weights in the network (backward pass). With the derivatives of the error for all the weights inside the network, some update rule of gradient descent is applied to move these weights towards the negative gradient of the loss function e.g., $\theta = \theta - \alpha \frac{\partial \mathscr{L}}{\partial \theta}$, where α is the learning rate that controls the magnitude of the updates. As it can be noted, all the components have to be differentiable in order to perform BP and estimating the parameters of the models.

This strategy entails lots of practice decisions to stabilize the training process, which can be prone to vanishing/exploding gradients due to the saturation nature of some activation functions, or to overfit, thus losing the ability to generalize correctly on unseen sample distributions. Some techniques have been used as standard design choices in the neural network design for this purpose, such as weight initialization, training modes, regularization, or specific update rules. The following items detail those techniques used in the experiments performed in this thesis:

- **Weight initialization**: an initial weight configuration is required in order to train neural networks by using gradient descent. Intuitively, the more similar the initial weights are to the optimal weights, the better results obtained in the first training epochs and the faster the convergence is. Usually, the weights are initialized to some

random values drawn from normal distributions. Some initializations based on this have become the most widely used in the designing of neural networks: Glorot [48], and truncated normal distributions. In all the experiments of this thesis, we used truncated normal distributions, that discard and redraw values more than two standard deviations from the mean. This initialization, with a low standard deviation, was crucial for training deep models such as the presented in chapter 4.

- **Dropout**: as most of the models used in this thesis were trained on small corpora, they are prone to overfit even if they are trained during few epochs. We used Dropout in some of these models in order to prevent overfitting. In dropout, a random variable $r_j^l \sim Bernoulli(p)$ controls the override of the neuron j of the layer l with a probability p. By this way, the output of the neuron is defined as $\tilde{s}_j^l = r_j^l s_j^l$ where s_j^l is the neuron output before the override. This strategy forces that, in each step, the activated neurons are different from those activated in previous steps, so, priming the specialization of all the neurons for capturing relevant patterns. It can be seen as an implicit ensemble of very different connectivity patterns in the network [49]

- **Input noise**: random noise is typically added to the weights of the networks as a form of regularization to improve the learning stability. In some experiments, we used Gaussian noise in the inputs to perform data augmentation, due to the reduced size of some corpora we used in the experimentations, in order to reduce overfitting. Also, this input noise has an additional regularization effect, as it is equivalent to impose a penalty on the norm of the weights [46, 50].

- **Training mode**: gradient descent training can be done on different bunch of samples, which are known as training modes. Three different modes are considered in the literature: batch gradient descent, stochastic gradient descent, and mini-batch gradient descent. In batch gradient descent, the weights are updated after computing the gradients for all the samples in the dataset, however, for large datasets, the time to take a single gradient step becomes prohibitively. By contrast, stochastic gradient descent updates the weights after computing the gradients for individual samples in the dataset. In this case, the gradient is an expectation of the gradient for all the samples, using only one sample, so, it could not be a good estimation. In between batch and stochastic gradient descent, mini-batch gradient descent aims to address these problems by increasing the number of samples used for computing the gradients. This number of samples is commonly known as batch size. In this thesis we only used the mini-batch training mode with different batch size depending on the experiment e.g., in chapter 4 we used batches of 2048 sequences due to, deep language models

based on Transformers are highly benefited from training on large batches and large corpora, but in §3.1 we simply used batches of 32 sequences as the size of the corpora used is small in comparison to the previous case.

- **Normalization**: normalization techniques are used to make neural networks more stable through normalization of the inputs (from the input layer and/or from hidden layers) by re-centering and re-scaling them. Two widely used approaches are batch normalization [51] and layer normalization [52]. Both of them are very similar, as batch normalization normalizes the inputs across the batch dimension and layer normalization does the same across the features. Usually, they normalize each batch/sample such that the elements have zero mean and unit variance, and then these elements are scaled and shifted by the learnable parameters γ and β, $N_{\gamma,\beta}(x_i) = \gamma\hat{x}_i + \beta$. We used layer normalization in all the experiments involving Transformer models, while batch normalization was used for the experiments with Convolutional Neural Networks [39] and simple feedforward networks like Deep Averaging Networks [53].

- **Skip connections**: although the problem of vanishing/exploding gradients have been largely addressed by means of proper normalizations, initializations, and activation functions, when very deep models are trained with gradient descent and BP, the performance with the training set gets saturated and then degrades rapidly [54] if the number of hidden layers grows. This degradation manifests the difficulty of the optimization process, as models with higher capacity could obtain worse performance than those with lower capacity, even in the training set. This should be not possible as, by construction, one always can consider identity layers to build larger networks with the same performance as smaller ones. Unfortunately, it seems that it is difficult to be considered by the optimization process. To address this inconvenience, skip connections were proposed for allowing the gradients to flow through blocks of layers [54]. In this way, earlier network outputs can be passed more directly to deeper parts of the networks, as the length of the shortest path between the output layer and early layers is shorter, to improve the signal propagation. In this thesis, we used skip connections to build residual blocks in the TWilBERT model presented in chapter 4.

- **Update rules**: gradient descent poses some challenges, mainly related to the learning rate, that make difficult the optimization process. In practice, it is difficult to find proper learning rates, and also, it should be annealed during training. Furthermore, the same learning rate is used in order to update all the weights, which is not recommended for example, if the data is sparse. Different optimizers have been proposed in the literature to address these challenges. In this thesis, we used Adaptive Moment Estimation

(Adam) [55] for all the models, except TWilBERT, where we used AdamW and LAMB [56]. The idea behind Adam consists in computing adaptive learning rates for each parameter, and, if weight decay is considered, it is known as AdamW. LAMB is a layer-wise adaptive update rule especially useful for large batches, that shown empirically how deep models like BERT [57] can be trained in about an hour, instead of several days.

- **Class imbalance countering**: in some classification problems, the class imbalance biases the outputs of the models towards the most populated classes, being thus not able to generalize to the less populated classes with which obtain a very low recall. Although there are many different and widely used approaches such as undersampling and oversampling (either repeating samples or generating new synthetic ones by means of data augmentation), they pose some problems related to the amount and the quality of the samples in the datasets. Fortunately, there are other approaches that do not involve modifications of the corpora such as loss weighting and loss functions based on evaluation metrics that consider the imbalance among the classes. On the one hand, in loss weighting, each class has an associated multiplicative factor, that is higher for minority classes and lower for majority classes. In this way, the errors are weighted by these factors, thus forcing the models to classify better the samples of the minority classes. Usually, the factor for the class c, w_c, is defined in terms of the number of samples in the class c and the number of samples in the most populated class $\hat{c}$ e.g., $w_c = \frac{\#\hat{c}}{\#c}$. On the other hand, in this thesis, we propose the use of loss functions based in evaluation metrics that consider the imbalance among the classes, with two different purposes. The first one is to address the class imbalance like the previous approach, and, in fact, they have shown to be very effective on this aim (§3.2). The second one is to address the mismatch between cross-entropy and some evaluation metrics widely used to evaluate text classification models, like macro-averaged F_1 (§A.1).

At this point, we defined the basic form of a feed-forward network to define a parameterized mapping $y = f_\theta(x)$ where the input x is a real-valued vector, however, this configuration is usually not the most suitable for NLP problems. In most of the NLP cases, it is required to reason about documents, which are compositions of tokens whose interpretation is tied to their relationships. Thus, it is complex to represent documents by means of a single real-valued vector. In the following section, several techniques and models to approach sequence modeling tasks are discussed, with special emphasis on those used in this thesis. Furthermore, it should be noted that most of the techniques presented in this section are generalizable to sequence modeling tasks.

2.2 Sequence Modeling

Most of the Natural Language Understanding and Generation tasks are inherently sequence modeling tasks that require reasoning about the text structure, basically, by means processing the text as sequential data of variable length. A rough taxonomy of the text sequence modeling tasks is shown in Table 2.1, that distinguishes three different granularities depending on the sequence length of the inputs and the targets: many $\rightarrow$ one, many $\rightarrow$ many and many $\rightarrow$ many$^+$. These granularities are directly related to the typology of the task, and they are usually known as text classification, sequence labeling, and text generation respectively. It should be noted that in all the considered tasks, the input consists of many elements, thus considering the compositional nature of the text. The tasks addressed in this thesis fall under text classification (sentiment analysis, emotion detection, and irony detection) and sequence labeling (extractive summarization).

Table 2.1 Taxonomy of text sequence modeling tasks, as usually defined in the literature.

Type	**Tasks**
many $\rightarrow$ one (text classification)	sentiment analysis, emotion detection, fake news detection, textual entailment, multiple-choice question answering, ...
many $\rightarrow$ many (sequence labeling)	POS tagging, Name Entity Recognition, Extractive Summarization...
many $\rightarrow$ many$^+$ (text generation)	hierarchical text classification, dialogue generation, machine translation, abstractive summarization, ...

The most predominant approaches to deal with text sequentiality by means of neural networks are:

- **Collapsing**: collapse the temporal dimension of text sequences. Some examples are bag-of representations and word embedding collapsing. Word embedding collapsing has been extensively used in this work as a robust baseline, and it basically consists of a mapping from $\mathbb{R}^{T\times d} \rightarrow \mathbb{R}^d$. It is the basis of the Deep Averaging Network (DAN) [53], where the mapping is the average of the word embeddings that is used as input to a fully connected feedforward network.

- **Memory Bounded Networks**: feedforward networks (without internal memory) that can deal with arbitrary-length contexts for each token in the sequences. A naive

approach consists directly on applying a fully connected feedforward network to each token in the sequence. So, that network has to learn all the aspects of the language independently at each position, without taking into account any context for the tokens. NetTalk was the first proposal to consider the context [58], by using the left and right contexts of each token at each timestep. Later, Time Delay Neural Networks were proposed to classify patterns with shift-invariance, and modeling the context at each layer of the network. More recently, although they have been proposed in 90's [59, 60], 1-D Convolutional Neural Networks (CNN) gained a lot of interest in the NLP field [39, 61, 62], and they consist on applying the multiple convolution kernels, shared across time, to compute representations in terms of a small number of neighboring members of the input. Finally, the Transformer model [11] is the most recent example of memory bounded networks, that has boosted the state of the art in almost all NLP tasks. It is based on self-attention mechanisms that generalize the fully connected feedforward networks for sequence modeling by means of computing similarities among all the tokens of the sequence, thus reducing the complexity to find relationships among words to $\mathcal{O}(1)$.

- **Recurrent Neural Networks**: This type of models differs from the previous ones in that it contains additional parameters (internal memory) to consider previous states and outputs of the model at each timestep. In their basic form (known as simple RNN), the projection of the input at timestep t is combined with the projection of the state in the timestep $t-1$ to compute the current state. While the Memory bounded networks are trained with Backpropagation, the optimization technique used to train these networks is Backpropagation Through Time (BPTT). This training strategy conducts to unstable gradients, which are prone to vanish or to oscillate. For this reason, the most predominant approaches are gated RNN such as Long Short-Term Memory (LSTM) [10] and Gated Recurrent Unit (GRU) [63], that alleviates the gradient instability and are also capable of internally regulate the context (amount of memory) required at each timestep.

In the experimentations, we used approaches based on Memory Bounded Networks and Recurrent Neural Networks, focusing on studying those composed by self-attention mechanisms, especially Transformer encoders. We used DAN and LSTM with attention mechanisms (Att-LSTM) as baselines in §3.1 and §3.3. In §3.2.3 we used CNN to evaluate the loss functions that approximate evaluation metrics in order to train Deep Learning models. In chapter 4, we pretrain Transformer encoders to learn bidirectional contextualized representations of Spanish tweets, and we also provide a framework for pre-training and

finetuning these encoders. In §5.2 and §5.3, we used hierarchical Att-LSTM [41] and we propose a hierarchical Transformer encoder in order to select salient sentences from the attention mechanism, for extractive summarization. By this way, Transformer encoders have been used in all the experimentations except in those of §3.2.3. We roughly describe all these systems in the following subsections.

2.2.1 Convolutional Neural Networks

In [59] and [60] were proposed parameter sharing and position invariance strategies for handwritten digit and phoneme recognition. These works were the first successful attempts to extract local features and combining them to form higher-order features, considering the idea of identifying distinctive features of objects at different locations. Nevertheless, they are also the basis for the modern CNN architecture, that has been ubiquitous in the Computer Vision field due to they are inductively biased towards some desired properties in vision: local receptive fields, shared weights, and spatial sub-sampling. In that sense, AlexNet represented the explosion of CNN in Computer Vision, reducing drastically the error rate on the ILSVRC competition by taking profit of GPU resources and recent techniques for successfully training an 8-layered CNN. The explosion of CNN in the NLP field was later, although they had been proposed for dealing with 1-D sequences [59]. Concretely, for the last five years, they gained a lot of interest in the NLP research community after they improved the state of the art in several sentence classification tasks [39, 61, 62, 64], by being trained on top of pre-trained word embeddings. Furthermore, as these models are highly parallelizable, thus fast to compute, and with few parameters, they were widely accepted in the research community.

CNN consists of a stacked application of convolutional layers and pooling operations, that serve as feature extractors, usually for classification models implemented in terms of fully connected feed-forward networks. The convolutional layers are modeled by a set of learnable weights, called kernels, that are locally convoluted with the inputs to obtain feature maps. While in Computer Vision they compute features from 2-dimensional image patches, in NLP they work on 1-dimensional sequences of n-grams, thus computing features that can be interpreted as continuous combinations of n-grams at different depths. Figure 2.3 shows the dependencies modeled by this way in CNNs, with the aim of comparing with other sequence modelers discussed in the following sections. Note that, the receptive field, in terms of n-grams, grows with the number of stacked convolutional layers, as the features in the layer l are continuous n-grams computed

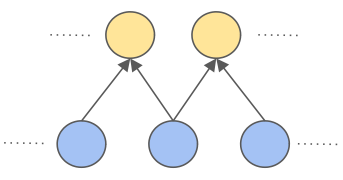

Fig. 2.3 Dependencies modeled by CNNs

from the output of the layer $l-1$ (also continuous n-grams). The pooling operator, in his basic form, is an unparameterized way for increasing the receptive field, reduce the number of parameters in the network and improve the local translation invariance in order to generalize better.

For some text classification experiments of this thesis, we used the architecture of [39], that is a simple CNN with only one convolutional layer on top of pretrained word vectors. This is the main difference with respect to the usual CNN, as instead of growing deep, they grow wide, by computing different kernels of different heights, in parallel from the input[1]. Concretely, a set of kernels with different heights $\{k_1,...,k_K\}$ are applied to the input $X \in \mathbb{R}^{T\times d}$, where T is the length of the sequence and d the dimensionality. This set of learnable kernels can be defined as $\Theta = \{\theta_1,...,\theta_K : \theta_i \in \mathbb{R}^{k_i\times d\times F}\}$, where F is the number of kernels (the same for all the heights). Thus, the input is convolved by each one of these kernels, before applying an activation function, to get the feature maps, $h_i \leftarrow \varphi(X \circ \theta_i) \in \mathbb{R}^{(T-k_i+1)\times F}$. A max-pooling operator is applied then to each h_i, $h_i \leftarrow \text{MaxPool}(h_i) \in \mathbb{R}^{\left\lfloor \frac{T-k_i+1}{\text{pool\_size}} \right\rfloor \times F}$. Then, each h_i is flattened as a single vector of size $\left\lfloor \frac{T-k_i+1}{\text{pool\_size}} \right\rfloor \cdot F$, and all these flattened vectors are concatenated, $h = [h_1,...,h_K] \in \mathbb{R}^{\sum_{i=1}^{K} \left\lfloor \frac{T-k_i+1}{\text{pool\_size}} \right\rfloor \cdot F}$. Finally, in our case, for text classification, this vector representation h was used as input for a fully connected feedforward network to perform the classification on top of the extracted features. For more details about the architecture, we refer the reader to the reference paper [39] and to §3.2.2.

2.2.2 Recurrent Neural Networks

This kind of networks contain additional parameters in order to consider previous states of the model in each timestep, thus drawing cyclic directed graphs differently from feedforward networks. The simplest RNN processes sequentially, from left to right, each vector (x_t) of a sequence $X \in \mathbb{R}^{T\times d}$ at each timestep (t), combining it with previous states of the network (h_{t-1}), as shown in Eq. 2.1.

$$h_t = \varphi(Wx_t + Uh_{t-1} + b) \tag{2.1}$$

where $W \in \mathbb{R}^{d_h\times d}$ and $U \in \mathbb{R}^{d_h\times d_h}$ are the weight matrices for projecting the inputs and the states respectively, d_h is the dimensionality of the state, $b \in \mathbb{R}^{d_h}$ is the bias and φ is the activation function. By this way, an output sequence $y = \{y_1,...,y_T\}$ can be computed from

[1] Just for curiosity: in that years the application of deep models was difficult in the NLP field, due to the lack of pre-training with large corpora like in computer vision, so, possibly that decision for growing wider instead of deeper was due to this. Furthermore, this is more similar to how text classification was addressed historically, by using different n-gram sizes directly extracted from the input, instead of from subsequent representations.

all the hidden states $h_{1\leq t\leq T}$ either by $y = h$ such as the Jordan networks [65] or by means of augmented neural networks that are compositions of simple RNNs and fully connected feed-forward networks such as the Elman networks [66]. The dependencies modeled by RNNs are shown in Figure 2.4. While the Memory bounded networks are trained with Backpropagation, the optimization technique used to train RNNs is Backpropagation Through Time (BPTT). Roughly speaking, it basically consists in unrolling the network a fixed number of timesteps and propagate backwards the gradients from later timesteps to early ones. This strategy, directly applied on simple RNNs, lead to unstable gradients which are prone to vanish or to oscillate if the sequence is relatively large, thus losing the capability to take into account long-term relationships. Furthermore, the output depends on all the past information, which makes difficult to the network to adjust its internal states to retain shorter contexts. In order to address these issues, some techniques can be used like non-saturating activation functions or gradient clipping. However, the most widely stablished approach to deal with that issues is to use gated recurrent networks that can regulate the flow of the information through the recurrency, such as LSTM [10] and GRU [63]. In this thesis we used LSTMs (with attention mechanisms, that will be discussed in the next section) as defined in the following equations:

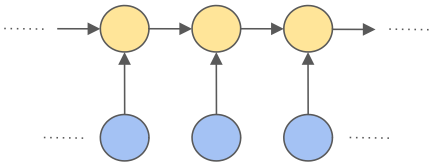

Fig. 2.4 Dependencies modeled by RNNs

$$
\begin{aligned}
\mathrm{i}_t &= \sigma(W^{\mathrm{i}}x_t + U^{\mathrm{i}}\mathrm{h}_{t-1}) \\
\mathrm{f}_t &= \sigma(W^{\mathrm{f}}x_t + U^{\mathrm{f}}\mathrm{h}_{t-1}) \\
\mathrm{o}_t &= \sigma(W^{\mathrm{o}}x_t + U^{\mathrm{o}}\mathrm{h}_{t-1}) \\
\tilde{\mathrm{c}}_t &= tanh(W^{\mathrm{c}}x_t + U^{\mathrm{c}}\mathrm{h}_{t-1}) \\
\mathrm{c}_t &= \mathrm{f}_t \odot \mathrm{c}_{t-1} + \mathrm{i}_t \odot \tilde{\mathrm{c}}_t \\
\mathrm{h}_t &= tanh(\mathrm{c}_t) \odot \mathrm{o}_t
\end{aligned}
\tag{2.2}
$$

where $\mathrm{i}_t, \mathrm{f}_t, \mathrm{o}_t$ y $\tilde{\mathrm{c}}_t$ are the outputs from the input, forget, output and context gates respectively in the timestep t, σ and $tanh$ are the sigmoid and hyperbolic tangent activation functions, c_t is the context (internal memory) and h_t is the hidden state. All these outputs are in $\mathbb{R}^{d_h}$, where d_h is the dimensionality of the hidden state, so, all the weight matrices W are in $\mathbb{R}^{d_h \times d}$ and the weight matrices U are in $\mathbb{R}^{d_h \times d_h}$. The purpose of each layer is to regulate the information flow from previous timesteps in c (internal memory) and in h (hidden state). On the one hand, c_t is regulated by the forget gate f_t, that indicates how much information has to be erased from the previous memory, plus the updated memory state for the current timestep $\tilde{\mathrm{c}}_t$. At the

same time, $\tilde{c}_t$ is also regulated by i_t. Finally, the hidden state, h_t, is computed by means of modulating the context c_t using the output gate o_t, that indicates how much to reveal from the internal memory.

One of the weaknesses of the discussed RNNs is that the sequences are processed from left to right, thus lacking bidirectionality as they do not take into account the information flow from right to left. To address this issue, bidirectional RNNs were proposed [67]. Although not all the NLP tasks require bidirectionality e.g., text generation where there is no access to future information, these RNNs are especially useful in tasks where it is required to consider bidirectional contexts. Nowadays, the most relevant example is the cloze task, implemented in terms of bidirectional language modeling [68] or as masked language modeling [57]. For instance, in the example "*Neural* ____ *improved the state of the art in NLP*", it is required the right context to disambiguate the word "*networks*" from other neural-related terms. Differently from the Transformers that deal naturally with bidirectional contexts (§2.2.4), bidirectional RNNs are based on combining left-to-right and right-to-left flows of the sequences, concretely $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ computed in terms of $\overrightarrow{h}_{t-1}$ and $\overleftarrow{h}_{t+1}$ respectively. By this way, the output of a bidirectional neural network is a combination between $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, typically, the concatenation.

The output of all the recurrent models discussed in this subsection, for a given input sequence $X \in \mathbb{R}^{T \times d}$, is a sequence of the hidden states, $H \in \mathbb{R}^{T \times d_h}$. In order to address many $\rightarrow$ one tasks such as text classification or sentence/document representation learning, it is required to collapse the hidden states in order to obtain a vector representation of the sequence. Some common decisions intended to this aim consist in picking the last hidden state H_T, or perform both unweighted poolings, such as max pooling or average pooling, or weighted poolings like attention mechanisms, that will be discussed in the next subsection.

2.2.3 Attention Mechanisms

When processing sequential inputs, humans tend to focus on the most relevant information to perform a specific task. This process is known as attention and it can be incorporated into neural networks by means of attention mechanisms that compute the relevance of each part of the input in different ways. To understand how the attention mechanisms work is convenient to think about the information retrieval framework. Under that context, there are two distinguishable information sources, a collection of documents $V = \{v_1, ..., v_N\}$ (values) indexed by $K = \{k_1, ..., k_N\}$ (keys) and a given query q. For example, let V be a set of Wikipedia articles, K the set with the titles of each article, and q a query introduced by the user. In order to perform the retrieval of the most similar documents for a given

query q, the similarity between q and each $k_i \in K$ is computed. Then, the documents $v_i : \alpha(k_i, q) > \alpha(k_{j \neq i}, q)$ are returned, where α is a similarity function.

This idea was firstly introduced in the machine learning field by [69, 70] in order to weight the labels according to the similarity of an input and a set of points, for performing linear regression. In that formulation, the output for a new input point $q \in \mathbb{R}$ is defined by $f(q) = \sum_{i=1}^{N} \alpha(q, k_i) v_i$, where $k_i \in \mathbb{R}$ and $v_i \in \mathbb{R}$ are the abscissa and the ordinate of the dataset respectively. By this way, the output for the query q is a weighted sum of the values of all the data points, where the weights are higher if q and k_i are similar in terms of α. More recently, attention mechanisms in modern neural networks were proposed for encoder-decoder models under the context of Neural Machine Translation (NMT), in order to address the bottleneck of collapsing a full source sequence into a fixed-length single context vector. First approaches of encoder-decoder models [71] initialized the decoder with the context vector computed from the last output of the encoder. Thus, the decoder is restricted to work with a summary of the meaning of the whole source sequence. However, different parts of the output typically depend on focusing on specific parts of the inputs, rather than considering the information of the full sequence. Furthermore, it is difficult to collapse all the information of a sequence into a single context vector and typically, information from early elements in the sequence is forgotten in that representation for long temporal dependencies. To address this issue, attention mechanisms were proposed in order to induce shortcuts that allow the decoder to directly focus on specific outputs of the encoder. The general equations of the attention mechanisms for encoder-decoder models are shown in Eqs. 2.3 and 2.4, where they can be seen as a form of weighted average pooling in terms of compatibilities between queries and keys.

$$c_i = \sum_{j=1}^{T} \alpha_{ij} v_j \tag{2.3}$$

$$\alpha_{ij} = \frac{e^{a(q_i, k_j)}}{\sum_{t=1}^{T} e^{a(q_i, k_t)}} \tag{2.4}$$

Under this context, $v_j \in \mathbb{R}^{d_v}$ and $k_j \in \mathbb{R}^{d_k}$ are the hidden state of the encoder at timestep j, $q_i \in \mathbb{R}^{d_q}$ is the hidden state of the decoder at timestep i, $c_i \in \mathbb{R}^{d_v}$ is the context vector seen by the decoder at timestep i, and $\alpha_{ij} \in [0, 1]$ is the normalized attention weight between q_i and k_j computed by means of an attention function a. Several attention functions have been proposed to compute compatibilities between queries and keys such as additive attention [42], general attention, concat attention, dot product attention [43] and scaled dot product attention [11].

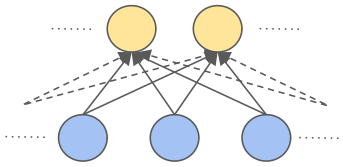

Fig. 2.5 Dependencies modeled by the attention mechanism.

The dependencies modeled by this attention mechanisms can be seen in Figure 2.5. The approaches, where queries and keys are computed from different information sources (decoder and encoder respectively in the encoder-decoder context) are known as cross-attention mechanisms. If the queries and keys come from the same source of information, it is known as self-attention. Furthermore, if the attention is performed on the full sequence of hidden states is called global attention, while if it is performed on some specific hidden states is called local attention [43].

However, encoder-decoder models are usually not well suited for some tasks like text classification, as it is required to understand the sequence instead of generating new text. In this setup, a vector representation of the sequence is computed by means of an encoder, and used in order to perform the classification, either obtained from the last hidden state or by means of pooling (including attention mechanisms as weighted average pooling). The capability of focusing on relevant elements of the sequence is also interesting for classification tasks, and it can be also incorporated by means of attention mechanisms. Differently from the encoder-decoder context, where the attention is computed from the point of view of the decoder, for text classification the attentions are typically computed only from the encoder [2]. This kind of attention is known as self-attention. It is also worth mentioning the location attention [40, 43, 72], where the attention function a is a learnable function that only depends on the queries q_i, thus, attention weights are not computed by means of a compatibility function between two inputs, instead, they are computed individually for each hidden state q_i. We used extensively this attention function along the experimentations with RNNs, in the form shown in Eqs 2.5, 2.6 and 2.7.

$$c = \sum_{i=1}^{T} \alpha_i v_i \tag{2.5}$$

$$\alpha_i = \frac{e^{a(q_i)}}{\sum_{t=1}^{T} e^{a(q_t)}} \tag{2.6}$$

$$a(q_i) = \varphi(w q_i^\top + b) \tag{2.7}$$

where $c \in \mathbb{R}^{d_h}$ is the vector representation of the sequence, $v_i \in \mathbb{R}^{d_h}$ and $q_i \in \mathbb{R}^{d_h}$ are the hidden state of the encoder at timestep i, $w \in \mathbb{R}^{d_h}$ is the weight vector of the attention function and $b \in \mathbb{R}$ is the bias. In this case, the weight vector of the attention function can be seen as a

[2] We assume the existence of only one source sequence, in case of considering multiple information sources, cross-attention mechanisms can also be used for text classification tasks.

high-level representation of a fixed query "what is the informative word" [41]. This was the basis of almost all the works with attentional RNNs seen in the text classification literature, until the Transformer era. One of these works, that is very relevant to this thesis, are the Hierarchical Attention Networks (HAN) [41] we used for extractive summarization (§5.1). HAN were proposed for document classification, considering the structural dependence of the words to compose sentences, and the sentences to compose documents. Roughly, they consist in computing a document vector representation as a weighted average of their sentence vector representations, by using attention mechanisms at sentence level. And, at the same time, compute each sentence vector representation as a weighted average (also using attention mechanisms) of its words. Eqs. 2.8 and 2.9 show how the vector representation, c, of a document is computed from the representations of its sentences, c_i (Eq. 2.8), and how the vector representation of each sentence i is obtained from the word representations of its words (Eq. 2.9).

$$\begin{aligned} c^s &= \sum_{i=1}^{T} \alpha_i^s v_i^s \\ \alpha_i^s &= \frac{e^{a(q_i^s)}}{\sum_{t=1}^{T} e^{a(q_t^s)}} \\ a^s(q_i^s) &= w^s \varphi(W^s q_i^s + b^s)^\top \\ q_i^s = v_i^s &= \mathrm{RNN}^s(c_1^\omega, ..., c_T^\omega)_i \end{aligned} \tag{2.8}$$

$$\begin{aligned} c_i^\omega &= \sum_{j=1}^{P} \alpha_{ij}^\omega v_{ij}^\omega \\ \alpha_{ij}^\omega &= \frac{e^{a(q_{ij}^\omega)}}{\sum_{p=1}^{P} e^{a(q_{ip}^\omega)}} \\ a^\omega(q_{ij}^\omega) &= w^\omega \varphi(W^\omega q_{ij}^\omega + b^\omega)^\top \\ q_{ij}^\omega = v_{ij}^\omega &= \mathrm{RNN}^\omega(e_{i1}, ..., e_{iP})_j \end{aligned} \tag{2.9}$$

where T is the number of sentences in the document, P is the number of words in each sentence, the superscripts s and ω denotes sentence-level and word-level computations respectively, e denotes word embeddings and $\mathrm{RNN}(x)$ denotes the sequence of hidden states given x as input. It is a straightforward extension of the Eqs. 2.5, 2.6 and 2.7, in order to work both at sentence and word levels. The dependencies computed by HAN are shown in Figure 2.6.

Also note that, the attention function a uses an additional matrix ($W^s \in \mathbb{R}^{d_h^s \times d_q^s}$ and $W^\omega \in \mathbb{R}^{d_h^\omega \times d_q^\omega}$ for sentence level and word level respectively) to project the hidden states previously to compute the attention weights. For more details about HAN, we refer the reader to the reference paper [41] and to §5.1.1.

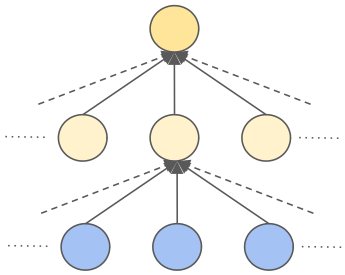

Fig. 2.6 Dependencies modeled by HAN.

An important aspect of the attention mechanisms is the interpretability. Nowadays, there is an increasing interest in understanding the behavior of deep models, and the attention weights have been extensively studied for this purpose in recent years. Attention conveniently gives us one weight per element in the sequence, ideally denoting the relevance of that element in a specific task. There is a lot of controversy about the adequacy of the attention mechanisms for explainability purposes, with mixed evidence on whether it can be used to this aim [73–77]. Despite the discussions about the best-ever technique for interpretability, in this thesis, we used extensively attention mechanisms, both integrated into RNNs and as sequence modelers (§2.2.4), and we tried to study them in order to extract useful linguistic knowledge learned by the models when they address downstream tasks such as: distinguishing correct and incorrect summaries for documents, sentiment analysis, and irony detection.

2.2.4 Transformers

As shown in previous subsections, typical approaches for sequence modeling were based on RNNs and CNNs, however, they pose some problems that can be naturally addressed by using self-attention mechanisms as sequence modelers by themselves [11]. In the first case, the sequential computation makes difficult the parallelization on the temporal dimension and it falls when the dependencies in the sequence path are complicated. Furthermore, it is difficult to interpret the relationships among the sequence tokens. In the second case, they are trivial to parallelize, but as it exploits local dependencies it is required to use many layers to consider long-distance dependencies. While CNNs and RNNs require $\mathcal{O}(\log n)$ and $\mathcal{O}(n)$ steps respectively, in order to observe the dependency between any pair of words (maximum path length), self-attention mechanisms are a constant path length solution $\mathcal{O}(1)$, also trivial to parallelize, that can replace sequential computation "completely" and explicitly model all-vs-all relationships among the tokens of a sequence.

The model that only employs attention for modeling sequences is known as Transformer [11]. The Transformer was introduced for NMT as an encoder-decoder model where the encoder performs self-attention, and the decoder is identical to the encoder, with an additional

attention layer to perform cross-attention between the decoder self-attention (q) and the encoder outputs (k and v). In this thesis, we only used the encoder part of the transformer (namely Transformer encoder), which performs self-attention on the queries, keys and values obtained as transformations of the sequence tokens. In its simplified form, this is equivalent to Eqs. 2.3 and 2.4, when $a(q_i,k_j) = \frac{q_i k_j^\top}{\sqrt{d_k}}$ (scaled dot-product), where q_i y k_j (and also v_j) are tokens of the same sequence. However, the attention layers of the Transformer model, rather than only computing the attention once, run a multi-head mechanism through the scaled dot-product attention multiple times in parallel. The purpose behind this is to allow the model to jointly attend to information of different representation subspaces in order to focus on different features and relationships between the tokens of the sequence, for example, negative polarity tokens should be attended by the other tokens when detecting a negative tweet and positive polarity tokens when detecting a positive tweet (§3.1.4). The dependencies computed by the multi-head attention mechanism are shown in Figure 2.7. Also, it is more parallelizable and the dimensionality of k, q, and v can be smaller than in the single-head attention. The multi-head self-attention mechanism is an extension of Eqs. 2.3 and 2.4, where H different attention heads are used, and the output representation of the token i, c_i, is computed as a projection of the concatenation of the H outputs computed by the attention heads, as shown in Eq. 2.10.

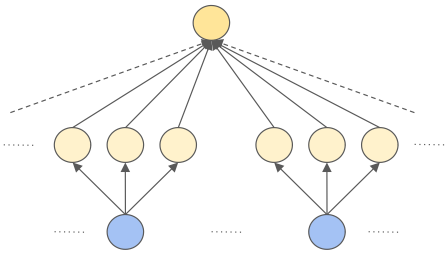

Fig. 2.7 Dependencies modeled by multi-head attention.

$$
\begin{aligned}
c_i &= [c_i^1;...;c_i^H]W^O \\
c_i^h &= \sum_{j=1}^{T} \alpha_{ij}^h v_j^h \\
\alpha_{ij}^h &= \frac{e^{a(q_i^h,k_j^h)}}{\sum_{t=1}^{T} e^{a(q_i^h,k_t^h)}} \\
a(q_i^h,k_j^h) &= \frac{q_i^h {k_j^h}^\top}{\sqrt{d_k}} \\
q_i^h = X_i W_Q^h, &\quad k_j^h = X_j W_K^h, \quad v_j^h = X_j W_K^h
\end{aligned}
\tag{2.10}
$$

where the superscript h is used to index the heads, $X \in \mathbb{R}^{T\times d}$ is the input sequence, $c_i \in \mathbb{R}^{d_h}$ is the computed representation for the token X_i; $W_Q^h \in \mathbb{R}^{d\times d_q}$, $W_K^h \in \mathbb{R}^{d\times d_k}$ and $W_V^h \in \mathbb{R}^{d\times d_v}$ are the projection matrices of queries, keys and values respectively for the attention head h, and $W^O \in \mathbb{R}^{(H\cdot d_v)\times d_h}$ is the weight matrix to compute the final representation from the

concatenated values of the H attention heads. Additionally, the multi-head mechanism can be also applied hierarchically, following the idea shown in Eqs. 2.8 and 2.9. We proposed the Hierarchical Transformer Encoders for extractive summarization in §5.1.2.

Basically, the Transformer encoder (also the decoder, although we have not considered it in this thesis), is a stack of layers composed by two modules: the multi-head attention mechanism and a bottleneck position-wise feedforward network applied on top of the attention mechanism. Furthermore, the outputs of each module are normalized [52] and they are residually connected with its inputs [54]. It is worth noting that the positional information is lost if it is not included with the input sequence. To address this issue several strategies has been proposed like learnable positional embeddings [78] or unparameterized positional encodings [11], both for relative [79] or absolute positions [11, 78]. In this thesis, we only considered the unparameterized absolute positional encoding from [11]. As this model has been instantiated in several sections of this thesis, we refer the reader to the reference paper [11] and to each one of that sections §3.3.2, §3.1.2 and §5.1.2 in order to see specific instantiations for different tasks. Finally, the tradeoff between efficiency and performance of these models has made them very suitable to perform transfer learning after being pretrained in self-supervised ways for text representation learning and then finetuned in downstream tasks. In the following subsection, we discuss this aspect in more detail.

2.3 Text Representation Learning

The featurization process is one of the keys of the machine learning success in NLP. However, it is difficult to manually define the best set of features to address specific NLP tasks with machine learning approaches. Most of the traditional representation approaches were local, in the sense that each component of the representations is only tied with one represented concept. The most known approach is the one-hot representation, where each component corresponds to each term in a vocabulary. Thus, given a vocabulary $\mathcal{V} = \{w_1, ..., w_N\}$, a token w_i is represented by means of a sparse symbolic vector with a single 1 and $N-1$ zeros, $v(w_i)_j = \mathbb{1}_{i=j}$. The one-hot representation can be straightforwardly extended to sequences of tokens by means of summing or intersecting all the one-hot vectors of the sequence. This approach has been one of the most effective and widely used for text classification, and it is known as the bag-of approach. Depending on the constituents, the one-hot representations can represent characters, subwords, words, or even task-specific features like polarity information, either individually or in terms of n-grams if some kind of locality has to be considered. However, this representation suffers from several problems. First, the one-hot representations of all the tokens in the vocabulary are orthogonal, thus,

the representation is not informative enough as every two distinct one-hot vectors are at the same distance, losing the notion of similarity among tokens. Also, as each component of the vector representation is directly tied to one token, the size of that vectors is the same as the vocabulary size, so, for large vocabularies, this representation is not practical. Furthermore, as the representation is not contextual, in the sense that a token is always represented by the same vector, some interesting context-dependent properties of the language, like the polysemy, are lost. All these problems difficult the task of machine learning models, due to they directly learn statistical patterns from the representation of the input, so, the better is the representation, the easier is to learn some specific tasks.

In contrast to local representations, distributed representations can represent many-to-many relationships between concepts and components, in a way that each concept is represented by many components of the vector representation and each component participates in the representation of many concepts. Thus, these representations alleviate the curse of the dimensionality suffered by local representations. The simplest example of distributed representation consists on modeling a vocabulary of size N with binary vectors of $log_2(N)$ components. However, it is difficult to define distributed representations in order to capture linguistic knowledge useful for addressing some specific tasks. Fortunately, deep learning gives us a framework in order to learn distributed representations even from self-supervised learning setups, which has become one of the major breakthroughs in the NLP field during these last years. These distributed representations are known as embeddings, which can be used for representing from characters to full documents in continuous spaces that preserve the notion of similarity. In this thesis we extensively used distributed word representations extracted from neural networks trained for language modeling objectives, so, we will focus only on them in this section.

Many kinds of neural models have been proposed historically for learning distributed representations, mainly for word [57, 68, 80–84] and sentences [85–89]. Feed-forward Neural Network Language Model (NNLM) [90] was the first successful attempt intended to this aim by means of a feed-forward network trained as a probabilistic n-gram language model. They proposed to perform a mapping from any element of $\mathscr{V}$ to a real vector by means of a matrix of free parameters that represent the feature vectors associated with each word in the vocabulary. This idea has been the basis of modern approaches for learning distributed representations. Later, some works intended to modeling variable-length sequences were proposed [91], however the most widely used approaches to learn distributed representations were the Skip-gram and the Continuous Bag of Words (CBOW) models [80]. These two approaches are very similar to NNLM with two major improvements: to propose a novel training objective for language modeling that takes into account additive compositionality,

and to alleviate the bottleneck of NNLM when computing the softmax layer over the entire vocabulary. These two improvements allowed to learn from large corpora, which have shown to be one of the key factors to learn high-quality distributed representations.

However, this kind of approaches are non-contextual, in the sense that each word is always represented by the same vector, independently of its context. The most recent breakthrough was intended to overcome this issue by means of deriving the embeddings from the hidden layers of some kind of neural encoder [57] (and also [68]). BERT [57] is a Transformer encoder trained on two self-supervised objectives: masked language model and next sentence prediction. Many modeling improvements on the vanilla BERT have been proposed [82–84, 92, 93]. For more information about them, we refer the reader to a recent survey that synthesizes the most relevant contributions in the BERT field, in terms of pretraining tasks, efficiency, pretraining data, interpretability, and multilingualism [94]. In this thesis we proposed an adaptation of BERT for the Spanish language and the Twitter domain, that integrates also some aspects of ALBERT [82], RoBERTa [83] and SpanBERT [92]. We will discuss these BERT-like approaches in more detail in the subsection §2.3.2. The most recent works to learn contextualized distributed representations are based on autoregressive Transformers pretrained with denoising tasks such as text infilling, sentence permutation, gap sentence generation, etc., and they are the state of the art in many language generation and comprehension tasks [95, 96]. Although they have been excluded from this document, we are currently working with them for some future works.

All the contextual embeddings discussed here are used under the transfer learning paradigm. Specifically, these neural networks are first pretrained on self-supervised objectives in order to capture general linguistic knowledge from large corpora of raw text, and later they are finetuned specifically on the downstream task. Formally, a neural network (typically a Transformer), $f_\theta : X \rightarrow y$ where X is the noisy/masked input and y is the expected reconstruction, is trained to minimize the loss for the reconstruction objective, $\arg\min_\theta \mathscr{L}_{LM}(\theta, \mathscr{D})$, where $\mathscr{D}$ is the raw text dataset and $\mathscr{L}_{LM}$ is the loss function for pretraining. After the pretraining, a task-specific finetuning is made by jointly learn θ and θ_T, $\arg\min_{\theta,\theta_T} \mathscr{L}_T(\theta, \theta_T, \mathscr{D}_T)$, where θ_T are the new parameters for the task, $\mathscr{L}_T$ is the loss function for finetuning and $\mathscr{D}_T$ is the dataset for the downstream task.

In the following subsections, we describe in more detail the embeddings we used in this work. On the one hand, non-contextual embeddings obtained by means of the Skip-gram model §2.3.1 and, on the other hand, contextual embeddings obtained from Transformer encoders §2.3.2.

2.3.1 Non-Contextual Embeddings

Non-contextual embeddings, in comparison to the contextual embeddings we will discuss in the following subsection, represent each token always with the same vector, independently of its context. This is because these approaches collapse the information of all the contexts where a token appears in only one single vector. So, the notion of context is integrated into this single representation following the principles of distributional semantics: "you shall know a word by the company it keeps" [97], basically, words that occur in similar contexts tend to have a similar meaning. In order to capture this, most of the non-contextual approaches record word co-occurrence by means of sliding windows over large corpora. In this subsection we briefly describe the non-contextual approach we used in this thesis: the Skip-gram model [80], that was released under the Word2Vec framework together with CBOW and several techniques in order to improve the training efficiency on large corpora.

The Skip-gram model is formalized as follows. Let X be a sequence of tokens $X = \{x_1, ..., x_T\}$ such that x_t is the one-hot representation of the word at position t, and f_θ be a fully connected one hidden layer network with a softmax output layer, where $\theta = \{W, U\}$, $W \in \mathbb{R}^{d \times |\mathscr{V}|}$ and $U \in \mathbb{R}^{|\mathscr{V}| \times d}$. The objective of the Skip-gram model is to learn the parameters θ, in order to minimize the negative average of the log probabilities of the surrounding tokens $\{x_{t-k}, x_{t-k+1}, ..., x_{t+k-1}, x_{t+k}\}$ of each token x_t given a sliding window of size k. This objective is formalized in Eq. 2.11 for training with a single sample X.

$$\mathscr{L}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-k \leq j \leq k \\ j \neq 0}} \log p_\theta(x_{t+j} | x_t) \tag{2.11}$$

It is worth noting that, to train this model, typically a set of pairs (x_t, x_{t+j}) is built for the context surrounding on a word x_t, extracted with the sliding window of size k. Given such set of pairs of words, the network has to compute $p_\theta(x_{t+j}|x_t)$ and sum them for all the pairs where x_t is the anchor, in order to compute the loss function and backpropagate the gradients to update θ. Regarding the posterior probability modeled by Skip-gram, the basic formulation defines $p_\theta(x_{t+j}|x_t) = softmax(UWx_t)_{i(x_{t+j})}$, where $i(x)$ refers the index of the token x in the vocabulary. However, the computation of the softmax on the entire vocabulary requires to evaluate $\mathscr{V}$ output nodes, which is not efficient for training on large vocabularies. Instead, two strategies

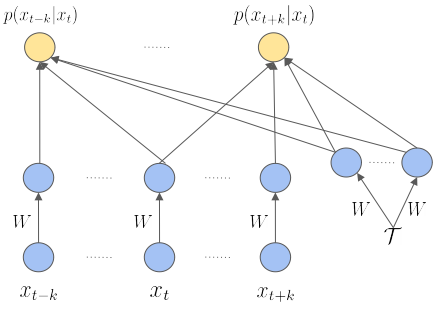

Fig. 2.8 Skipgram model with negative sampling.

were proposed to address this issue: hierarchical softmax and negative sampling [80]. On the one hand, hierarchical softmax aims at efficiently approximate the full softmax, with $\log_2(|\mathscr{V}|)$ steps, by means of a binary tree representation of the $|\mathscr{V}|$ components of the output layer. On the other hand, negative sampling approximates the posterior by means of contrastive estimation using a set of negative tokens $\mathscr{T}$ that do not pertain to the surrounding context of the token x_t, as shown in Eq. 2.12. Also, Figure 2.8 shows a general picture of the skip-gram model with negative sampling.

$$p_\theta(x_{t+j}|x_t) \approx \frac{(Wx_{t+j})^\top (Wx_t)}{(\sum_{x' \in \mathscr{T}} (Wx')^\top (Wx_t)) + (Wx_{t+j})^\top (Wx_t)} \tag{2.12}$$

Once the model is trained following the objective of Eq. 2.11, the weight matrix W can be seen as a lookup table from which the embedding of a token x can be extracted. This is one of the main differences with respect to the models for computing contextual embeddings, that cannot be interpreted as token-only lookup tables. In this thesis, we used the Skip-gram model in almost all the works, except chapter 4 where we introduce contextual representations for Spanish tweets. In all these works, we pretrained Skip-gram models with Spanish tweets and news articles, depending on the addressed task, using negative sampling as a method to overcome the softmax bottleneck.

2.3.2 Contextual Embeddings

In recent years, the Natural Language Processing community have been moving from non-contextual representations [80, 81] towards contextual ones [57, 68, 82–84, 98]. In the first case, each token is represented by one embedding that condenses information of all the contexts where that token appears. While in the second case, each word is represented by different embeddings depending on the context of the word e.g., the representation of the token *mouse* is different in the sentences s_1 ="The trackball mouse works perfectly", s_2 ="The mouse was caught by the cat", s_3 ="The mouse of my laptop is failing". However, the representations of *mouse* in s_1 and s_3 should be more similar than with respect to s_2 where the meaning of *mouse* is different. So, these approaches can naturally model complex features of the tokens depending on specific contexts such as polysemy, coreference, etc. The approaches intended to compute contextual representations derive them from the hidden layers of some kind of neural encoder applied on sequences of tokens, and they are pretrained and then finetuned on downstream tasks.

Among the approaches for computing contextual representations [57, 68, 82–84, 95, 96], in this thesis, we focused on BERT. BERT [57] was the first work intended to pretrain a Transformer model on large corpora by means of two pretraining objectives: Masked Language

Model (MLM) and Next Sentence Prediction (NSP). On the one hand, MLM is basically a cloze task where random tokens are masked, forcing the model to use the bidirectional context of a given masked token to predict it. This objective, along with the Transformer encoder, allows BERT to naturally model bidirectional contextual representations. On the other hand, the NSP signal was proposed with the aim of learning the coherence by means of a binary classification which consists in determining if a text segment A precedes a text segment B in the source.

BERT can be formalized as follows. Let $A = \{a_1, ..., a_T\}$ and $B = \{b_1, ..., b_P\}$ be sequences of tokens (sub-words in this case), f_θ a Transformer encoder with two output layers: one softmax layer for the MLM objective (f_θ^{MLM}) and a sigmoid layer for NSP (f_θ^{NSP}), and $X = \{\text{CLS}, a_1, ..., a_T, \text{SEP}, b_1, ..., b_P, \text{SEP}\}$ the concatenation of A and B, where CLS (x_1) is a special symbol for classification output, and SEP (x_{T+2} and x_{T+P+3}) is the special symbol to separate non-consecutive token sequences and to delimit the input. In order to train the model, a set of tokens of X is randomly masked in three different ways: by the special symbol MASK (80% of the times), by a random token (10%) or non-masked (10%). So, let $\Gamma = \{\gamma_1, ..., \gamma_K : \gamma_i \notin \{1, T+2, T+P+3\}\}$ the set of indexes of the masked tokens in X, where K is the number of masked tokens, the objective for training f_θ^{MLM} on the MLM objective is as stated in Eq. 2.13.

$$\mathscr{L}_{MLM}(\theta) = -\frac{1}{K}\sum_{k=1}^{K} \log p_\theta(x_{\gamma_k}|X_{-\Gamma}) \tag{2.13}$$

where x_{γ_k} denotes a masked token of X, $X_{-\Gamma}$ denotes the sequence X with the tokens indexed by Γ masked and $p_\theta(x_{\gamma_k}|X_{-\Gamma}) = f_\theta^{MLM}(X_{-\Gamma})_{\gamma_k}$. In addition to the MLM objective, the NSP objective is used to learn coherence between A and B. NSP can be formalized as follows. Let $y = \delta_{B\to A}$ evaluated to 1 if B precedes A in the original source or to 0 otherwise, and $\hat{y} = f_\theta^{NSP}(X_{-\Gamma})_1$ is the output for the CLS token, the NSP objective is shown in Eq. 2.14.

$$\mathscr{L}_{NSP}(\theta) = -((y \log \hat{y}) + (1-y)(\log(1-\hat{y}))) \tag{2.14}$$

In order to train following the NSP objective, pairs of A, B sentences are sampled, where 50% of the time B is the next sentence that follows A in the source, and the remaining 50% of the time B is a random sentence sampled from the corpus. A schematic picture of the BERT model can be seen in 2.9. It is worth noting that, for the sake of clarity, we have not considered in the formulation neither the positional embeddings nor the segment embeddings

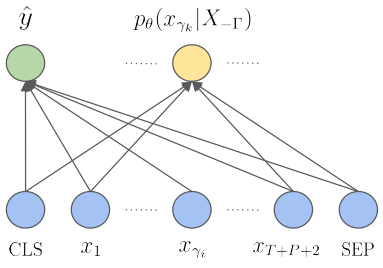

Fig. 2.9 BERT schema.

[57], however, they are mandatory to inject positional
information and to distinguish the sentence to which
each token pertains. BERT is typically used for language understanding tasks by, first, pretraining on large corpora, following Eqs. 2.13 and 2.14 in order to capture linguistic knowledge, and second, transferring this knowledge to downstream tasks by means of finetuning. Eqs. 2.15 and 2.16, show both steps, where $\mathscr{D}$ is the dataset for pretraining, θ_T are the task-specific parameters added to the model f_θ and $\mathscr{L}_T$, and $\mathscr{D}_T$ are the loss function and the dataset for the finetuning step respectively.

$$\arg\min_{\theta} \mathscr{L}_{MLM}(\theta, \mathscr{D}) + \mathscr{L}_{NSP}(\theta, \mathscr{D}) \tag{2.15}$$

$$\underset{\theta, \theta_T}{argmin}\ \mathscr{L}_T(\theta, \theta_T, \mathscr{D}_T) \tag{2.16}$$

Typically, in the finetuning step (Eq. 2.16), all the parameters of the pretrained Transformer encoder are jointly updated together with the task-specific parameters, however, depending on the properties of the downstream task, it should be more convenient to update only some specific layers, or even freeze all of them (basically removing θ from Eq. 2.16). In the latter case, BERT is used as a feature-based approach, similar to the non-contextual models discussed in the previous subsection, where the representations obtained from BERT are immutable and the greatest modeling burden falls on the task-specific parameters applied on top of them.

BERT and several variants of its underlying structure are the state of the art for learning contextual representations that are useful in many NLP tasks. We based some works of this thesis (chapter 4) on some variants of the BERT architecture, improving it in several directions [82, 83, 92]. On the one hand, the benefits of the NSP signal, as defined in [57], have been a controversial topic in the literature [92, 99], as it is easy to distinguish randomly sampled *next sentences* only focusing on the topic instead of inter-sentence coherence [82]. However, sentence coherence is an important aspect for language understanding. To better model it, the SOP signal was proposed on the AlBERT model [82]. SOP is a reformulation of NSP where pairs of unordered sentences are used to force the model to learn inter-sentence coherence instead of topic coherence as induced by NSP. Regarding the MLM objective, SpanBERT [92] proposed several span masking strategies, using a span boundary objective for predicting each token in a masked span using the tokens on its boundary. Finally, RoBERTa [83], is a BERT model with a careful design of its hyper-parameters, training corpora and practical strategies. The main novelties of RoBERTa were: not considering the NSP signal, a dynamic masking strategy, instead of defining a single masking pattern for each sample, and training

with large batches, that shown to improve the perplexity in the MLM objective as well as the performance on downstream tasks. It is worth saying that, nowadays, the modeling of contextual representations by means of “BERT-like" models is constantly evolving in an unprecedented career inside the NLP field, and, we only considered in this thesis a very brief part in which we focused. So, for more information, we refer the reader to §4.1 and to a recent survey that synthesizes the most relevant contributions in the BERT field, in terms of pretraining tasks, efficiency, pretraining data, interpretability, and multilingualism [94].

The main challenges addressed in this thesis, related to the BERT models, were the target domain and language pretraining. Although the pretrained parameters are publicly released for most of the BERT-like models, their target are general domains and the English language, which makes it difficult the application in non-standard setups e.g., language understanding with Spanish tweets. Concretely, we propose TWilBERT, a specialization of BERT for both the Spanish language and the Twitter domain, that leverages successful modifications of the BERT architecture [82, 83, 92].

Chapter 3

Text Analytics in Social Media

To understand the current impact of social media platforms, we have to go back to two key dates in human history: 1969 and 1991. In 1969, the first interconnected network (ARPANET), established the first connection between four different nodes (University of California, Los Angeles, the Stanford Research Institute, University of California, Santa Barbara, and the University of Utah) to communicate several academic and state institutions. This allowed for better coordination of scientific projects and better organization of military research projects, in a decentralized way. This way, attending to the definition of social media as the set of computer-aided technologies that facilitate the creation of sharing information, ideas, and other forms of expression via virtual communities and networks [100], the objective of ARPANET can be seen as a precursor of the social media idea, where the virtual community was constrained to a set of universities, and the information and the ideas discussed were about specific research projects. With the aim of extending the size of virtual communities, from ARPANET, other interconnected networks were proposed as Usenet, Eunet and the widely known and established nowadays, Internet, that allowed to build a single logical network of global scope, thus allowing the emergence of potentially infinite virtual communities. However, although the development of general-purpose computers grew exponentially during the 1980s, these interconnected networks were not useful for a large part of the population due to the difficulty of searching and accessing the content available on the Internet. This was the case until, in 1991, the development of the World Wide Web became public, which allowed the distribution of hypermedia documents (texts, images, videos, or other multimedia content) easily accessible through the Internet by means of web browsers, thus facilitating the navigation between these documents.

However, the initial philosophy of document consumption by the public was offline, that is, some users created static content that could be viewed by other users on the network, but without interaction between content creators and consumers, and even no interaction

between consumers, which made it difficult to emerge virtual communities of users. This philosophy was the mainstream until the first half of the 2000s, where the majority of social and instant messaging applications such as Internet Relay Chat, Messenger, or specialized forums, grown rapidly in terms of users, due to the popularization of the Internet and the World Wide Web. In these applications, the offline consumption philosophy disappeared to give way to simultaneous interactions among users, allowing the emergence of virtual communities whose users discuss personal or general-purpose issues. However, despite the continuous increase in Internet use since then, most of these instant messaging and social applications have stagnated, suffering a large decline in the number of users since the 2010s. This decline coincided with the popularization of other types of platforms: social media platforms, which have allowed the emergence of virtual communities as large as entire countries, for example, Facebook, with 2.7 billion users, would be the most populous country in the world, above China.

Therefore, social media platforms have been established, throughout the 2010s, as the predominant platforms for sharing and discussing content, as shown in Figure 3.1. Currently, there is a great diversity of this type of platform, and depending on the type of content they have specialized in different market segments such as video sharing and discussion (YouTube), biography and experiences (Facebook), microblogging (Twitter), social news aggregation (Reddit) or photo and video sharing (Instagram). Despite the differences among the content typologies, all the social media platforms share common aspects, which have been key factors in their explosion. Some of these common aspects are: the users can create specific profiles that are their virtual reflection in a global community, the platforms facilitate interactions among similar users to other similar individuals or communities, the contents such as text, photos, or videos are generated through all online interactions, and the access to these media platforms is carefully designed by means of web-based applications or mobile device applications in order to be present during all the daily life of the users.

These platforms have had, during the last years, an overwhelming impact on the daily life of individuals and organizations. On the one hand, media platforms have been extensively used: by organizations such as governments to interact with citizens and to monitor the public opinion; by businesses to perform marketing research, communication, or customer segmentation; by organizations to analyze personality traits of recruitment candidates; by schools to evaluate admissions; and even in criminal investigations to assist searches for missing people. On the other hand, individuals use social media platforms daily, as news sources, as social tools to interact and keep up with their community, or as a self-presentational tools to influence others by carefully managing their self-image or virtual identity in social contexts.

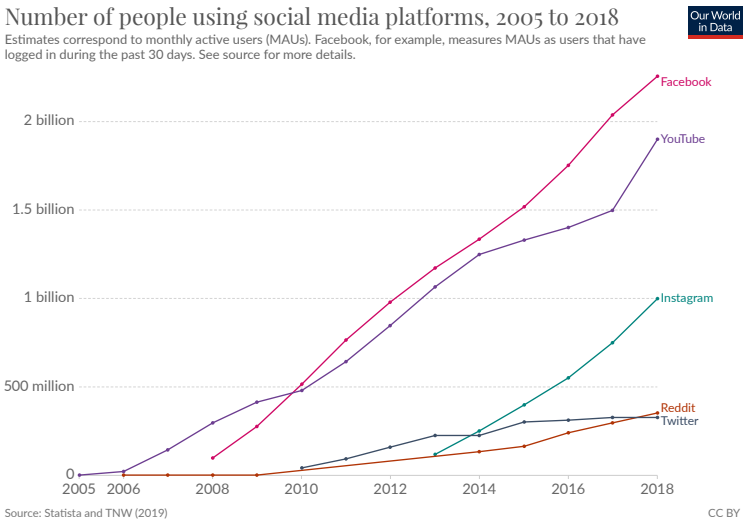

Fig. 3.1 Number of people using social media platforms (Facebook, YouTube, Twitter, Reddit and Instagram) since 2005 to 2018. Source: Statista and TNW (2019) `https://ourworldindata.org/rise-of-social-media`.

Individuals tend to express their opinions, on these media platforms, in a straightforward/spontaneous way, like in real social contexts. This is an important aspect because now, we can study social problems by focusing on the population that habit in the media platforms. However, it is necessary to highlight that this population does not exactly represent the world population. Instead, it is a biased subset where a great disparity is present, such as: people with medium/high **socioeconomic status**, an **extreme polarization** about issues of general interest such as politics or sports, **stereotypes** towards aspects of individuals such as culture or religion, knowledge and use of specific **jargons** of virtual social environments such as hashtags, emoticons and abbreviations, large presence of **young population** compared to older populations, ubiquitous presence of automated content generation profiles (**bots**), not all users **contributes equally** in terms of generated content[1] (10% of users generate the 80% of the content, following approximately the Pareto principle), and extreme **echo chamber** effects. In spite of these biases, social media platforms have posed a very interesting and populated environment to analyze individuals and communities. Especially Twitter, in which we focus on the development of this thesis, is the social network where the largest number of discussions on controversial issues take place. Differently from other social networks like Facebook, whose objective is typically to socialize with known communities of friends and family, or Instagram, that seeks for mere visual pleasure of users, the inherent nature of

[1] `https://www.pewresearch.org/internet/2019/04/24/sizing-up-twitter-users/`

Twitter encourages the searching for information and concise discussions, through limited-length texts of 280 characters (commonly known as tweets), about relevant issues, in specific communities within a global community (e.g., Twitter is the social network with the largest presence of political opinion leaders).

In this thesis, we focus on addressing text classification problems on Twitter, with a special focus on sentiment analysis, as well as on other problems that influence it, such as emotions and irony. It should be noted that we have not considered additional information to the text such as images, videos, and audio, but we consider that it is an important future line of research to gain full understanding in these fields. Furthermore, we also explored, although less extensively (and, in some cases, being excluded from this document), other text classification problems in Twitter such as: stance detection, to analyze the stance of the users regarding controversial topics, topic classification, to detect topics of interest for specific applications; or hate speech detection, intended to detect hate among users, typically fostered by extreme polarization and stereotypes.

This chapter is organized as follows. In §3.1, we discuss deeply the sentiment analysis problem and our proposal based on transformer encoders, along with an extensive experimental study. In §3.2, our work on the emotion detection field is discussed, where we propose a novel way to optimize evaluation metrics in large sets of emotional classes. Finally, we present our work in irony detection in §3.3, where we proposed a transformer encoder model and we deeply study it to interpret its connection with sentiment analysis and the ironic features learned by the system.

3.1 Sentiment Analysis

Sentiment analysis is the research field devoted to understand the underlying sentiment of subjective opinions communicated by humans in social environments. These opinions are typically related to events, individuals, products or services offered by enterprises, organizations, etc. [101], and this was the key that made the investigation of sentiment analysis flourish: its potential applications in the industry. Also, the overwhelming expansion of individuals and organizations in social networks was a key factor, thus growing the necessity to extract insights from the users' opinions in this kind of platforms. The interest in sentiment analysis is not focused on a specific industrial sector, but rather it is scattered across all the industry, being used in a plethora of relevant environments such as: political tendency identification [102], user experience evaluation [13, 14], consumer confidence and political opinion [103], election prediction [104], prediction of box-office revenues for movies [105] or stock market prediction [106]. Although sentiment analysis as discussed

in this introduction is focused in sentiment classification, it is convenient to highlight that there are many other research lines where sentiment is one of the fundamental pillars, and almost all of them remain unexplored, due to the targeting of the research community in the sentiment classification. Among these promising research lines are: sentiment reasoning or sentiment-aware language generation [107].

A historical taxonomy of sentiment analysis tasks [101] consists in distinguishing two granularities: document-level and sentence-level. On the one hand, document-level refers to understanding the whole sentiment in a document, assuming that the document is only related to a single entity, and, potentially, it mentions several aspects of the entity. On the other hand, sentence-level is focused on understanding the sentiment of sentences that convey opinions (potentially inside a document). To do this, at first is required to detect sentences that convey opinions (both subjective and objective, as some objective opinions can also convey sentiment more subtly than the subjective ones) and, later, to analyze the polarity of these salient sentences. In both cases, the goal is to understand the overall sentiment, however, both in document-level and in sentence-level, the texts can mention different entities, and different aspects of each entity, with different sentiment towards each one. Therefore, both granularities are implicitly assuming that the texts convey a single overall sentiment for all the entities and their discussed aspects. In order to tackle this lack of specialization, aspect-level sentiment analysis was considered [108], as a fine-grained analysis, that can be integrated into both the previous approaches, to characterize the sentiment towards each entity and their aspects/features. Figure 3.2 shows an example from the review of RoBERTa paper [83], where the negative sentences are written in red, positive sentences in green and neutral sentences in gray. It can be observed the sentiment at document-level (that matches the final decision), at sentence-level, and for some specific aspects of the paper such as experimental evaluation, significance, and novelty.

Decision: Reject [**Overall sentiment**]
Comment: This paper conducts an extensive study of training BERT and shows that its performance can be improved significantly by choosing a better training setup (e.g., hyperparameters, objective functions) [**Experimental evaluation**]. I think this paper clearly offers a better understanding of the importance of tuning a language model to get the best performance on downstream tasks [**Significance**]. However, most of the findings are obvious (careful tuning helps, more data helps). I think the novelty and technical contributions are rather limited for a conference such as ICLR [**Novelty**]. These concerns are also shared by all the reviewers. The review scores are borderline, so I recommend to reject the paper.

Fig. 3.2 Example from the review of (one of the most used BERT-like models) RoBERTa (`https://openreview.net/forum?id=SyxS0T4tvS`). The green sentences convey positive sentiment, the red ones convey negative sentiment and gray refers to neutral sentiment.

A formal definition of the opinions, that are the main source of analysis in sentiment analysis tasks is given in [101]. Following [101], an opinion can be defined in terms of a quintuple $(e_i, a_{ij}, o_{ijkl}, h_k, t_l)$ where e_i is the entity, a_{ij} is an aspect related to the entity e_i, h_k is the opinion holder, t_l is the timestamp when the opinion was emitted and s_{ijkl} is the sentiment expressed by the author h_k about the aspect a_{ij} of the entity e_i with timestamp h_k. Therefore, the broader objective of the sentiment analysis problem is to detect all the quintuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ in a given document $\mathscr{D}$. From this broad definition of the objective, we can identify the components of the previous taxonomy e.g., document-level sentiment analysis can be viewed as collapsing the sentiment for all the aspects of all the entities mentioned to obtain (e, s_{kl}, h_k, t_l), and sentence-level can be interpreted in the same way than document-level but for each sentence in $\mathscr{D}$.

The sentiment s_{ijkl} can be modeled in different ways. The most common approach consists on using a discrete taxonomy of sentiments: negative, positive and neutral (that typically means no sentiment expressed). Also, in some works, the neutral class is considered with different meanings, and it is split into two different classes [2, 109–112]: neutral and none. In these cases, the term neutral refers to the neutralization of positive and negative sentiments (both expressed with the same intensity) while the term none means no sentiment expressed. These discrete classes can be extended to consider different intensities of the sentiment e.g., strong negative, negative, neutral, positive, and strong positive. Outside of the discrete taxonomy, the sentiment intensities can also be studied, in a more fine-grained way than the previous approach, by constraining them to some continuous interval. It is convenient to highlight that, these taxonomies are oversimplifications of the sentiment analysis task, and as most of the works on sentiment analysis work under them, performance saturation has been reached in these research directions [107].

As stated before in this section, a very interesting environment to work with sentiment analysis is Twitter, where users straightforwardly express their opinions by means of, typically, short sentences (tweets). These tweets present some specific nuances that do not occur in sentiment analysis for normative text and increase the complexity of the task, such as lack of context, due to the limited length of the tweets (originally 140 characters and 280 since November, 2017), use of informal language, abbreviations, spelling mistakes, elongated words, emoticons, user mentions, hashtags, etc. Furthermore, we explore sentiment analysis in Spanish tweets that poses an additional complexity due to the lack of massive high-quality language-specific resources. Despite the Spanish language is the second most natively-spoken language in the world (being the official language in 21 countries) and the third most used

in Twitter[2], the NLP research for the Spanish language is, by far, not as extensive as for the English language, and it is typically limited to following in the wake of advances in the English language. Furthermore, as the field of sentiment analysis is underexplored in the Spanish language[3], the performance of the systems in Spanish corpora has also reached a saturation point, like for the English corpora [107], but in a much lower performance bound [2, 111, 112]. For this reason, we encourage to continue researching on this direction and studying other languages that remain underexplored.

Although some works from the English research community were intended to propose corpora for sentiment analysis in Spanish [113], the Workshop on Sentiment Analysis (TASS) at the Spanish Society for Natural Language Processing (SEPLN) has been the reference workshop that boosted the research in this field, by proposing corpora intended to this aim since 2012 until our days. In all the editions of the workshop, document-level sentiment analysis (called sentiment analysis at global level in the workshop) has been proposed. Only in TASS 2013, sentiment analysis at entity level was proposed, on a semi-automatically annotated corpus of 68.000 Twitter messages (General-Corpus), written in Spanish by about 150 well-known personalities and celebrities of the world of politics, economy, communication, mass media, and culture, and collected between November 2011 and March 2012. Since TASS 2014 [114] until TASS 2018 [112], aspect-based sentiment analysis was proposed on two human-annotated corpora of tweets: Social-TV, composed of 2773 tweets about football and collected during the 2014 Final of Copa del Rey championship in Spain, and Spanish Tweets for Opinion Mining at aspect level about POLitics (STOMPOL), composed by 1284 tweets related to political aspects about the political campaign of the 2015 Spanish general elections. All the editions of TASS until 2017 [111], considered the General-Corpus for sentiment analysis at document-level, however, although it was very useful in the first years, the methodology to build the corpus is open to criticism due to only the training set (10%) was manually annotated, and the gold standards of the test set were generated by pooling the outputs of all the participants of the TASS 2012 and manually reviewing ambiguous decisions after applying a voting schema on these outputs. For this reason, in TASS 2017, the InterTASS corpus was presented [111], which is composed of 3413 manually annotated Spanish tweets collected from July 2016 to January 2017. In subsequent years InterTASS was extended [2, 112, 115], by considering corpora for different Spanish variants: Costa Rican, Peruvian, Mexican, Uruguayan, and Spanish (from Spain). It is especially

[2] `https://www.vicinitas.io/blog/twitter-social-media-strategy-2018-research-100-million-tweets#language`

[3] A search in Google Scholar with the terms: "sentiment analysis $LANGUAGE$" can be useful to see a broad estimation for the amount of research in some languages. Following this, we show the number of works for a subset of the most spoken languages: 1.290.000 references for English, 397.000 for Chinese, 272.000 for Spanish, 235.000 references for Japanese, and 75.700 for Arabic.

interesting, as all these Spanish variants exhibit a large amount of lexical and even structural differences [2], thus showing the necessity of developing language-specific resources for them, and also broadening the scope of interest also for Latin American researchers and industry. Although the methodology followed for collecting the tweets of the General-Corpus also considered the diverse nationality of the authors, it was implicitly (all together with no distinction), so no different sets for each language were considered.

Now we will discuss the most recent works on the TASS workshop (a deeper study of all the approaches presented to the TASS competitions until 2016 can be seen in [116]). Until 2016, the predominant approaches were based on traditional supervised approaches, using classifiers like logistic regression, SVM, and Naive Bayes, applied on top of syntactic and stylistic features combined with task-specific resources like polarity lexicons, while only one neural-based approach was proposed [117]. Much effort was dedicated across these years to build lexicons for Spanish [118–121], however, its use has declined over time due to the increase in the quality of representations, typically based on word and sentence embeddings, and they are nowadays only used in combination with deep learning approaches to induce linguistic knowledge in them. It was not until 2016 when the first system based on word embeddings was presented [122] (3 years after the explosion of the word embeddings in the English research community [80]), that used the Spanish Billion Words Corpus and Embeddings [123]. Interestingly, until 2016, although the results obtained by using word embeddings and some neural-based approaches were promising, they have not surpassed the best results of the competitions, which typically had been achieved using SVM and handcrafted features. This was so until 2017, with the new InterTASS corpus, where our system based on Deep Averaging Networks on top of in-domain word embeddings trained with 87M of Spanish tweets [18] was the best system of the competition. Also, the second best-ranked system was based on CNN, using word embeddings trained with a collection of documents composed of news, Wikipedia articles, subtitles, etc. [124]. From 2017, deep learning techniques were established as a *de facto* standard, being used by 80% of the participants in TASS 2018, and by all of them in TASS 2019 and TASS 2020. The most relevant research shift, since then, was made since 2019, with the emergence of approaches based on finetuning multilingual pretrained bidirectional language models for language understanding [57]. The first proposal based on them, for the TASS workshop, was made in [125]. However, this approach performed worse than our system, which is also based on the Transformer Encoder (backbone architecture of BERT), but trained from scratch with the TASS corpora by using non-contextual pretrained Twitter word embeddings for Spanish as inputs [16]. This showed the need of developing language-specific pretrained models, and

the competitive behavior of the self-attention mechanisms to compute word representations, also for the Spanish language [126, 127].

In addition to the TASS, that is the reference workshop for training and evaluating sentiment analysis approaches in Spanish, many other works explore sentiment analysis in other environments where the Spanish language is used: opinion analysis of microblogging data [128], deception detection [129], user experience evaluation [13], voting intention inference [130], marijuana infodemiology [131], financial analysis [132], early detection of infectious diseases [133], analysis of medical opinions [134], analysis of user reviews about restaurant and hotels [135], products [136], and analysis of Spanish online videos [137].

It is worth noting that, although we are still far from achieving results similar to those obtained in English on sentiment analysis reference corpora, we also reached a saturation point in this workshop [107]. Furthermore, the recent decline in participation and effort devoted to the sentiment analysis task, for the Spanish language, exacerbates these problems. However, there are other factors that affect directly the sentiment analysis, and they can be an interesting research line to address this saturation and increasing the performance. Among these factors, we highlight the emotions, the negation, and the figurative language like irony or sarcasm, which are commonly known (in the English literature) for degrading the performance of the sentiment analysis systems that are not able to tackle these aspects [138–145]. These aspects have not been extensively explored for the Spanish language, therefore, we consider that they are interesting lines of research to study their relationships with sentiment analysis and to improve the performance of the sentiment analysis systems when they deal, for example, with ironic tweets (which is one of the rhetorical devices most used in social networks to convey non-literal meaning). For this reason, some of the work done in this thesis is intended to cover these aspects, in spite of it was not possible to study deeply their relationships due to the lack of high-quality annotated resources.

In this subsection, we focus on document-level sentiment analysis with Spanish tweets. We propose the use of Transformer encoders, trained from-scratch on the downstream task, on top of pretrained word representations computed from Spanish tweets. To evaluate the adequacy of our proposal, we performed an extensive experimentation on Task 1 of the TASS 2019 workshop for several Spanish variants, where our system obtains very competitive results, being the best-ranked system in 3/5 Spanish variants and the second-ranked in 2/5 variants. Also, we performed several qualitative analyses with the aim of understanding the behavior of the self-attention mechanisms of the proposed model. We consider that a competitive sentiment analysis system should be able to relate and identify several aspects to determine the global polarity of a tweet. In this thesis, we considered two aspects to be analyzed: the polarity of the words and the presence of sentiment modifiers such as polarity

shifters or reversers in the tweets. We hypothesize that the attention heads of the Transformer encoders are specialized, after being trained for the sentiment analysis task, in detecting this kind of aspects. To study this specialization, we mainly focused on analyzing the average attention that each word receives from all the other words in the self-attention mechanisms, considering all the occurrences of the word in a given sample set. In our analysis, we found that this specialization actually occurs e.g., two attention heads are specialized in detecting positive and negative words independently.

3.1.1 Corpora

In order to validate our proposal for sentiment analysis on Twitter in the Spanish language, we participated in the Task 1 of TASS 2019 [2] [4]. The TASS 2019 workshop considered two different tasks: Task 1, Monolingual document-level sentiment analysis, where the systems have to be designed and evaluated for each individual variant; and Task 2, Crosslingual document-level sentiment analysis, intended to evaluate the generalization across the Spanish languages spoken in different countries. Both tasks consist on assigning global polarity to tweets on four classes $\mathbb{C} = \{N, NEU, NONE, P\}$. Classes P and N refers to positive and negative sentiment respectively. Class NEU refers to the case where both positive and negative polarities are present in the tweet. The $NONE$ class is used for tweets which do not convey any polarity.

The organizers provided the InterTASS corpora, composed by tweets from 5 different Spanish-speaking countries: Spain (ES), Peru (PE), Costa Rica (CR), Uruguay (UY) and Mexico (MX). It is especially interesting, as all these Spanish variants exhibits a large amount of lexical and even structural differences [2], thus showing the necessity to develop language-specific resources for them. The Spanish corpus was collected between 2016 and 2017, capturing tweets in Spanish, that contain at least one adjective, and more than four words. The Peruvian corpus was collected during 2018. There is not information, about the scrapping methodology for the rest of the variants, available in [2]. All of them were built in a similar way: each tweet was annotated by at least three annotators and, if there is not agreement, either two new annotators were used to disambiguate or a it is discussed the labeling in order the annotators reached a consensus. For each Spanish variant, three sample sets were provided: training set (TR), development set (DV) and test set (TS). Only one Spanish variant could be used both for training and testing the system (mono-lingual setup). Consequently, five different evaluations, one per Spanish variant, were proposed. Some statistics of the InterTASS corpora are shown in Table 3.1.

[4] During the development of this thesis, we also participated in previous and posterior editions of TASS with different approaches [15, 17, 18]

Table 3.1 Number of tweets per class in all the sample sets of InterTASS for all the Spanish variants.

	ES			CR			PE			UY			MX		
Class	TR	DV	TS	TR	DV	TS	TR	DV	TS	TR	DV	TS	TR	DV	TS
N	474	266	663	310	143	459	228	107	485	367	192	587	505	252	745
NEU	140	83	195	91	55	151	170	56	368	192	90	290	79	51	119
NONE	157	64	254	155	72	220	352	230	176	94	51	82	93	48	111
P	354	168	594	221	120	336	216	105	435	290	153	469	312	159	525
Σ	1125	581	1706	777	390	1166	966	498	1464	943	486	1428	989	510	1500

The InterTASS corpus is unbalanced, it is biased towards the *N* and *P* classes, except in the training and development sets of the PE variant, where the most frequent class is *NONE*. However, in the test set of this variant, the class distributions differs, being *N* and *P* the most frequent classes. Moreover, the class *NEU* is usually the less populated class in all Spanish variants. Some examples from the corpus are shown in Figure 3.3. It should be noted that there are some aspects that poses difficulties such as: lexical mistakes (sepriembre/*seprember*), metaphors like "cagadisima" (indicating fear) or "eres un sol" (to say someone is appreciated), effusiveness (!!), mentions to other users "@cris...", and even multimodality if it is required to process multimedia content from the urls.

N: Qué deprimente, mañana ya sepriembre. QUE SE ACABAN LAS VACACIONES!! (*How depressing, tomorrow is already seprember. VACATION IS ENDING !!*)

NONE: He decidido empezar a procesar las fotos macro del verano `https://...` `https://...` (*I have decided to start processing the macro photos of the summer* `https://...` `https://...`)

NEU: estoy contenta y cagadisima a la vez (*I am happy and I am afraid at the same time*)

P: @cris... gracias, guapa!! menuda promoción me haces!! eres un sol (*thank you, beautiful !! What a promotion you make me !! You are so good*)

Fig. 3.3 Examples of each class from the training set of InterTASS, also translated to English.

3.1.2 Proposed Approach

Our system is based on the Transformer [11] model. Initially proposed for machine translation, the Transformer model gets rid of convolution and recurrences to learn long-range relationships. Instead of this kind of mechanisms, it relies on multi head self-attention, where multiple attentions among the words of a sequence are computed in parallel to take into account different relationships among them. This reduces the computational complexity per layer (being also more parallelizable) and the max path length of dependencies among words to $\mathscr{O}(1)$ (instead of $\mathscr{O}(\log n)$ or $\mathscr{O}(n)$ in the cases of convolution and recurrent mechanisms

respectively). This effect is particularly interesting on this task, as the model has to learn these dependencies with few samples.

Concretely, we use the encoder part of the Transformer model in order to extract vector representations that are useful to perform sentiment analysis. We denote this encoding part of the Transformer model as Transformer Encoder (TE). Figure 3.4 shows the representation of the proposed architecture for the addressed task.

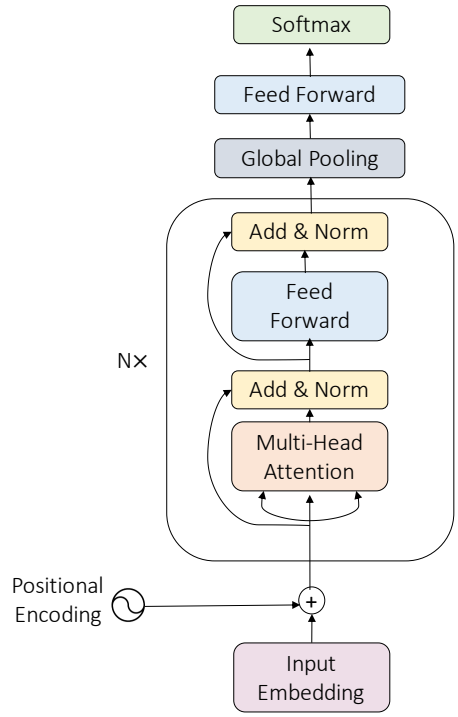

Fig. 3.4 Transformer encoder model used in the experimentation. Our implementation of this model for text classification can be seen in `https://github.com/jogonba2/TE-TextClassification`

The input of the model is a tweet $X = \{x_1, x_2, ..., x_T \ : x_i \in \{1, ..., V\}\}$ where T is the maximum length of the tweet and V is the vocabulary size. This tweet is passed through a d-dimensional pretrained embedding layer, E, frozen during the training phase. Moreover, to consider positional information we also experimented with the sine and cosine functions proposed in [11].

This, encoded as $P \in \mathbb{R}^{T \times d}$ is added to the embedding representation of the tweet to be used as input to the first encoder layer $X^0 \in \mathbb{R}^{T \times d}$, as show in Eq 3.17.

$$X^0 = \{\underbrace{P_1 + E(x_1)}_{X_1^0}, ..., \underbrace{P_T + E(x_T)}_{X_T^0} \ : X_i^0 \in \mathbb{R}^d\} \tag{3.1}$$

After the combination of the word embeddings with the positional information, dropout [49] was used to drop input words with a certain probability p to regularize the model. On top of these representations, N transformer encoders are applied, which rely on the multi-head scaled dot-product attention shown in Eqs 3.18 - 3.20. These encoders are identical to [11], including the layer-normalized [52] residual connections.

$$MultiHead(A,B,C) = [head_1;...;head_h]W^O \tag{3.2}$$

$$head_i = Attention(AW_i^Q, BW_i^K, CW_i^V) \tag{3.3}$$

$$Attention(Q,K,V) = softmax(\frac{QK^\intercal}{\sqrt{d_k}})V \tag{3.4}$$

where $W_i^Q \in \mathbb{R}^{d\times d_k}$, $W_i^K \in \mathbb{R}^{d\times d_k}$, $W_i^V \in \mathbb{R}^{d\times d_k}$, $W^O \in \mathbb{R}^{h\cdot d_k\times d}$, are the projection matrices for query, key and value of the head i and for the output of the multi-head attention respectively; and h is the number of heads for the multi-head attention mechanism.

The output for only one encoder, S, is computed as shown in Eq 3.24 for a given sample X^0.

$$M = MultiHead(X^0, X^0, X^0) \tag{3.5}$$

$$L = LayerNorm(X^0 + M) \tag{3.6}$$

$$F = max(0, LW_1 + b_1)W_2 + b_2 \tag{3.7}$$

$$S = LayerNorm(L + F) \tag{3.8}$$

where $M, L, F \in \mathbb{R}^{T\times d}$ are the intermediate outputs from the encoder, $W_1 \in \mathbb{R}^{d\times d_{ffw}}$, $W_2 \in \mathbb{R}^{d_{ffw}\times d}$ are the weights of the position-wise feed forward network, and $S \in \mathbb{R}^{T\times d}$ is the output of the encoder. When several encoders are stacked, the input of a encoder is used directly as input to the next encoder.

Due to a vector representation is required to train classifiers on top of these encoders, a global average pooling mechanism was applied on S. The resulting vector is used as input to a single-layer feed-forward network, whose output layer computes a probability distribution over the four classes of the task $\mathbb{C} = \{N, NEU, NONE, P\}$.

We use Adam as update rule with $lr = 0.001, \beta_1 = 0.9$ and $\beta_2 = 0.999$ and Noam as learning rate schedule [11] with 15 *warmup_steps*. Due to the imbalance in all the Spanish variants subsets, weighted cross entropy is used as loss function considering the distribution of each class in the training set. Concretely, we used the proportion between the most frequent class and the frequency of a given class, $w_i = \frac{\max_{c \in \mathbb{C}} n_c}{n_i}$, where n_i is the number of samples of the class i in a given set, being $w_i = 1$ if i is the most frequent class and $w_i > w_j$ if i is less frequent than the class j in the given sample set.

In order to initialize the embedding layer of our system with a rich in-domain representation for the words of the task, a 300-d skipgram model [80] was trained on Spanish tweets. Specifically, this model was trained by using 87M tweets from several Spanish variants, downloaded by streaming during several months in 2017 in our laboratory. We use a Twitter streamer for obtaining those tweets (including retweets) with at least one Spanish stopword such as "que", "de" or "donde". The behavior obtained by both word embedding models has been proven previously in several text classification tasks [17, 18, 22, 25, 146].

Regarding to the preprocessing, we have applied the same preprocess steps to all the given data, both the tweets used to learn the Word2Vec embeddings model and those provided by the organization to train the systems. Firstly, a case-folding process is applied to all the tweets, secondly, we tokenized the tweets by using TokTokTokenizer from NLTK [147]. Thirdly, user mentions, hashtags and URLS are replaced by three generic-class tokens (*user*, *hashtag* and *url* respectively). Finally, elongated tokens are diselongated allowing the same vowel to appear only twice consecutively in a token (e.g., *jaaaa* becomes *jaa*).

3.1.3 Evaluation

In order to validate our proposal for sentiment analysis in Twitter and to select the best model to participate in the 2019 edition of TASS competition, we carried out some experimentation on the development set. To train the models, we fixed some hyper-parameters such as $batch\_size = 32$, $d_k = 64$, $d_{ff} = d$ and $T = 50$. Other hyper-parameters such as p, $warmup\_steps$ or h were established, considering the results obtained in previous experiments, to $p = 0.7$, $warmup\_steps = 5$ epochs and $h = 8$.

Moreover, we compared our proposal, which is based on Transformer Encoders (TE), with other deep learning systems such as Deep Averaging Networks (DAN) [53] and Attention Long Short-Term Memory Networks [10] (Att-LSTM), that are commonly used in related text classification tasks and they were the backbone for most of the promising approaches in previous TASS editions.

We were also interested in observing how the use of positional encodings and the number of encoder layers affect to the results obtained, due to the model is not pretrained in any self-supervised task and the corpora are small. Specifically, we train different models removing the positional information (TE-NoPos) and using 1 or 2 encoders. We tested all these combinations only on the ES variant and the best two configurations were also applied to the remaining variants (PE, CR, UY, MX).

The results in terms of macro-F_1 (MF_1), macro-recall (MR), macro-precision (MP) and Accuracy (Acc) achieved by all the systems considered in the development phase, for all the Spanish variants, are shown in Table 3.2. It can be seen that the best transformer encoders models (1-TE-NoPos, 2-TE-NoPos) outperform the DAN and Att-LSTM approaches by a margin of $\sim$5 points for MF_1 measure. This is due to the great improvement in both MR ($\sim$6 points) and MP ($\sim$3 points).

Table 3.2 Results on the development set for the different Spanish variants.

	MP	**MR**	$\boldsymbol{MF_1}$	**Acc**
		ES		
DAN	47.66	48.46	47.94	56.28
Att-LSTM	50.00	48.14	48.83	58.00
1-TE-NoPos	52.80	**54.38**	**53.34**	60.75
1-TE-Pos	46.26	46.56	46.25	55.94
2-TE-NoPos	**52.85**	53.03	51.47	**61.27**
2-TE-Pos	47.31	48.79	47.71	56.11
		PE		
1-TE-NoPos	**49.06**	**50.43**	**49.51**	**54.62**
2-TE-NoPos	46.29	46.00	44.92	46.79
		CR		
1-TE-NoPos	**55.36**	**56.10**	**54.56**	**58.46**
2-TE-NoPos	52.14	52.36	51.71	55.13
		UY		
1-TE-NoPos	54.71	**56.63**	**54.83**	57.20
2-TE-NoPos	**55.82**	53.56	54.29	**58.64**
		MX		
1-TE-NoPos	**53.59**	55.03	**54.10**	**63.52**
2-TE-NoPos	52.78	**57.34**	54.07	60.78

The use of the positional information in the TE approaches decreases the system performances (1-TE-Pos versus 1-TE-NoPos and 2-TE-Pos versus 2-TE-NoPos) i.e. the positional information, represented by sine and cosine functions added to the word embeddings, is not useful to the classification. However, the results obtained by Att-LSTM, which considers the positional information by its internal memory, are better than those obtained by the 1-TE-Pos

and 2-TE-Pos approaches in almost all the metrics. This suggests that the way in which the positional information is considered for the TE models is not well suited for this task. The 1-TE-NoPos model obtains better results, in terms of MR and MF_1, than the 2-TE-NoPos model, outperforming its results on $\sim$2 points in terms of MF_1. This behavior is observed in almost all the variants, except in the MX variant, where both models obtain similar results in terms of MF_1 and 2-TE-NoPos outperforms 1-TE-NoPos in terms of MR.

Table 3.3 shows the results, at class level, achieved by the best model (1-TE-NoPos) for all Spanish variants. In most cases, the results obtained for the *N* and *P* classes are better than those obtained for the other classes, except in the PE variant, where the *NONE* class is the one that obtains the best results, possibly because *NONE* is the most populated class in the training set of the PE variant. For all Spanish variants, as expected, the most difficult class is the *NEU* class due to the fact that this class requires the systems to understand the neutralization of positive and negative sentiments.

Table 3.3 Results at class level for the 1-TE-NoPos model and all Spanish variants on the development set.

	N			NEU			NONE			P		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
ES	**73.03**	**73.31**	**73.17**	30.56	26.51	28.39	46.34	59.38	52.05	61.25	58.33	59.76
PE	51.40	51.40	51.40	27.27	26.79	27.03	**64.88**	57.83	**61.15**	52.67	**65.71**	58.47
CR	**74.58**	61.54	**67.43**	27.87	30.91	29.31	46.09	**73.61**	56.68	72.92	58.33	64.81
UY	**69.70**	47.92	56.79	34.51	43.33	38.42	50.00	58.85	54.05	64.64	**76.47**	**70.07**
MX	**73.93**	**75.40**	**74.66**	30.91	33.33	32.08	44.07	54.17	48.60	65.47	57.23	61.07

In order to study in detail the behavior of our best system (1-TE-NoPos), we computed the confusion matrix for the ES variant, that can be seen in Table 3.4. Note that, the *NEU* class is highly confused with the *N* and *P* classes, indicating that our model detects the presence of sentiment (positive or negative), but it is not capable to detect when both sentiments occur together. In addition, it can be observed that the *N* and *P* classes are also confused with each other, being the most confused between them.

Table 3.4 Confusion matrix (1-TE-NoPos) on the ES variant development set.

	N	**NEU**	**NONE**	**P**
N	195	25	18	28
NEU	25	22	13	23
NONE	9	6	38	11
P	38	19	13	98

In light of the results of the development phase, we decided to use 1-TE-NoPos system to participate in the TASS 2019 competition. Table 3.5 shows the official results for all Spanish

variants and the position of our system (ranked using F_1 measure) in each variant [2]. As it can be seen, our system is ranked in first place for the ES, MX and UY variants and in second place for CR, and PE variants.

Table 3.5 Official results and ranking of our system on the TASS 2019 competition [2]

	MF_1	MP	MR	Rank
ES	50.68	50.52	50.85	1/6
CR	49.58	49.84	49.33	2/6
PE	44.74	45.63	43.82	2/6
UY	51.54	49.68	53.55	1/6
MX	50.10	49.05	51.21	1/6

3.1.4 Analysis

With the aim of understanding the proposed model, we have analyzed the behavior of the self-attention mechanisms. A competitive sentiment analysis system should be able to combine several aspects to determine the polarity of a tweet. Among others, some of these aspects are the polarity of the words and the presence of sentiment modifiers such as polarity shifters or reversers in the tweets. We hypothesize that the attention heads of our system should capture some of these aspects. In order to determine what heads react to these aspects, we computed the average attention that each word receives from each head considering all the occurrences of the word in a given sample set.

The development set of the ES variant is used to verify that the model generalizes and captures interesting relationships even in samples that it has never seen. Formally, from the set of samples χ with vocabulary $\mathscr{V}$ and the trained model Θ, it is possible to calculate the attention given by the head k to a word w in the sample set χ. To do this, from each sample x of the set χ and each head k, the matrix $B \in \mathbb{R}^{|x|\times|x|}$, which contains the attentions of this head after a forward pass on the model Θ, is computed. We formalized this computation in Algorithm 1.

The columns of this attention matrix are averaged to obtain $B' \in \mathbb{R}^{|x|}$. This matrix B' contains the attention that head k gives to each word in x, computed as the average of the self-attentions in the head. Finally, the attention of each word in each head, α_{wk}, is calculated by averaging the attention given by head k to word w in all the samples. Once α is computed, it is possible to observe if some heads are specialized in word-level properties that are necessary to determine the sentiment of a tweet. Mainly, the polarity of each word, that has to be considered by the model to infer a global sentiment by means of compositionality, considering the polarity modifiers.

Algorithm 1 Compute the average word attentions captured by the model on a set of samples.

Input: $\mathcal{V}$ vocabulary, set of samples χ, trained Transformer Encoder Θ
Result: α_{wk} the average attention of head k for word w

procedure COMPUTEWORDATTENTIONS(χ, Θ)
  for $w \in \mathcal{V}$ **do**
    for $1 \leq k \leq h$ **do**
      $\alpha_{wk} \leftarrow 0$
    end for
  end for
  for $x \in \chi$ **do**
    for $1 \leq k \leq h$ **do**
      $B \leftarrow softmax(\frac{\Theta(x)_{Q_k}\Theta(x)_{K_k}^{\top}}{\sqrt{d_k}})$
      $B' \leftarrow \frac{1}{|x|}\sum_{i=1}^{|x|} B_{ij}$
      for $w \in x$ **do**
        $\alpha_{wk} \leftarrow \alpha_{wk} + B'_w$
      end for
    end for
  end for
  for $w \in \mathcal{V}$ **do**
    for $1 \leq k \leq h$ **do**
      $\alpha_{wk} \leftarrow \frac{\alpha_{wk}}{c_w}$
    end for
  end for
end procedure

Figure 3.5 shows the attention of all heads (from 1 to 8) for 6 words with high polarity. These words are extracted from the ElHuyar [119] lexicon. First row in Figure 3.5 shows the attention per head of three words with positive polarity (*best*, *wonderful* and *cool*) and the second row corresponds to three words with negative polarity (*worst*, *horrible* and *shit*). It can be observed that the attention heads 4 and 5 react with high intensity when the polarity is negative and positive respectively. Moreover, head 4 does not react when the polarity is positive, the same behavior is observed for head 5 when the polarity is negative. Furthermore, heads 6 and 7 seem to attend to the negative words and not to the positive ones; head 3 reacts more intensively to positive words rather than negative ones. Also, Table 3.6, shows the top-10 words most attended by the heads 4 and 5, having all of them, negative and positive connotations respectively.

We extended the study to all words of the ElHuyar polarity lexicon [119] that appear in the vocabulary $\mathcal{V}$. Figure 3.6 shows average attentions per head for positive and negative words from ElHuyar. It can be observed the same behavior, of the heads 4 and 5, than in the previous case with the words of the development set of TASS. In this case, the negative words receive higher attention than the positive ones. In particular, the head 4 reacts more to negative words than the head 5 reacts to positive words.

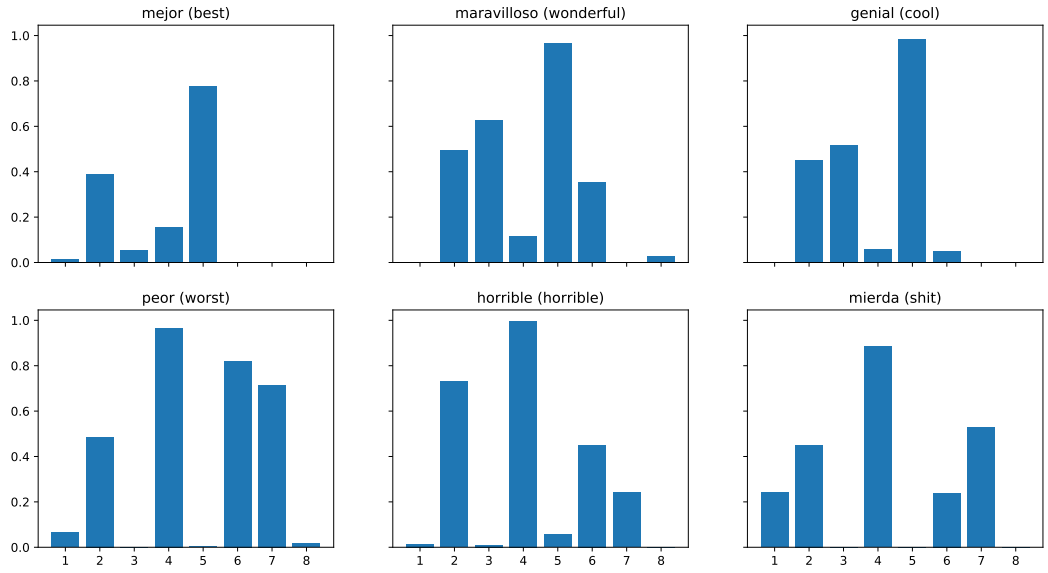

Fig. 3.5 Attentions for several words that contains sentiment.

Table 3.6 Top-10 most attended words by the attention heads 4 and 5, including the average attention of each one.

Heads	(w, α_{wk})
Head-4	(insoportable, 1.00), (clasista, 0.99), (gilipollas, 0.99), (perverso, 0.99), (grasioso, 0.99), (insufrible, 0.99), (asco, 0.99), (soporto, 0.99), (despreciable, 0.99), (machismo, 0.99)
Head-5	(preciosa, 0.99), (bonica, 0.99), (hermosa, 0.99), (favoritos, 0.99), (guapas, 0.99), (cantas, 0.99), (adorables, 0.99), (mejores, 0.99), (enamoradisima, 0.99), (personita, 0.99)

To confirm the capability of the heads 4 and 5 detecting the polarity of the words, we designed a classifier that uses only the average attention of the heads 4 and 5 (α_{w4} and α_{w5}), in order to determine the polarity of each word w of the vocabulary $\mathscr{V}$. This classifier is formalized in Eq. 3.9.

$$\mathscr{C}(w) = \begin{Bmatrix} P & \text{if } \alpha_{w4} \leq \alpha_{w5} \\ N & \text{if } \alpha_{w5} < \alpha_{w4} \end{Bmatrix} \tag{3.9}$$

We tested the performance of classifier $\mathscr{C}$ by classifying all words of ElHuyar lexicon that appear in the vocabulary. Note that the words in ElHuyar have only positive or negative polarity. The classifier achieved an Accuracy of 74.75% which confirms the ability of the attention heads 4 and 5 capturing the polarity at word level.

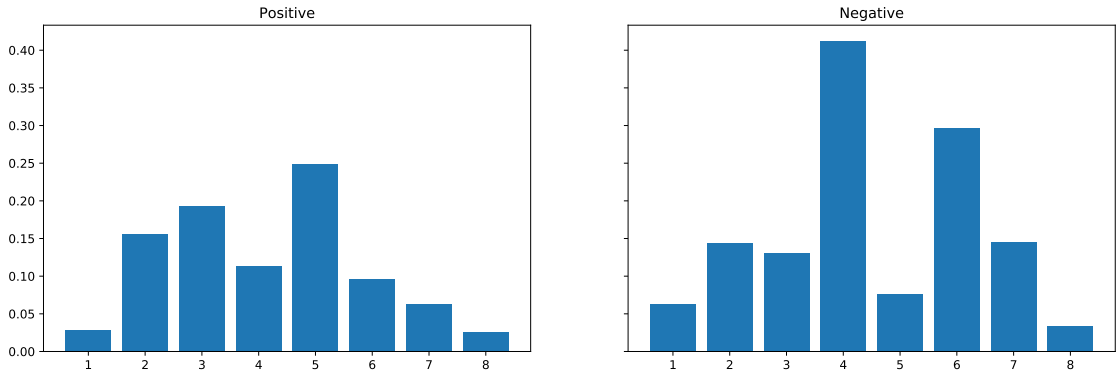

Fig. 3.6 Sum of attentions for all the attention heads on the words of ElHuyar.

We attempted to address the Task 1 of TASS 2019 for ES variant using only the information of heads 4 and 5 and the ElHuyar lexicon. To do this, we designed a classifier based on the sum of the polarity of the words. The classifiers works as follow: if the sample does not contain any word with polarity then, its class is $NONE$, if the sample contains the same number of positive and negative words its class is NEU, otherwise the class of the sample is P or N depending of the number of positive and negative words.

This classifier is directly computable on any polarity lexicon (e.g ElHuyar), however to use the heads 4 and 5 of our system we need to design a mechanism to discretize the polarity of each word based on the outputs of both heads. In our case, we compute a probability distribution over the P and N classes by means of a softmax function on the attentions computed from the two heads. To discretize this function, we used a threshold $\varepsilon = 0.165$ experimentally defined. This classifier, SumPolClassifier, is defined in the Algorithm 2.

In order to use the SumPolClassifier with ElHuyar lexicon, $p(N|w)$ and $p(P|w)$ are obtained directly from the lexicon. Table 3.7 shows the results of SumPolClassifier applied to the development set of the ES variant of Task 1 of TASS 2019 both with heads 4 and 5, and ElHuyar lexicon. It can be seen how the results in terms of macro-F_1 are similar in both approaches. Both systems classify similarly the classes NEU, $NONE$ and P. However, the recall on the class N with the heads 4 and 5 is significantly lower than with ElHuyar although they have more precision.

Finally, we studied how attention heads react to words that are supposed to be polarity shifters or polarity reversers. Figure 3.7 shows average attentions per head for eight of these words. The words in the first row (*not*, *never*, *neither* and *anybody*) are polarity reversers and the words in the second row (*very*, *nothing*, *forever* and *something*) are polarity shifters.

It can be seen that head 1 reacts to all the shifters and reversers. This head does not react to positive or negative words (see Figures 3.5 and 3.6). In addition, heads 4 and 5 do not

Algorithm 2 SumPolClassifier based on the heads 4 and 5 to classify the polarity of tweets.

Input: sample set χ and α the attentions per head of all word w in the vocabulary $\mathcal{V}$.
Result: $\hat{y}$, labels assigned by the classifier to all samples in the sample set.

procedure SUMPOLCLASSIFIER(χ, α)
  for $x \in \chi$ **do**
    pol $\leftarrow 0$
    neutralized $\leftarrow$ false
    for $w \in x$ **do**
      $p(N|w) \leftarrow \frac{e^{\alpha_{w4}}}{e^{\alpha_{w4}}+e^{\alpha_{w5}}}$
      $p(P|w) \leftarrow \frac{e^{\alpha_{w5}}}{e^{\alpha_{w4}}+e^{\alpha_{w5}}}$
      if $|p(N|w) - p(P|w)| \geq \varepsilon$ **then**
        neutralized $\leftarrow$ true
        if $p(N|w) > p(P|w)$ **then**
          pol $\leftarrow$ pol - 1
        else
          pol $\leftarrow$ pol + 1
        end if
      end if
    end for
    if pol > 0 **then**
      $\hat{y}_x \leftarrow P$
    else
      if pol < 0 **then**
        $\hat{y}_x \leftarrow N$
      else
        if neutralized **then**
          $\hat{y}_x \leftarrow NEU$
        else
          $\hat{y}_x \leftarrow NONE$
        end if
      end if
    end if
  end for
end procedure

Table 3.7 Results of SumPolClassifier both using the heads 4 and 5, and ElHuyar lexicon on the development set.

	Heads 4/5			**ElHuyar**		
	P	R	F_1	P	R	F_1
N	63.73	41.14	50.00	62.10	53.59	57.53
NEU	14.05	24.29	17.80	16.25	18.57	17.33
NONE	23.45	33.76	27.68	27.32	31.85	29.41
P	57.26	56.78	57.02	56.30	59.32	57.77
Macro	39.62	38.99	38.12	40.49	40.83	40.51

react to shifters nor reversers because these words do not have polarity per se. However, the attention values for head 1 are not relatively high except in the case of *no* and *always*. These results seem to indicate that, although it reacts fairly well to common shifters and reversers, it is necessary to reinforce the attentions dedicated to this type of words, in order to improve the capabilities of our system to compose a global sentiment for the tweets. It is also remarkable that all the polarity reversers and the polarity shifter *nothing*, all of them with negative connotation, are attended by the head 7 that was related to the negative polarity as previously discussed.

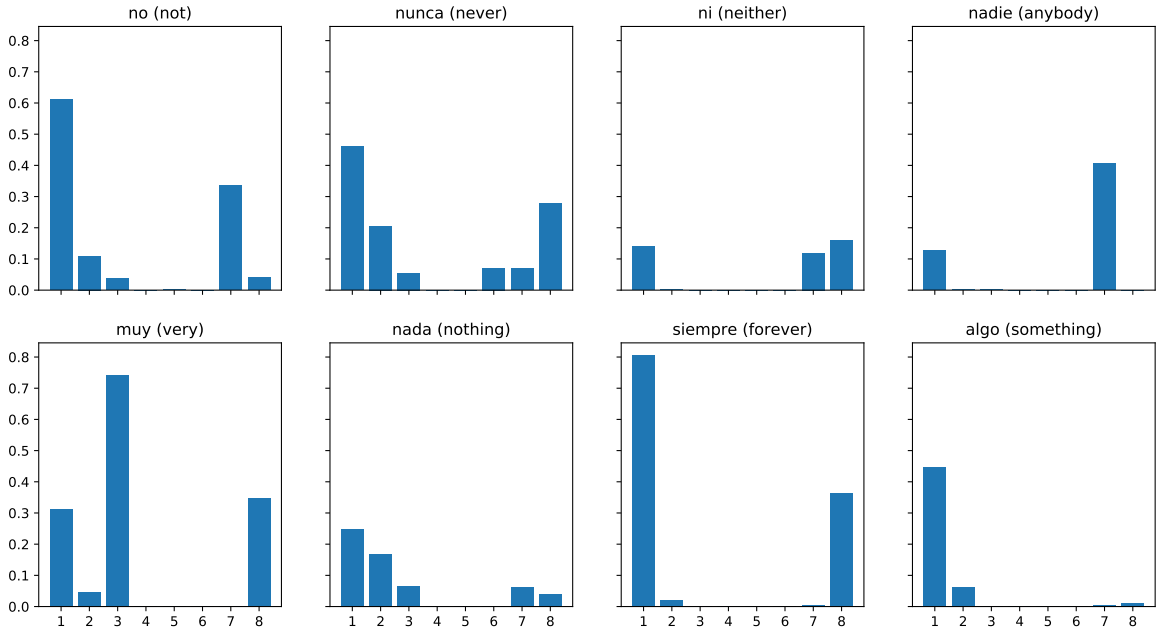

Fig. 3.7 Attention per head on polarity reversers and shifters.

3.2 Emotion Detection

Understanding the emotions is an important aspect for understanding also the sentiment expressed in opinions. Emotions are closely related to the sentiment, being the strength of the sentiment directly related with the intensity of certain emotions [101]. Although there is not scientific consensus on the definition of emotions, they are typically considered as complex physiological and psychological states that influence the users to act in their environment, acting as a key motivational operator [148]. In that sense, emotions are closely connected to sentiments, as the sentiments are highly organized mental attitudes, used to convey the emotional thoughts of the individual that emerge as a response to some event. In this way, the sentiment is built on top of the emotions, and, while the emotions are physiological and psychological aspects, sentiment has a social condition, connecting primary emotions with actions [149]. It is also interesting to note that emotions do not have to always correspond to a unique feeling. For example, imagine that several people feel fear in an escape room. After that experience, some of these people will likely think positively about that experience. However, if the same people feel fear in a terrorist attack, none of them will think of it as a positive experience.

The boundaries between emotions are fuzzy, being difficult to know where one emotion ends and another emotion begins. Furthermore, there is not consensus about how many emotions are there, as they can be combined to form more complex emotions, the emotion space is very huge to be characterized in terms of taxonomies. Due to this, several works were intended to define taxonomies of basic emotions (those emotions that are universal and irreducible) [150, 151] and continuous models where emotions are points in n-dimensional spaces [152]. Emotions have also a great impact on the nervous system [153, 154], thus being physiologically expressed by means of, among others, facial expressions, body movements, pitch shifts, or even speech fluency. By this way, emotions exhibit a multimodal behavior, so, we require systems capable to tackle this multimodality in order to perform complex studies that allow us to fully understand them.

In the computer science literature, the terms “emotion” and “sentiment” are typically used interchangeably, however, from a psychological point of view, as discussed above, this is a very strong assumption. For this reason, a recent trend consists of considering both emotion recognition and sentiment analysis as two broad and different subfields inside the Affective Computing field [155, 156]. Affective Computing is a multidisciplinary field, intended to study techniques for performing affect recognition in different modalities and granularities [155]. Under this definition, sentiment analysis can be interpreted as a coarse analysis, on a set of few classes e.g., positive, negative, and neutral; while emotion detection aims for a more fine-grained classification, on a large set of classes. As discussed in the first paragraph,

there is no consensus about a unique categorization of emotions, and this categorization is strongly required for computational analyses, as it defines the set of labels to work with. In order to address such problem, the research community has typically focused on using (and potentially extending) mainly three categorizations of the emotion space: the Ekman's 6 basic emotions, the Plutchik's wheel (8 emotions), and the Russell's valence-arousal-dominance model (continuous representation of emotions).

On the one hand, the Ekman's proposal [150] establishes that the emotions can be classified in a discrete taxonomy of 6 irreducible basic emotions: anger, fear, disgust, surprise, sadness, and happiness, that can be blended to compose more complex emotions. Based on the blending of emotions, the Plutchik's wheel of emotions was proposed [151]. In this modeling, eight basic emotions, matched in four bipolar dyads, were identified: joy/sadness, anticipation/surprise, anger/fear, disgust/trust. They were organized as a wheel, following properties that correlate them. Concretely, similar emotions are near in the representation space and bipolar emotions are placed on the opposite side of the space. From these basic emotions, Plutchik distinguished also the intensity and the emotions that can arise from the eight basic emotions depending on the intensity. These emotions were represented on top or bottom of the basic emotions, depending on the intensity, shaping a flower with eight three-layered petals inside the wheel. The boundaries of the petals represent complex emotions that emerge by combining the two adjacent basic emotions e.g., trust + fear $\rightarrow$ submission. The previous two models are built on top of the basic emotion structure, however, other predominant approaches are built upon n-dimensional spaces where the emotions emerge from combinations of the dimensions. The most popular approach based on this idea is the Russell's valence-arousal-dominance model (VAD) [152]. In this proposal, a continuous three-dimensional space composed of three orthogonal axes: valence, which determines the positive or negative tendency (positiveness/negativeness); arousal, which measures the degree of excitation (active/passive); and dominance, which indicates whether the situation is under control or not (dominance/submission).

The taxonomies are the most used in the emotion recognition field. In this case, considering unimodality on text, the systems are required to classify the texts on a finite set of emotion labels. Many corpora have been built for this purpose, being the International Survey on Emotion Antecedents and Reactions (ISEAR) one of the most representative, that provides English descriptions of emotional events, categorized by means of 7 different emotions such as anger, disgust, fear, guilt, joy, shame, and sadness [157]. However, from a computational linguistic perspective, focused on short-texts from domains like Twitter or newspapers, the most relevant corpora aimed to perform emotion detection have been proposed in the SemEval workshop [113, 158–161], for the English language, and in the

TASS workshop [112, 115], for the Spanish language. Among these corpora, three different domains can be identified: newspapers [112, 158], Twitter [113, 160], and open-domain dialogues between humans and conversational agents [159]. For the newspapers domain, [158] proposes an emotion classification task focused on English news headlines, with the aim of categorizing the headlines in a predefined set of six emotion labels. As the headlines are a key factor to grab the reader's attention, they tend to provoke emotions by appealing to affective and emotional features, which made them a particularly suitable content to use on emotion detection tasks. Similarly, [112] is focused on the emotional categorization of RSS feeds extracted from different online newspapers written in several variants of the Spanish language. However, its purpose is different from [158] in the sense that the objective is to categorize each RSS feed in terms of its emotional safeness, considering positive emotions as safe and negative emotions as unsafe. This is especially interesting from a marketing perspective, as if companies want to promote their brands, the advertisements should better be associated with news that evoke positive emotions. Regarding the Twitter domain, in [113] an array of subtasks for inferring the affectual state of a person from their English written tweets was proposed. The subtask dedicated to emotion classification is the E-c subtask, where, given a tweet, the objective is to classify it in one or more of eleven given emotions that best represent the mental state of the user. In the same way, [115] focuses on categorizing Spanish tweets as neutral or as one of the six Ekman's basic emotions. An interesting proxy to perform emotion detection in Twitter consists in predicting emojis. Emojis are one of the main components of communication in social media environments, being used to convey information about people, scenes, objects or ideas. Moreover, they carry sentiment and emotion information that can be used to achieve better language understanding and to improve highly subjective tasks such as sentiment analysis and emotion detection. One of the most relevant works in this regard is [160], which proposes an emoji prediction task where the objective is to predict the best-suited emoji for Spanish and English tweets. For the open-domain dialogues domain, [159] proposed a task for contextual emotion detection in dialogue contexts. They observed that the context of ongoing dialogues could change the emotion of the utterances, and this is especially important for conversational scenarios such as digital assistants or conversational agents. Specifically, the objective of this task is to categorize an utterance, emitted by a user, given the two previous utterances (from the user and the conversational agent), following a set of four emotional classes included in the Ekman's basic emotions.

Regarding the systems proposed in the literature to address the emotion detection tasks, it can be observed the progressive evolution since [158] until our days. Initially, most of the systems were unsupervised and rule-based approaches that exploited lexical resources

with emotional content such as SentiWordNet, WordNetAffect, or information retrieval systems to compute statistical associations between emotions and headlines [162–165]. Some supervised approaches were also proposed, mainly based on traditional machine learning systems such as Naive Bayes, Nearest Neighbors and SVM classifiers that exploited bag-of-words representations [166–168]. Over the years, Deep Learning approaches and distributed representations have become ubiquitous in this task. This can be seen in [112, 113, 160, 161], where almost all the participants used this kind of systems. These tasks were dominated by LSTM and CNN approaches, either as classifiers on top of word embeddings [169–171] or as feature extractors [172, 173] to build decision systems, mainly based on SVM or Gradient Boosting. Typically, attention mechanisms play a key role in these systems to identify salient words that convey emotional content [169, 173]. Also, emotional lexical resources are still used to enrich the systems with task-specific information [25, 171–173]. More recently, the use of pre-trained contextualized representation models has been democratized in the emotion detection task, as it can be seen in [115, 159]. By this way, transfer learning using BERT, ELMo and ULMFit was a popular choice among top teams [15, 174–176], although LSTM and CNN based approaches are still competitive against these pretrained approaches [21, 177, 178].

In some cases, despite that the set of emotion labels is finite, it could be large if it is composed of overlapped emotions, thus posing multi-label classification tasks. Besides, we work with Twitter, which means that the data is extracted from a strongly biased population where some basic emotions are conveyed more frequently than other emotions e.g., joy and sadness are more frequent than surprise and trust[5]. This imbalance biases the systems to predict much more frequently the most populated classes. To avoid this imbalance during the evaluation of the systems, a *de facto* standard is to use the M-F_1, where all the classes contribute equally to the global measure. However, this does not change the learning phase of the models, so, in this sense, it is interesting to consider mechanisms to incorporate this evaluation criterion in the learning phase of the models.

In deep learning approaches, the loss function is used by the back-propagation algorithm to guide the parameter estimation process. Although any differentiable function can be used as loss function, a few numbers of loss functions are usually used, without considering the measures used to evaluate a specific task. Two of the most used loss functions in the literature are the Cross-Entropy (CE), in its binary (BCE) or categorical versions, and the Mean Squared Error (MSE). These functions are individually computed for each sample, which means that class-level performance during the learning process is not taken into account. Some recent works proposed different loss functions for further improving deep learning

[5]Most used emojis in Twitter during 2019: `https://twitter.com/TwitterData/status/1204134086241062912`

systems in this direction. A study of how particular choices of loss functions affect deep models and their learning dynamics can be consulted in [179]. Most similar to our aim, two works intended to optimize the F_1 measure [180, 181]. The first one [180] introduced a novel plug-in rule algorithm that estimates all parameters required for a Bayes-optimal prediction in multinomial regression models. The last one [181] integrated the F_1 measure as training criterion in the backpropagation algorithm, for an image cleaning and enhancement binary task. In this thesis, we extend their work by defining commonly used evaluation metrics as loss functions to train neural networks for multi-class and multi-label emotion detection. In order to evaluate different loss functions on an imbalanced multi-label emotion detection corpus, we selected the *E-c: Detecting Emotions Multilabel classification* corpus used in subtask 5 of Task 1 (*Affect in Tweets*) at the International Workshop on Semantic Evaluation, SemEval-2018 [113].

3.2.1 Corpora

In this subsection, we present the main characteristics of the subtask 5 of Task 1: *"E-c: Detecting Emotions Multilabel classification"*[113] proposed at the International Workshop on Semantic Evaluation, SemEval-2018 [6]. We address the task for the English and the Spanish languages. This task can be formalized as a multi-class/multi-label classification problem, that is, given a text, the systems classify it in one, or more, of eleven given emotions (based on the Plutchik's wheel) that best represents the mental state expressed in the text. The corpus supplied by the organizers is a collection of tweets tagged with a set of emotions.

Table 3.8 shows the details of this corpus both for English and Spanish languages. It can be seen the number of samples per emotion in the partitions of training, development, and test that participating teams must use for their systems. It is also showed the total number of tweets. From these figures, it can be inferred that the average number of labels per tweet is about 2.3 for English and 1.7 for Spanish.

Figure 3.8 shows 4 examples from the training set of the SemEval E-c corpus, both for the English and the Spanish languages. It can be seen how the examples can convey different emotions such as (*joy*, *love*), or even more complex combinations such as (*anger*, *love*) or (*anger*, *optimism*).

The official competition metric used for ranking the teams in E-c task was multi-label accuracy (or Jaccard index) as defined in Equation EA.1 of the appendix §A.1. Since this is a multi-label classification task, each tweet can have one or more gold emotion labels, and one

[6] `http://alt.qcri.org/semeval2018/index.php?id=tasks`

Table 3.8 Data set distribution of the Emotion Classification task at Semeval-2018.

	English				Spanish			
Emotion	**Train**	**Dev**	**Test**	$\sum$	**Train**	**Dev**	**Test**	$\sum$
Anger	2544	315	1101	3960	1155	209	919	2283
Anticipation	978	124	425	1527	415	94	321	830
Disgust	2602	319	1099	4020	521	98	423	1042
Fear	1242	121	485	1848	373	74	298	745
Joy	2477	400	1442	4319	1087	201	873	2161
Love	700	132	516	1348	261	55	245	561
Optimism	1984	307	1143	3434	378	66	278	722
Pessimism	795	100	375	1270	578	115	495	1188
Sadness	2008	265	960	3233	845	143	644	1632
Surprise	361	35	170	566	169	37	122	328
Trust	357	43	153	553	175	31	122	328
$\sum$	16048	2161	7869	26078	5957	1123	4740	11820
#samples	6838	886	3259	10983	3561	679	2854	7094

Example 1: I'm absolutely in love with Laurie Hernandez, she's so adorable and is always so cheerful! (*joy*, *love*)

Example 2: The best revenge is massive success (*anger*, *optimism*)

Example 3: Mi enojo dura 5 minutos y después ya te extraño. [*My anger lasts 5 minutes and then I miss you.*] (*anger*, *love*)

Example 4: Diablo ni yo me lo creo [*Hell, I don't believe it*] (*surprise*)

Fig. 3.8 Examples from the training set of the SemEval E-c corpus both for the English and the Spanish languages. English translation is also considered for the Spanish examples.

or more predicted emotion labels. Apart from the official competition metric (multi-label accuracy), we also report m-F_1 and M-F_1 in order to validate our proposal

3.2.2 Proposed Approach

In this section, we present some differentiable functions that are approximations to the evaluation metrics discussed in §A.1. These functions can be used as loss functions to train neural networks for multi-class and multi-label classification problems, and they are especially relevant in our work to deal with emotion detection tasks. Also, we present the main characteristics of the deep learning system that is trained under these loss functions. Furthermore, we discuss the tweet preprocessing, the external resources used to add polarity/emotion information to the model and the tweet representation.

When neural networks are used to address classification problems, the last layer of the network needs as many neurons as classes, being n the number of classes. Let o_i be the output layer of the network when processing sample i (out of a total amount of m samples), and o_i^j the value of the jth neuron of o_i, that is, the value assigned to class c_j. Let y_i be the set of correct classes of sample i represented as a vector, thus $y_i^j = 1$ if sample i belongs to class c_j, otherwise $y_i^j = 0$. In order to determine, from o_i, the classes assigned to the sample i by the model, it would be necessary to set a threshold and to select those classes for which the value of o_i^j is greater than that threshold. This would make the resultant function non-differentiable. To solve this problem, in this work, we propose the use of the following approximations:

a) $|\theta_i| \approx \sum_{j=1}^{n} o_i^j$. The number of classes in the prediction for sample i, $|\theta_i|$, is approximated as the sum of the values of all the neurons in the output layer of the network, $\sum_{j=1}^{n} o_i^j$.

b) $|\gamma_i| = \sum_{j=1}^{n} y_i^j$. The number of classes in the reference for sample i, $|\gamma_i|$, is computed as the sum of the values in y_i, $\sum_{j=1}^{n} y_i^j$.

c) $|\gamma_i \cap \theta_i| \approx \sum_{j=1}^{n} y_i^j \cdot o_i^j$. The number of correctly predicted classes for sample i, $|\gamma_i \cap \theta_i|$, is approximated as the weighted sum of the output layer and the vector of correct classes for sample i, $\sum_{j=1}^{n} y_i^j \cdot o_i^j$.

d) $|\gamma_i \cup \theta_i| \approx \sum_{j=1}^{n} \left(y_i^j + o_i^j - y_i^j \cdot o_i^j \right)$. Applying the set theory for calculating the cardinality of a union set, the normalization factor of the Accuracy, $|\gamma_i \cup \theta_i|$, is computed using the three previous approximations.

e) $\sum_{i=1}^{m} [c_j \in \theta_i] \approx \sum_{i=1}^{m} o_i^j$. The number of samples for which the class c_j is predicted, $[c_j \in \theta_i]$, is approximated as the sum of the jth component of the output layer of the network for all samples, $\sum_{i=1}^{m} o_i^j$.

f) $\sum_{i=1}^{m} [c_j \in \gamma_i] = \sum_{i=1}^{m} y_i^j$. The number of samples with class c_j, $\sum_{i=1}^{m} [c_j \in \gamma_i]$, is computed as the sum of the jth component of the vector of correct classes for all samples, $\sum_{i=1}^{m} y_i^j$.

g) $\sum_{i=1}^{m}[c_j \in \gamma_i \cap \theta_i] \approx \sum_{i=1}^{m} y_i^j \cdot o_i^j$. The number of samples with class c_j correctly predicted, $\sum_{i=1}^{m}[c_j \in \gamma_i \cap \theta_i]$, is approximated as the weighted sum of the jth component of the output layer and the vector of correct classes for all the samples, $\sum_{i=1}^{m} y_i^j \cdot o_i^j$.

Using these approximations, soft versions of the evaluation metrics can be defined[7]. Equations 3.10, 3.11 and 3.12 present soft versions of Accuracy (SAcc), micro-F_1 (Sm-F_1) and macro-F_1 (SM-F_1). Note that the sign has been inverted because they are loss functions that should be minimized.

$$SAcc = -\frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{j=1}^{n}\left(o_i^j \cdot y_i^j\right)}{\sum_{j=1}^{n}\left(o_i^j + y_i^j - o_i^j \cdot y_i^j\right)} \tag{3.10}$$

$$\text{Sm-F}_1 = -2 \cdot \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(o_i^j \cdot y_i^j\right)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(o_i^j + y_i^j\right)} \tag{3.11}$$

$$\text{SM-F}_1 = -\frac{2}{n} \cdot \sum_{j=1}^{n}\frac{\sum_{i=1}^{m}\left(o_i^j \cdot y_i^j\right)}{\sum_{i=1}^{m}\left(o_i^j + y_i^j\right)} \tag{3.12}$$

These functions are approximations of Acc, m-F_1 and M-F_1 with the advantage of being able to work with continuous values of o_i^j. Nevertheless, when $o_i^j \in \{0,1\}$, the evaluation metrics and their soft versions are equivalent. Soft functions have been defined to satisfy that they were differentiable functions. Eqs. (3.13), (3.14) and (3.15) show the derivatives of the three soft metrics (SAcc, Sm-F_1 and SM-F_1) with respect to o_k^l, $1 \leq k \leq m$, $1 \leq l \leq n$.

$$\frac{\partial\, \mathbf{SAcc}}{\partial\, o_k^l} = -\frac{1}{m} \cdot \frac{y_k^l \cdot \sum_{j=1}^{n}\left(o_k^j + y_k^j\right) - \sum_{j=1}^{n}\left(o_k^j \cdot y_k^j\right)}{\left(\sum_{j=1}^{n}\left(o_k^j + y_k^j - o_k^j \cdot y_k^j\right)\right)^2} \tag{3.13}$$

[7] https://github.com/jogonba2/DEVM-TC

$$\frac{\partial\,\textbf{Sm-F}_1}{\partial\,o_k^l} = -2\cdot\frac{y_k^l\cdot\sum_{i=1}^{m}\sum_{j=1}^{n}(o_i^j+y_i^j)-\sum_{i=1}^{m}\sum_{j=1}^{n}(o_i^j\cdot y_i^j)}{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(o_i^j+y_i^j)\right)^2} \tag{3.14}$$

$$\frac{\partial\,\textbf{SM-F}_1}{\partial\,o_k^l} = -\frac{2}{n}\cdot\frac{y_k^l\cdot\sum_{i=1}^{m}(o_i^l+y_i^l)-\sum_{i=1}^{m}(o_i^l\cdot y_i^l)}{\left(\sum_{i=1}^{m}(o_i^l+y_i^l)\right)^2} \tag{3.15}$$

Note that M-F_1 and m-F_1 are computed over a set of samples. We decided to use mini-batch training mode [182] in order to compute SM-F_1 and Sm-F_1 over all the samples of a given batch. However, in the mini-batch mode, it is necessary to assign a scalar for each sample in order to update the weights of the model by using the back-propagation algorithm. Therefore, we used the value of the loss function computed over a full batch as loss value for all the samples in the batch. Although it is not required, we used the same strategy for SAcc. Also, it is convenient to highlight that the batch size is a key factor in the estimation of the soft metrics, for this reason, we study, in §3.2.3, how much the batch size influences the estimation and the performance of the models.

After defining the soft approximations of the evaluation metrics, now we will detail key aspects of the experimental setting we used in the experimentation. First, regarding the preprocessing of the corpus, we applied a tokenization process by means of the TweetMotif [183] package. After, we applied a normalization step that consists of lowercasing the words, and in addition, for the Spanish language, removing some language-specific characters e.g., accents, dieresis, or "ñ". Moreover, we performed a normalization process over unicode emoticons. This process can be useful due to the great variability of the emoticons, as most of them was not included in our word embeddings. To solve this, we replaced the unicode emoticons by their short name extracted from the Unicode Common Locale Data Repository, which contains a textual description of the emoticon's shape, e.g., ☺ → "Slightly Smiling Face". It's important to notice that similar emoticons have similar textual descriptions, consequently, it makes possible to establish relationships among them by using their descriptions.

Regarding to the external resources, we used distributed representations of words, concretely, word embeddings obtained using Word2Vec [80] models, in order to consider a rich representation of each token. Moreover, we used several lexicons to consider polarity/emotion information. This polarity/emotion information was combined with the embeddings of the words. On the one hand, for the English task, we used the following lexicons: AFINN [184],

Bing Liu's Opinion [108], MPQA [185], Sentiment140 [186], SentiWordnet [187], NRC Emotion Lexicon [188], NRC Hashtag Emotion Lexicon [189] and LIWC2007 [190]. As word embeddings, we used the pretrained model by [191] with more than 400 million English tweets. On the other hand, for the Spanish task, we used the following lexicons: ELHPolar [119], iSOL [120], MLSenticon [121] and the Spanish version of NRC Emotion Lexicon. As word embeddings, we trained a skip-gram model, with 300 dimensions for each word, from 87 million Spanish tweets.

With respect to the tweet representations, we represented each tweet x as a matrix $S \in \mathbb{R}^{n\times(d+v)}$, where n is the maximum number of words per tweet, d is the dimensionality of word embeddings and v is the dimensionality of the polarity/emotion features. In order to obtain this representation, we use an embedding model $h(w) \in \mathbb{R}^d$ and a set of lexicons $h^{'}(w) = [h^{'}_1(w), h^{'}_2(w), ..., h^{'}_l(w)] \in \mathbb{R}^v$ (note that all the features extracted from lexicons, for the word w, are concatenated). Therefore, given a tweet x with n tokens, $x = w_1, w_2, ..., w_n$, we represented it as a matrix S in which, each row i is the concatenation of the embedding of w_i ($h(w_i)$) and a vector with the polarity values of w_i in each lexicon ($h^{'}(w_i)$), $S = [h(w_1)|h^{'}(w_1), h(w_2)|h^{'}(w_2), ..., h(w_n)|h^{'}(w_n)]$. If a word w_i is out of vocabulary for the embedding models, we replace its embedding by the embedding of the word "unknown", $h(w_i) = h(\text{“}unknown\text{”})$. Similarly, if w_i is not included in any lexicon, $h^{'}(w_i) = [0, 0, ..., 0] \in \mathbb{R}^v$. Given a tweet, it can belong to several classes, from the set of classes $\mathbb{C}$, in an independent way i.e. $y = \{y_1, y_2, ..., y_{|\mathbb{C}|} : y_i \in \{0, 1\}\}$ with $|\mathbb{C}| = 11$. Due to the variable length of the tweets, we used zero padding at the start of a tweet if it does not reach the maximum specified length. Otherwise, if the length of a tweet is greater than the maximum, we truncated the tweet at the end. In the English task the average length of words per tweet is $n_{avg} = 19$, and the maximum length is $n_{max} = 85$. We decided to set the length n as the mean of n_{avg} and n_{max}. In the Spanish task, $n_{avg} = 16$, $n_{max} = 82$, therefore $n = \frac{n_{avg}+n_{max}}{2} = 49$.

Concerning to the Deep Learning system, we used a Convolutional Neural Network (CNN) architecture inspired in [39]. Following the CNN architecture, we applied one-dimensional convolutions with variable height filters in order to extract the temporal structure of the tweet over several region sizes. Note that the width of the filters is constant and equal to d and we only modify the height of the filters. Fig. 3.9 summarizes the model configuration and the hyper-parameters that we used in all the experimental work.

As it can be seen, we used 6 different region sizes (the filter height ranges from 1 to 6) and 128 filters for each region size. A total of 768 different filters were used in this architecture. After applying the filters, we obtained 128 output feature maps for each region size. Note than the output feature maps have length n due to we used a "same padding" scheme. In order to extract the most salient features for each region size, we applied 1D Global Max

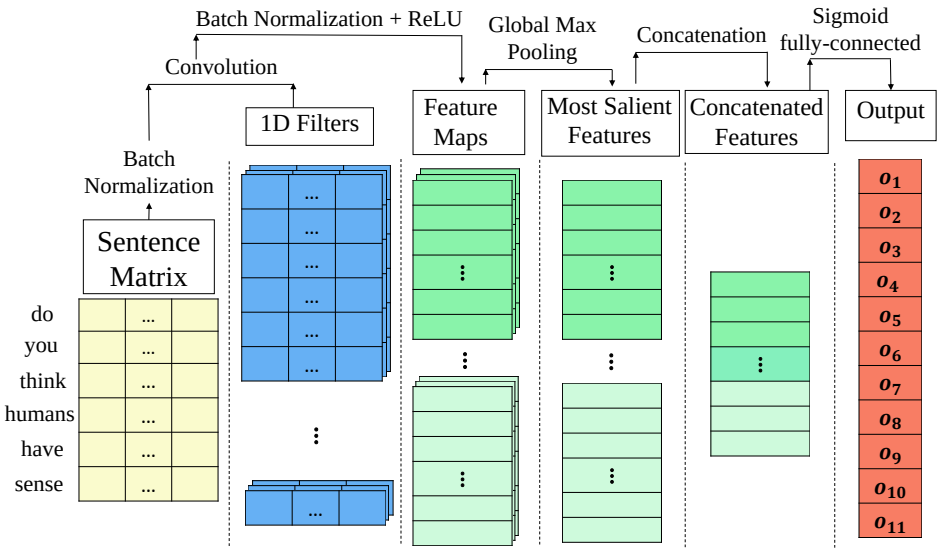

Fig. 3.9 CNN architecture for multi-label classification for *E-c: Detecting Emotions Multil-abel classification* task.

Pooling to the feature maps of each region size. Therefore, we obtained 6 vectors with 128 components, that were concatenated and used as input to a fully-connected layer which performs the multi-label classification decision. We used sigmoid activation functions to model the probability of each class independently of the probability of the other classes. Moreover, with the aim of improving the generalization and speed up the model convergence, we used BatchNormalization [51] after each convolution and after the input layer. To achieve non-linearity after each convolution, we used ReLU [192] activation functions. As optimization algorithm, we used RMSprop to learn the parameters of the network with respect to the proposed loss functions. Given the proposed architecture, $f : \mathbb{R}^{n\times(d+v)} \rightarrow \mathbb{R}^{\mathbb{C}}$, the steps to assign a set of classes to a tweet x are the following: first we obtain its representation matrix S; second, we make a forward pass in order to obtain the probability that x belongs to each class independently, $f(x) = \{o_1, o_2, ..., o_n\}$; and finally, we consider that the tweet x belongs to class j if $f(x)_j \geq 0.5$. We used this Deep Learning system for all the experiments conducted.

3.2.3 Evaluation

In this section, we present the experimental work conducted in order to evaluate the performance of the proposed loss functions. We study the impact of these loss functions on the overall results on the Emotion Classification task proposed at SemEval2018 competition.

A key parameter on the computation of the loss functions is the batch size. Therefore, we first studied the behavior of the functions by varying the batch size. This study was carried out on the development sets for English and Spanish defined in §3.2.1 (see Table 3.8). We considered 30 training epochs and the SAcc, SM-F_1 and Sm-F_1 loss functions. Fig. 3.10 shows the achieved results for English corpora. It can be seen the values of the evaluation metric per epoch on the development set for the CNN trained with the SAcc, SM-F_1, and Sm-F_1 loss functions, respectively, by varying the batch size $b \in \{16, 32, 64, 128\}$.

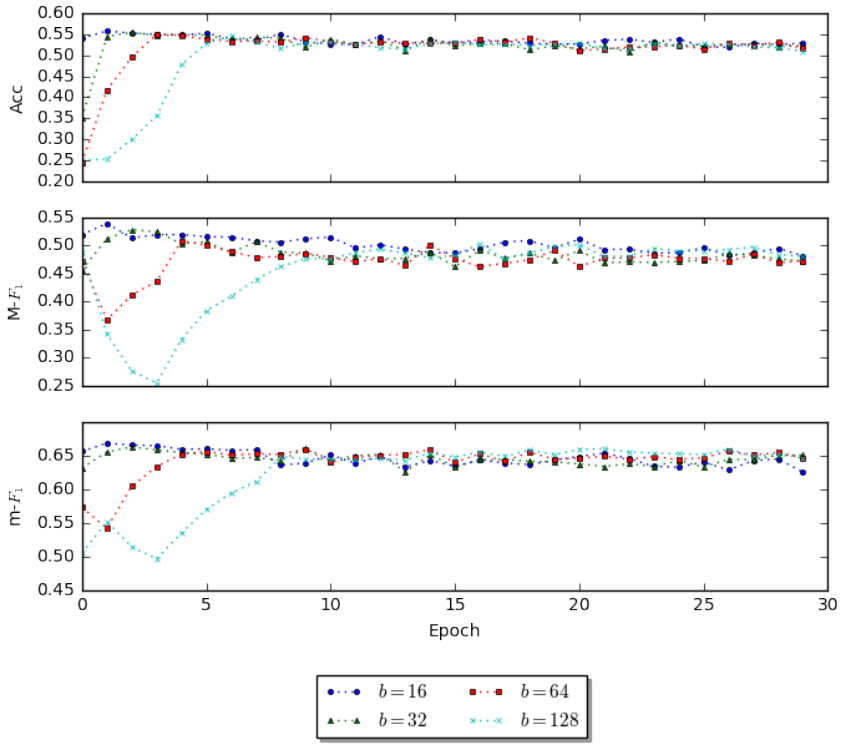

Fig. 3.10 Results of the evaluation metrics per epoch for CNN+SAcc, CNN+SM-F_1 and CNN+Sm-F_1 models varying the batch size (English development set).

As it can be seen in Fig. 3.10, in all cases, lower values of b produce faster convergence, that is, the maximum value of the evaluation metric is obtained in initial epochs. We hypothesized that, the faster convergence is due to more updates are made in each training epoch. Moreover, in all cases, the maximum value of the evaluation metric is obtained with the minimum batch size considered, $b = 16$. It can be also observed that generally, the batch sizes that obtained the best results were $b = 16$ and $b = 32$. In addition, with largest batch sizes, $b = 128$, the model takes more time to converge, but when it becomes stable, the results were similar compared to those obtained with smaller batch sizes. We performed a similar experimentation for the Spanish corpora and we have observed a similar behavior than in the

English corpora, that is, lower batch sizes produce faster convergence. From the performed study of how influence the batch size on the loss function optimization, we can conclude that the best size is $b = 16$, both for the English and Spanish data sets.

Once the best value of the batch size was set, our objective was to compare the performance of the system trained using the proposed loss functions, with the performance obtained using BCE and MSE as loss functions, on the test sets of the Emotion Classification task. To make this comparison more accurate, the confidence intervals of the three metrics used to evaluate the systems in the competition were computed. In order to compute these confidence intervals, we used the Bootstrap Confidence Intervals [193] approach. First, from the set of hypotheses provided by the system that we want to evaluate, we generated up to 5000 resamples by sampling with replacement from this original set of hypotheses. Each resample had the same size of the original set. Next, the value of the evaluation measure is calculated for each of the resamples. Finally, we compute the 95% confidence interval using the bootstrap distribution. Tables 3.9 and 3.10 show the results obtained by our systems both for the English and the Spanish corpora

Table 3.9 Results on the English test set

Loss	**Acc**	**M-F_1**	**m-F_1**
SAcc	**56.39±1.11**	49.72±1.15	67.32±0.96
SM-F_1	54.85± 1.09	**54.73±1.02**	66.75±0.97
Sm-F_1	55.44± 1.11	50.77±1.08	**67.60±0.95**
BCE	52.03±1.12	50.47±1.25	64.44±0.97
MSE	52.32±1.11	49.59±1.17	64.65±0.97

Table 3.10 Results on the Spanish test set

Loss	**Acc**	**M-F_1**	**m-F_1**
SAcc	**47.33± 1.43**	42.06±1.64	54.82±1.31
SM-F_1	45.26±1.40	**45.25±1.57**	**55.10±1.30**
Sm-F_1	44.20±1.47	42.34±1.66	54.83±1.34
BCE	44.08±1.41	40.42±1.51	52.70±1.34
MSE	44.22±1.44	41.28±1.52	54.10±1.33

It can be seen that the models trained with the proposed loss functions obtained the best results. This occurs both for English and Spanish test sets with significant improvements, compared to BCE and MSE, in all cases except when evaluating with m-F_1 on the Spanish test set. In addition, it can be observed how, generally, the model trained with the loss function that approximates the evaluation metric used, obtained the best results for that evaluation metric. This is true for all the cases studied, except for the m-F_1 measure for

Spanish in which it achieved the second-best result (note that this is the only case where the improvements with respect to BCE and MSE are not significant). Compared with the official results of the competition[8], our best result achieved the 1st position of 13 teams for Spanish and the 7th place of 34 teams for English.

3.2.4 Analysis

In this subsection, we perform two different analyses to study the system performance per class and how the loss functions approximate the evaluation metrics. First, we present the analysis of the performance of our system at class level. Tables 3.11 and 3.12 show the results of P, R, and F_1 for all classes with the CNN trained using SM-F_1 and Sm-F_1 as loss function respectively.

Table 3.11 Precision, Recall and F_1 per class, for English test set, with the models trained with SM-F_1 and Sm-F_1

	SM-F_1			**Sm-F_1**		
Emotion	P	R	F_1	P	R	F_1
Anger	72.41	81.29	76.59	68.29	85.10	75.78
Anticipation	26.80	38.59	31.63	35.67	14.35	20.47
Disgust	66.06	79.53	72.17	62.97	82.62	71.47
Fear	69.40	64.54	66.88	70.83	63.09	66.74
Joy	82.06	84.05	83.04	79.35	86.89	82.95
Love	49.67	73.06	59.14	57.76	56.98	57.37
Optimism	65.14	75.85	70.09	65.42	76.47	70.51
Pessimism	33.91	42.13	37.57	41.43	27.73	33.23
Sadness	64.57	71.56	67.89	61.78	72.92	66.89
Surprise	16.14	24.12	19.34	52.63	5.88	10.58
Trust	11.36	40.52	17.74	22.22	1.31	2.47
Macro Average	50.68	61.39	54.73	56.22	52.12	50.77
Micro Average	59.35	71.80	64.99	66.02	69.25	67.60

From the results for the English corpus, it can be observed that the system trained with SM-F_1 achieved higher values of F_1 for the minority classes, e.g., Trust ($F_1 = 17.74$ on development and $F_1 = 24.79$ on test) and Surprise ($F_1 = 19.34$ on development and $F_1 = 26.02$ on test), compared to those obtained by the system trained with Sm-F_1 (for Trust: $F_1 = 2.47$ on development and $F_1 = 15.19$ on test and for Surprise: $F_1 = 10.58$ on development and $F_1 = 16.56$ on test). This occurs because SM-F_1 considers the performance of all the classes in the same way, thus penalizing the model when it does not perform well

[8] `https://competitions.codalab.org/competitions/17751#results`

Table 3.12 Precision, Recall and F_1 per class, for Spanish test set, with the models trained with SM-F_1 and Sm-F_1

	SM-F_1			**Sm-F_1**		
Emotion	*P*	*R*	F_1	*P*	*R*	F_1
Anger	66.91	69.75	68.30	68.53	68.01	68.27
Anticipation	45.68	23.05	30.64	42.27	25.55	31.84
Disgust	46.96	38.30	42.19	48.81	34.04	40.11
Fear	64.15	45.64	53.33	60.09	43.96	50.78
Joy	80.30	72.85	76.40	81.19	71.71	76.16
Love	65.96	50.61	57.27	70.67	43.27	53.67
Optimism	32.02	29.14	30.51	43.81	16.55	24.02
Pessimism	38.94	19.19	25.71	38.52	18.99	25.44
Sadness	65.53	59.94	62.61	65.95	61.65	63.72
Surprise	25.81	26.23	26.02	37.14	10.66	16.56
Trust	25.00	24.59	24.79	33.33	9.84	15.19
Macro Average	50.66	41.75	45.25	53.67	36.75	42.34
Micro Average	60.12	50.57	54.93	63.90	48.02	54.83

on the minority classes. In contrast, Sm-F_1 favors majority classes, as the results are typically dominated by the performance on these classes. A similar behavior can also be observed in the results for the Spanish corpus. This is an important aspect to keep in mind in the design of a classification system.

Finally, an analysis of how the loss functions approximate the evaluation metrics is presented. For the English corpus, Fig. 3.11 shows loss function and the evaluation metric on training, development and test sets for the CNN trained using SAcc, SM-F_1, and Sm-F_1, respectively.

It can be observed how, throughout all the training epochs, the proposed loss functions follow the same trend as the evaluation metrics. This seems to indicate that they are good approximations to the evaluation metrics and consequently, we can make decisions about the evaluation metrics by looking only at the values obtained by the loss functions. Another interesting aspect to note is that the loss functions can be considered as a lower bound of the evaluation metrics. This can be observed in all the previous cases, where, for the different epochs, the loss functions do not overestimate the values of the evaluation metric. In addition, it draws attention how SM-F_1, although it correctly estimates the trend of the metric M-F_1 throughout the different epochs, obtains values much smaller than these ones. On the other hand, the SAcc and Sm-F_1 loss functions, follow the trend of the evaluation metrics that they approximate, and they obtain values similar to those obtained by those metrics. A similar behavior was observed in all the partitions for Spanish language.

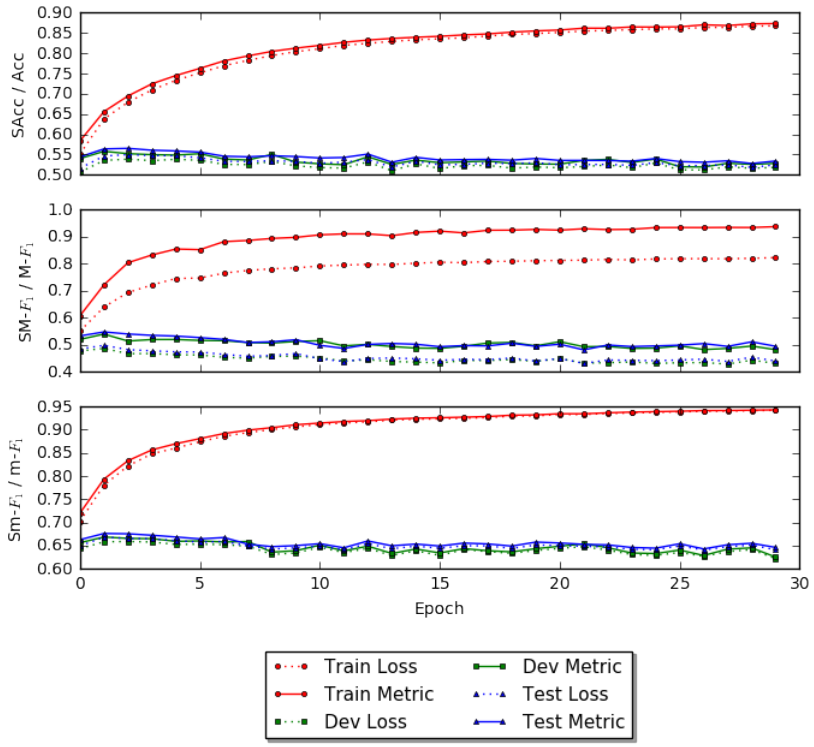

Fig. 3.11 Loss function and evaluation metric per epoch, on English training, development and test sets using CNN+SAcc, CNN+SM-F_1 and Sm-F_1, respectively.

3.3 Irony Detection

Human communication using natural language in social environments is influenced by the use of figurative language. Unlike literal language, where the meaningful units used in the interactions convey exactly the meaning that the author wants to express, figurative language aims to use words differently from the usual way, in order to transmit complex ideas in a more creatively. One of the most interesting rhetorical devices is the irony. Irony takes place in ambiguous situations where the literal meaning is opposite to the knowledge that the author has of the world and it is wanted to be transmitted [194]. Irony has been extensively studied in the pragmatics field, as it cannot be inferred only by an interpretation of the meaning in isolation, but it requires complex reasoning to understand the situations in which an utterance was made. In that sense, irony has a social inherent nature, and it can be identified inside of the pragmatics theories. Nowadays, the irony is extensively used in social networks to favor social interactions, evoking humor [194], diminishing or enhancing criticism [195], and getting the attention of the readers through the creativity [196].

The New Princeton Encyclopedia of Poetry and Poetics [197] identify eight different types of irony: classical, romantic, tragic, cosmic, verbal, situational, dramatic, and poetic

irony. The most common types of irony used in social networks are situational and verbal irony. On the one hand, situational irony is related to incongruous situations about specific events e.g., "A security company is the last victim of a malware attack", which is unexpected because one would assume the security companies are safe against malware attacks. On the other hand, verbal irony has been defined by several authors [194, 198, 199] as the communication of a meaning opposite to the literal meaning, e.g., "Oh look, we are having another storm in Sydney. How unusual.". A specific form of verbal irony is the sarcasm, which also has been studied in the literature [200]. It is a subset of verbal irony where a message aims to make a harmful criticism about someone or something.

The detection of the irony in text messages is a complex and subjective problem affected by a plethora of phenomena. Several relevant features have been identified to address the irony detection problem: polarity contrast [201], common sense knowledge [202], similes with "about" or "as" structures [196], punctuation marks or repetitions [203], affective features [204, 205], negation [206], contextual features [207], context incongruity [208], etc. However, computational approaches that follow the principle of text compositionality are not capable of explaining the textual irony only by means of the composition of the words of a message [209], mainly because the meaning of ironic messages is not literal, it cannot be interpreted in isolation without a context, and many irony markers are lost in the text message, such as kinesthetic (facial or hand gestures) [210] or speech features (voice tone, rhythm, silences, etc.) [211].

Irony also has a great impact on some computational approaches for NLP tasks in social media such as sentiment analysis [138, 142, 200], author profiling or deception detection [212], where the systems struggle if they are applied to ironic content. This impact is inherently related to the non-literal nature of the irony, as the current systems used for addressing these tasks rely on word correlations that assume literal meaning. In order to boost the research on irony detection for several languages, different workshops have been organized [139, 146, 213] to improve the understanding of the irony and to diminishing the faults of the computational approaches for NLP tasks when they are applied to ironic content.

Several workshops have been organized to address the irony detection problem for different languages such as Spanish [146], English [139], Italian [213] and Arabic [214].

For the Spanish language, the IroSVA shared task [146] was proposed within the 35th International Conference of the Spanish Society for Natural Language Processing (SEPLN). IroSVA aimed to identify the presence of the irony in tweets for three Spanish variants. A peculiarity of this task is that each tweet of the corpus has an associated context that consists of a short sequence of words that identify the scope of the tweet, e.g., "flat earth" or "book of Pedro Sánchez" (referencing the controversial book written by the Spanish prime minister).

Among the systems proposed by the participants for the task, the two best systems were based on Deep Learning approaches either as classifiers or as feature extractors. The best system was presented by our team [24]. It was based in the use of Transformer encoders, relying on multi-head scaled dot-product attention mechanisms, in order to contextualize pretrained Twitter word embeddings. A formalization of the model along with an extensive evaluation and result analysis both for Spanish and English languages is the scope of the current work. The second-ranked team in the IroSVA competition [215] experimented with the early fusion of traditional features (TF-IDF weighted n-grams) and distributed features (pretrained word embeddings and the internal representation of a pretrained LSTM for the task). As classifiers, they used Support Vector Machines (SVM) and Multi-Layer Perceptron on top of the input features. Contrary to these two approaches, the third most competitive approach [216] does not rely on Deep Learning architectures. In this case, the authors were interested in observing how several dependency-based features contribute to the irony detection. Concretely, they proposed the use of bag of dependency relations, bag of syntax paths, and bag of dependency relations to train Random Forests and SVM models.

For the English language, the task 3 of SemEval 2018 [139], co-located with the North American Chapter of the Association for Computational Linguistics (NAACL), aims to boost the work on irony detection on English tweets. In this workshop, two different subtasks were proposed. The first subtask consists in addressing the irony detection as a binary classification problem, whereas for the second subtask, the participants should distinguish among three different types of irony: verbal irony by means of polarity contrast, other verbal irony, and situational irony. Most of the participants addressed both subtasks by using Deep Learning approaches. The best system [217] was based on the use of Densely connected Bidirectional LSTM (D-BiLSTM) on top of a combination of word embeddings with Part of Speech Tags. Moreover, the system used a late fusion of the D-BiLSTM representations, several sentiment features (generated via the AffectiveTweets package of Weka), and a vector representation of the tweets generated by averaging the word embeddings. The system was trained to simultaneously solve three tasks, the two subtasks of the competition together with a hashtag prediction task. The authors of the second-best-ranked system [218] proposed an ensemble of two Attentional LSTM (Att-LSTM) which share the same architecture but operate on two different representation levels: words and characters. Both networks are only different on the first embedding layer. For the word level, the embedding layer was initialized with pretrained word representations learned from 550M English tweets. Regarding the character level, the embedding layer was randomly initialized and learned during the training of the model for the subtask of irony detection. In order to perform the ensemble of the two Att-LSTM, the authors tested two different approaches: unweighted average and majority voting. The

third best ranked work [219] studied how the sentiment, distributional semantics, and text surface features were related to the irony. The main effort of their work relies on detecting the polarity contrast at two different levels: polarity contrast between the same element of a tweet e.g., antithetical fragmented hashtags, and polarity contrast between two different elements of a tweet e.g., words and emojis sentimentally opposed. Also, they detected that in most ironic tweets, negative polarity is preceded by neutral or positive polarity. Therefore, they decomposed the tweets to take also into account these temporal relations. Both the polarity contrast and the surface features were combined with word embeddings and they were used as input to an ensemble soft voting classifier based on Logistic Regression and SVM paradigms. It is interesting to note that, most of the participating teams addressed the tasks by using emotional and polarity features in order to enrich their systems with the aim of explaining the irony by means of polarity contrast [25, 220, 221].

In addition to the works proposed in conference tasks, a lot of efforts have been made in order to analyze relevant features for irony detection. The most studied phenomenon is the impact of the polarity in the irony detection problem [143, 204, 222]. Also, the work presented in [219] shown that, in a certain context, too much of an emotion can imply the opposite sentiment, generating some kind of irony. Some works are focused on detecting implicit incongruencies among positive and negative words [141, 223]. In [141], the authors enrich the supervised learning on irony detection tasks by transferring knowledge from sentiment resources. They proposed three different Att-LSTM approaches that differ in the way of including the sentiment resources, either injecting the sentiment directly to the attention mechanisms or merging the output of different networks specialized on sentiment analysis and irony detection. In [223], the authors focused on identifying contrasting contexts, that is, positive sentiment followed by a negative situation. They learned a list of positive and negative phrases, using a bootstrapping algorithm, that are used for recognizing sarcasm in tweets.

Recently, pretrained contextualized BERT embeddings [57, 82, 83] become ubiquitous in many text classification tasks, and they have been progressively applied to the irony and sarcasm detection problems [224–227]. In [225], the authors finetuned the multilingual BERT for the IroSVA task and they compare the results with classical techniques for text classification such as SVM and Gradient Tree Boosting. In [226], the authors make a further pretraining of the multilingual BERT model with the Twitter domain, and they finetune the models under a multi-task setup for addressing irony detection, author profiling, and emotion detection in Arabic tweets. In [227], the sarcasm detection problem is addressed by using multimodal information such as speech, videos, and text. Pre-trained BERT was used to represent the textual utterances, showing a better performance than other strategies

such as averaging GloVe word vectors [228]. In [224], the pretrained RoBERTa [83] model was used to represent the sentences, that were further contextualized by means of a Recurrent Convolutional Neural Network to address irony and sarcasm detection. All these previous works are based on using pretrained BERT models either for finetuning them or for extracting sentence representations. However the use of the main mechanism of BERT, that is Transformers, has not been explored for contextualizing pretrained word embeddings in irony detection tasks. By this way, our work differs from them because it does not use the contextual representations learned from BERT, and instead, it is based on the backbone network of the BERT models (Transformer Encoders) for contextualizing Word2Vec word embeddings pretrained on the task domain (Twitter).

In this thesis, we propose the use of the Transformer architecture in order to contextualize pretrained word embeddings. Specifically, we contextualize Word2Vec word embeddings, trained with several millions of tweets both for the English and the Spanish languages. This strategy, opposite to the use of pretrained BERT, allows our system to be trained from in-domain representations using the same powerful backbone architecture as BERT. We evaluated the adequacy of our proposal on two corpora. For the Spanish language, the corpus of the Irony Detection on Spanish Variants shared task (IroSVA) [146] was used. For the English language, we used the dataset of the task 3: Irony Detection in English Tweets proposed in 2018 at the 12^{th} International Workshop on Semantic Evaluation (SemEval) [139]. Our system was the first-ranked system in the IroSVA competition and, to our knowledge, it has achieved the second-best result on the SemEval corpus. The implementation of this system is freely available, under request, for research purposes. Additionally, several analysis and algorithms are proposed in this section, in order to determine how the multi-head self-attention mechanisms of the Transformer are specialized on detecting ironic messages, with the aim of observing how the polarity, the relevance of individual words and the relationships among words, influence the irony detection problem. The main objectives of this section are: to study the irony detection problem for the English and the Spanish languages on two widely used corpora (§3.3.1); to present an approach based on Transformer Encoders for contextualizing pretrained Twitter word embeddings (§3.3.2 and §3.3.3); and to propose several analysis strategies towards the understanding of the behavior of Transformer Encoder models and the features captured by them when addressing the irony detection problem e.g., word polarity and relationships among words (§3.3.4).

3.3.1 Corpora

In order to validate our proposal for irony detection on Twitter, we evaluated it using two different corpora, one for the Spanish language and another for the English language. They

have been extensively used with the aim of training and evaluating state-of-the-art systems in both languages for irony detection tasks.

Regarding the Spanish language, we used the corpus provided in the IroSVA shared task [146] for training and evaluating our proposal. The IroSVA shared task, framed in the Iberian Languages Evaluation Forum (IberLEF) and co-located within the SEPLN, aims of determining if a tweet is ironic or not. Three different corpora with tweets from Spain, Mexico, and Cuba were provided by the IroSVA organization. A context of the tweets is also provided, that consists of a short sequence of words that identifies the scope of each tweet, e.g., flat earth or Mexico government. However, this kind of context does not give complementary information about the tweets, beyond identifying topics that are prone to be the object of irony. It is also important to note that, due to the fact that the organizers considered a specific context or event to build the corpus, tweets that seem to be non-ironic become ironic when considering external knowledge about its context.

The corpus was composed by 2400 training samples and 600 test samples for each Spanish variant. During the training phase, to adjust the models, from the original training set of the competition, we generated new training and development sets following an 87.5%-12.5% proportion for maintaining the relation of 2:1 between the non-ironic and ironic classes such as in the original training set. During the test phase, we used the original test set provided by the competition organizers. The size of each set is shown in Table 3.13.

Table 3.13 Corpus statistics for the ironic (I) and the non-ironic (No-I) classes

Corpus	Variant	Training		Development		Test	
		No-I	I	No-I	I	No-I	I
IroSVA	Spain	1400	700	200	100	400	200
	Mexico	1400	700	200	100	401	199
	Cuba	1400	700	200	100	400	200
SemEval	English	1544	1509	372	392	473	311

The official evaluation metrics proposed by the organizers were Precision, Recall, and F_1 in order to assess the performance of the systems. Due to the imbalance between the non-ironic and ironic classes, the macro-averaged F_1 measure was used to rank the participating systems.

For the English language, we used the corpus of the "Irony Detection in English Tweets" shared task [139] proposed in SemEval. The corpus was collected by crawling tweets with hashtags that indicate the presence of irony such as #irony, #sarcasm, and #not during one month. Following this process, a total amount of 4792 tweets were collected (2396 ironic tweets and 2396 non-ironic tweets). Training and test sets, following an 80%-20% proportion, were provided to the participants. It is important to highlight that the test set was modified

later by the organizers in order to remove some ironic samples that require context to be understood. From this corpus, two different subtasks were proposed. The first one consists in addressing the irony detection as a binary classification problem. The second one, consists in distinguishing among three different types of irony: verbal irony by means of a polarity contrast, other verbal irony, and situational irony. We only focused on the first subtask, that is the most related with the IroSVA shared task for the Spanish language. However, unlike IroSVA, most of the ironic messages do not require a context to be understood and they are based on conveying opposite meanings. In order to carry out the experimentation, we split the original training partition into training and development partitions, following an 80%-20% proportion. The statistics of each partition are also shown in Table 3.13. For evaluation purposes, standard evaluation metrics such as Precision, Recall and F_1 were also used. Concretely, in this case, the organizers consider the F_1 measure of the ironic class in order to rank the participating systems.

Table 3.12 shows some *ironic* and *no-ironic* examples for the SemEval and IroSVA corpora. It can be seen how the *ironic* example from IroSVA requires deeper understanding about the context than the *ironic* sample from SemEval, as in the IroSVA case, it is required very specific knowledge (about the book of the Spanish Prime Minister, Pedro Sánchez, and the plagiarism accusations of his thesis), while in the SemEval case, it is required to detect polarity contrast (rain/sleet/work and fun).

Example 1 (SemEval): Rain and sleet hun? Yeah I totally want to get dress and go to work. Sounds like fun #iwannagobacktobed (*ironic*)

Example 2 (SemEval): Simple way to be #fashionable n contribute to #empower thousands of #women and #weavers (*no-ironic*)

Example 3 (IroSVA): @sanchezcastejon Seguro que ha escrito él el libro,como la tesis. [@sanchezcastejon Surely he wrote the book,like the thesis.] (*ironic*)

Example 4 (IroSVA): Grave maniobra de Sánchez o estrategia ganadora con la figura del relator [Serious maneuver by Sánchez or winning strategy with the figure of the rapporteur] (*no-ironic*)

Fig. 3.12 Examples from the training set of the SemEval and IroSVA corpora. English translation is also considered for the Spanish examples.

3.3.2 Proposed Approach

In this subsection, we present the transformer architecture, along with its hyper-parameters, the resources and the preprocessing we used for performing the experimentation. As in our work on sentiment analysis, in this work we also focused on the Transformer Encoders

(TE), that reduce the computational complexity per layer and the maximum path length of dependencies among words to $\mathcal{O}(1)$, instead of $\mathcal{O}(\log n)$ or $\mathcal{O}(n)$ in the cases of convolution and recurrent mechanisms, respectively. Figure 3.4 shows the scheme of the proposed architecture.

Let $\mathbb{C} = \{0,1\}$ be the set of classes (0 denotes the non-ironic class and 1 denotes the ironic class), $X = \{x_1, x_2, ..., x_T \ : x_i \in \{0, ..., V\}\}$ be the input of the model where T is the maximum length of the tweet, $y \in \mathbb{C}$ the ground-truth of sample X, and V is the vocabulary size. This tweet is passed through a d-dimensional pre-trained embedding layer, E, frozen during the training phase, that is dependent on the language of the corpora used. Moreover, to consider positional information we also experimented with the sine-cosine function proposed in [11], defined in Eq. 3.16.

$$P_{(pos,2i)} = sin\left(\frac{pos}{1000^{\frac{2i}{d}}}\right) \quad P_{(pos,2i+1)} = cos\left(\frac{pos}{1000^{\frac{2i}{d}}}\right) \tag{3.16}$$

where *pos* is the position and *i* is the dimension. This heuristic exploits the cyclic nature of sine and cosine functions to represent the positional information of the words in a text. Furthermore, unlike learned positional embeddings [78], it is able to generalize to unseen lengths and it does not require parameters to learn the positional information, with a negligible computational overhead before training the models. This positional information, encoded as $P \in \mathbb{R}^{T \times d}$, is added to the embedding representation of the tweet, $X^0 \in \mathbb{R}^{T \times d}$, to be used as input to the first encoder layer as shown in Eq 3.17.

$$X^0 = \{\underbrace{P_1 + E(x_1)}_{X_1^0}, ..., \underbrace{P_T + E(x_T)}_{X_T^0} \ : X_i^0 \in \mathbb{R}^d\} \tag{3.17}$$

After the combination of the word embeddings with the positional information, dropout [49] is used to drop input words with a certain probability p to regularize the model. On top of these representations, N transformer encoders are applied, which rely on the multi-head scaled dot-product attention shown in Eqs 3.18 to 3.20.

$$MultiHead(A,B,C) = [head_1; ...; head_h]W^O \tag{3.18}$$

$$head_i = Attention(AW_i^Q, BW_i^K, CW_i^V) \tag{3.19}$$

$$Attention(Q,K,V) = softmax(\frac{QK^\intercal}{\sqrt{d_k}})V \tag{3.20}$$

where $W_i^Q \in \mathbb{R}^{d \times d_k}$, $W_i^K \in \mathbb{R}^{d \times d_k}$, $W_i^V \in \mathbb{R}^{d \times d_k}$, $W^O \in \mathbb{R}^{h \cdot d_k \times d}$, are the projection matrices for query (Q), key (K) and value (V) of the head i and for the output of the multi-head attention respectively; h is the number of heads for the multi-head attention mechanism; and $head_i \in \mathbb{R}^{T \times d_k}$ is the output of the head i. The output for only one encoder, S, is computed as shown in Eq 3.24 for a given sample X^0.

$$M = MultiHead(X^0, X^0, X^0) \tag{3.21}$$

$$L = LayerNorm(X^0 + M) \tag{3.22}$$

$$F = max(0, LW_1 + b_1)W_2 + b_2 \tag{3.23}$$

$$S = LayerNorm(L + F) \tag{3.24}$$

where $M, L, F \in \mathbb{R}^{T \times d}$ are the intermediate outputs from the encoder, $W_1 \in \mathbb{R}^{d \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d}$ are the weights of the position-wise feed forward network, $S \in \mathbb{R}^{T \times d}$ is the output of the encoder, and *LayerNorm* denotes Layer Normalization [52]. When several encoders are stacked, the input of a encoder is directly used as input to the next encoder. Due to a vector representation is required to train classifiers, on top of the output of the last encoder, a global average pooling was applied on S. The pooled vector, $G \in \mathbb{R}^d$, was used as input for a single-layer feed-forward network, whose output layer computes a probability distribution over the two classes of the task $\mathbb{C} = \{0, 1\}$, as shown in Eq. 3.25.

$$O = softmax(max(0, GW_3 + b_3)W_4 + b_4) \tag{3.25}$$

where $O \in \mathbb{R}^{|\mathbb{C}|}$ is a probability distribution over $\mathbb{C}$, $W_3 \in \mathbb{R}^{d \times d_o}$ is the weight matrix of the hidden layer applied on top of G and $W_4 \in \mathbb{R}^{d_o \times |\mathbb{C}|}$ is the weight matrix of the output layer. Due to the imbalance in all the corpora used for the experimentation, weighted cross entropy is used as loss function for training the network, considering the distribution of each class in the training set. This is shown in Eq. 3.26, where $\mathscr{D}$ is the dataset, $\mathscr{L}$ is the loss function and f is our model parameterized by θ. Concretely, we used the proportion between the most frequent class and the frequency of a given class, $w_j = \frac{\max_{c \in \mathbb{C}} N_c}{n_j}$, where N_j is the number of samples of the class j in a given set, being $w_j = 1$ if j is the most frequent class, and $w_j > w_k$

if class j is less frequent than the class k in the sample set. We used Adam [55] as update rule and Noam [11] as learning rate schedule.

$$\mathscr{L}_{\mathscr{D}}(\theta) = -\frac{1}{n}\sum_{i=1}^{N}\sum_{j=1}^{|\mathbb{C}|} y_{ij}\ log\ f(x_i;\theta)_j\ w_j \tag{3.26}$$

Following the experimental setup proposed in [11], for both tasks we fixed most of the hyper-parameters with the aim of minimizing the impact of the hyper-parameter tuning when comparing our proposal with other state-of-the-art systems. Specifically, $d_o = 512$, $h = 8$, $d_k = d_o/h = 64$ and $d_{ff} = d$ as stated in [11]. We defined $batch\_size = 32$ and $T = 50$ in order to be slightly higher than the maximum length of the tweets in the training set. Also, as in [11], we used the Adam update rule [55] with $lr = 0.001, \beta_1 = 0.9$ and $\beta_2 = 0.999$, and Noam learning rate schedule with $warmup\_steps = 15$ epochs. We limited the depth of the Transformer Encoder to only one layer due to the limited number of samples on both corpora available to train the models. In the training step, early stopping, with a patience of 20 epochs, was used as stopping criterion.

In order to incorporate task-related knowledge to our model, we initialized the embedding layer with non-contextualized pretrained word representations. These representations are highly dependent on two main aspects: the domain and the language. Regarding the domain, we only used pretrained representations of words that appeared on tweets from the social network Twitter. With respect to the language, we used two different word embedding models, one for each language.

For the English language, we used the pretrained word embeddings provided in [191]. These embeddings were trained by the authors of [191] using 400M English tweets collected from 1/3/2013 to 28/2/2014. Moreover, they determined the best values for some hyper-parameters such as the dimensionality and the topology. The result of their experimentations was a 400-dimensional skip-gram model which we used directly in our proposal. For the Spanish language, we decided to use the same architecture than [191] with a slightly lower dimensionality due to the difference between the number of samples to train the model. In this case, the pretrained representations were extracted from a 300-dimensional skip-gram model. This model was trained in our laboratory by using 87M Spanish tweets from several Spanish variants. We downloaded these tweets by means of a Twitter streamer, listening for Spanish tweets (including retweets) that contain several common Spanish words such as "que", "de" and "donde". The stream process was performed from 1/6/2017 to 1/7/2017. The competitive behavior obtained by both word embedding models have been proven in several text classification tasks [16–18, 22, 25].

Regarding the preprocessing, firstly, a case-folding process was applied to all the tweets, secondly, we tokenized the tweets by using the TokTokTokenizer from NLTK [147]. Thirdly, user mentions, hashtags, and URLS were replaced by three generic-class tokens (*user*, *hashtag* and *url*, respectively). Finally, elongated tokens are diselongated allowing the same vowel to appear only twice consecutively in a token (e.g., *jaaaa* becomes *jaa*).

3.3.3 Evaluation

In this subsection, an exhaustive evaluation of the proposed approach is presented. The performance of our system, based on TE, is compared with other deep learning systems, such as Deep Averaging Networks (DAN) [53] and Att-LSTM [229]. This comparison is only performed on the development set, while in the test set, the results of this proposal are compared against the systems of other participants. It is important to note that DAN, TE and Att-LSTM implement a pooling strategy based on averaging, either an unweighted average such as DAN and TE or a weighted average such as Att-LSTM. We used the word embeddings described in the previous section to train all the models. Another interesting aspect to take into account for irony detection is the positional information. It is intuitive to think that this information is useful for detecting the irony, due to the sequentiality is a relevant factor for some types of irony e.g., irony by means of polarity contrast. For this reason, the effect of the positional information in the results is also studied in our experimentation. Specifically, two different TE models are tested, one with sine-cosine positional information (TE-Pos) and another one without this positional information (TE-NoPos). For all the experimentation, Precision, Recall and F_1 for the two classes, along with their macro-averaged version of F_1 (MF_1) were considered.

The results obtained for IroSVA task on the development set in the three Spanish variants are shown in Table 3.14. This table only shows the results of the best two systems for the Spain (SP) variant, for the Mexico (MX) and Cuba (CU) variants, due to all the other systems obtain worse results than them in MX and CU.

It can be seen in Table 3.14 that simpler models (DAN and TE-NoPos), with less parameters and without positional information, obtained the best results for all the evaluation metrics. Concretely, for the variant from Spain, the best results were obtained by TE-NoPos, although DAN outperformed it in terms of $P(1)$ and $R(0)$. For the other two variants (Mexico and Cuba), the TE-NoPos system also achieved the best results outperforming those obtained by DAN model for all the evaluation metrics.

Table 3.15 shows the results obtained on the development set of the SemEval task for the English language. Note that, all the systems are biased towards the ironic class and all

Table 3.14 Results on the IroSVA development set for the three Spanish variants.

System	$P(0)$	$P(1)$	$R(0)$	$R(1)$	$F_1(0)$	$F_1(1)$	MF_1
				Spain			
DAN	84.13	**72.83**	**87.50**	67.00	85.78	69.79	77.78
Att-LSTM	84.32	61.74	78.00	71.00	81.05	66.05	73.54
TE-NoPos	**88.77**	69.91	83.00	**79.00**	**85.79**	**74.18**	**79.98**
TE-Pos	83.33	62.96	80.00	68.00	81.63	65.38	73.51
				Mexico			
DAN	80.11	55.26	74.50	63.00	77.20	58.88	68.04
TE-NoPos	**82.35**	**59.29**	**77.00**	**67.00**	**79.59**	**62.91**	**71.25**
				Cuba			
DAN	75.83	55.06	80.00	49.00	77.86	51.85	64.85
TE-NoPos	**82.83**	**64.71**	**82.00**	**66.00**	**82.41**	**65.35**	**73.88**

of them obtained better results in terms of the $F_1(1)$ compared to the $F_1(0)$. The Att-LSTM system is the system that shows the most balanced behavior between both measures.

Opposite to IroSVA, Att-LSTM obtained the best results in almost all the metrics, although TE-Pos outperformed it for $P(0)$ and $R(1)$. The differences between both versions of TE are around 5 points for the MF_1 measure in favor of TE-NoPos. DAN and TE-NoPos obtained similar results of MF_1 measure, but they are not the best models. Generally, the conclusion for the English corpus is different to the conclusion for the Spanish one: the most complex model Att-LSTM (with more parameters and considering positional information by its internal memory) shows the best behavior. Nevertheless, due to the fact that TE-NoPos outperforms TE-Pos system in both corpora, it seems that the use of positional information in the Transformer architecture was not useful for the corpora considered. A deeper analysis would be necessary to determine if the negative results, achieved when including positional information, are due to the positional information itself or to the way in which this information is included in the model.

Table 3.15 Results on the SemEval development set for the English language.

	$P(0)$	$P(1)$	$R(0)$	$R(1)$	$F_1(0)$	$F_1(1)$	MF_1
DAN	75.34	62.29	45.16	85.97	56.47	72.24	64.36
Att-LSTM	72.20	**67.17**	**59.41**	78.83	**65.38**	**72.54**	**68.96**
TE-NoPos	72.91	63.16	49.19	82.65	58.75	71.60	65.18
TE-Pos	**76.44**	59.49	35.75	**89.54**	48.72	71.49	60.10

Now, the results on the test set of both tasks are presented. The results on the Spanish IroSVA corpus of our TE-NoPos model were published in [146]. Table 3.16 shows the results, for all Spanish variants, of the best participating teams in the IroSVA competition ranked

according the official evaluation measure (MF_1) average for all the Spain variant. Our system outperformed the second-ranked system (CIMAT) by almost to 3 points in average for all the Spanish variants.

Table 3.16 Results on the IroSVA test set in terms of MF_1. Our system is marked with †.

System	Spain	Mexico	Cuba	AVG
TE-NoPos†	**71.67**	**68.03**	65.27	**68.32**
CIMAT	64.49	67.09	**65.96**	65.85
LDSE	67.95	66.08	63.35	65.79
JZaragoza	66.05	67.03	63.35	64.90
W2V	68.23	62.71	60.33	63.76
ATC	65.12	64.54	59.41	63.02

Regarding the SemEval task, two different systems on the test set were evaluated. The TE-NoPos system proposed in this work and the Att-LSTM system that obtained the best results on the development set. These results have been obtained after finishing the competition. Table 3.17 shows the results of both systems along with those obtained by the best participating teams in the competition. The systems are ranked according to the official evaluation measure ($F_1(1)$). It is interesting to observe that, although in the validation set, the best system in terms of $F_1(1)$ is Att-LSTM, on the test set, TE-NoPos outperforms it. The best results of $F_1(1)$ obtained by TE-NoPos in comparison to Att-LSTM are due to the increment up to 10 points of $R(1)$ while both systems maintain similar precision on the ironic class $P(1)$. The two systems are biased towards the ironic class, this is the same behavior observed on the development set. Nevertheless, the results obtained by the two systems are very competitive, obtaining the second and third position of the ranking.

Table 3.17 Results on the SemEval test set ranked in terms of $F_1(1)$. Our systems are marked with †.

System	Acc	$P(1)$	$R(1)$	$F_1(1)$
THU_NGN	**73.50**	63.00	80.10	**70.50**
TE-NoPos †	66.96	54.83	**94.86**	69.49
Att-LSTM †	68.75	57.17	84.56	68.22
NTUA-SLP	73.20	**65.40**	69.10	67.20
WLV	64.30	53.20	83.60	65.00
NLPRL	66.10	55.10	78.80	64.80
NIHRIO	70.20	60.90	69.10	64.80

The performance of our systems is similar to the best ranked system in terms of $F_1(1)$. However, the difference in terms of Accuracy shows that the THU_NGN system is better

detecting the non-ironic class. A deeper study is required to analyze the bias towards the ironic class in our systems compared to the THU_NGN system. Several factors, such as the weighting strategy for the cost-sensitive learning, and the use of a multi-task setup for learning the models, could influence this bias.

3.3.4 Analysis

In this subsection, several analyses are presented, with the aim of explaining how the TE-NoPos system is able to tackle with the irony detection problem. With this study, we pretend to analyze some useful features, captured by our model, for detecting the ironic class e.g., word polarities, relationships among words and relevant individual words. First, we intended to detect which attention heads of our system are more related with the detection of the ironic class. Considering these heads, we studied, for ironic samples, the polarity and relevance of individual words as long as the relationships among words. To carry out these analyses, we used the combination of the training and development sets, with the aim of having a higher number of samples for obtaining more robust conclusions.

First, in order to detect the attention heads that play a highly relevant role in the detection of irony, we performed an ablation process of the attention heads of the trained system TE-NoPos. The main purpose addressed in this section is to answer the following question: are there attention heads specialized on detecting the irony? It is reasonable to think that the competitive results obtained by our system for both languages are due to its ability to capture relevant patterns related with the irony. Therefore, we hypothesize that there are some attention heads that react more to word relationships related to irony.

An iterative ablation process was performed to detect the attention heads whose influence in predicting the ironic class is greater. Concretely, this process consists of iteratively deactivating the output of some attention heads. To do this, the output of each head i we want to ablate is masked, and its output is propagated to the next layers of the network as a zero matrix, $head_i = \mathbf{0}^{T \times d_k}$. Then, we can observe the influence that head i have on the results obtained by the system in terms of the F_1 measure of the ironic class ($F_1(1)$).

During the ablation process, all the $2^h - 2$ combinations of h heads taken from 1 at a time, to $h-1$ at a time, are iteratively evaluated with the aim of observing the worsening of the F_1 for the ironic class. In our study, only the combinations that worsen the previous worst result were taken into account. We hypothesize that the heads that have appeared in more combinations during the successive worsening are those most related with the detection of the irony.

After finishing the iterative process, the heads that most reacted to irony were detected. In Table 3.18, the number of times that each attention head belongs to a combination that

worsens the previous worst results are shown, both for the IroSVA and for the SemEval corpora.

Table 3.18 Number of times that each attention head appears in a combination that worsens the results, in terms of $F_1(1)$, after a previous worsening of the results. The total number of worsening during the process is also shown together with the number of occurrences of each head.

Corpus	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
IroSVA	**16/18**	**11/18**	**13/18**	**10/18**	8/18	9/18	4/18	5/18
SemEval	**8/11**	0/11	**10/11**	**6/11**	2/11	3/11	3/11	**8/11**

It is clearly observable that in the English corpus the number of occurrences of the attention heads can be divided in two balanced clusters: those heads that appear in the process more than the half of times and those that appear less than the half. However, in the Spanish corpus there are some attention heads (H_4 and H_5) that are near o exactly on the half number of worsening, i.e. for the Spanish corpus the detection of irony is more scattered among all the attention heads. We considered that, the attention heads that occur more than the half of the times are highly related with the detection of the ironic class. These specialized heads were included in the set H_{ironic}. The remaining heads, less related with the ironic class, were included in the set $H_{non-ironic}$. Thus, in both corpora, there are 4 attention heads that appear more than the half of times ($H_{ironic} = \{H_0, H_1, H_2, H_3\}$ for IroSVA, and $H_{ironic} = \{H_0, H_2, H_3, H_7\}$ for SemEval) and 4 attention heads that appear less than half of the times.

Once the attention heads related to the ironic class detection are identified, it is possible to ablate them in order to observe the results of the system in terms of $F_1(0)$ and $F_1(1)$ without considering them. Table 3.19 shows the results of the TE-NoPos system when no heads are masked (None column in Table 3.19), when only H_{ironic} are masked and when only $H_{non-ironic}$ are masked, for both tasks. It can be seen that in both corpora the results in terms of $F_1(1)$ highly decrease when H_{ironic} is masked.

For the English corpus, masking $H_{non-ironic}$ almost does not affect the $F_1(1)$ results, indicating that those attention heads are not highly related with capturing the ironic class.

Table 3.19 Results on training+development set when masking is not applied, masking H_{ironic} and masking $H_{non-ironic}$.

Corpora	None		H_{ironic}		$H_{non-ironic}$	
	$F_1(0)$	$F_1(1)$	$F_1(0)$	$F_1(1)$	$F_1(0)$	$F_1(1)$
IroSVA	**91.98**	**85.48**	74.11	67.79	85.09	73.03
SemEval	61.88	**71.62**	**69.53**	56.56	41.08	70.17

In addition, masking $H_{non-ironic}$ leads also to a high worsening of $F_1(0)$, suggesting that these heads are related to the detection of the non-ironic class. Therefore, it seems that there are attention heads specialized in detecting the ironic class, those in H_{ironic}, and others specialized in detecting the non-ironic class, those in $H_{non-ironic}$. For the Spanish corpus, masking H_{ironic} or $H_{non-ironic}$ leads to a high decrease of the performance over all the classes. This suggests that the detection of the ironic and non-ironic classes is highly scattered among all the attention heads, as stated before when we discussed the creation of the H_{ironic} and $H_{non-ironic}$ sets. The worsening of the results of the ironic class when certain heads are masked seem to support our hypothesis, stated at the beginning of this section, about the specialization of the attention heads.

After determining the attention heads that play a highly relevant role in the detection of irony, we want to study if the attention heads in H_{ironic} implicitly capture sentiment information. The aim of this study is to determine if this information is useful for detecting the presence of irony. To achieve this goal, we propose a method to compute, for each head k, the average attention that each word w receives from all the other words w' in its context, averaged for all the occurrences of w in the set of samples $\mathscr{D}$. The context of each word w inside a tweet is determined by all the words of the tweet. Algorithm 3 shows this proposal to compute the average attention per word, for each head. From the set of samples $\mathscr{D}$ with vocabulary $\mathscr{V}$ and the trained model f, we compute the average attention given by the head $k \in H_{ironic}$ to each word w in $\mathscr{V}$. To do this, from each sample $X \in \mathscr{D}$ and each head k, the matrix $B \in \mathbb{R}^{|X|\times|X|}$ is computed. The matrix B is the output of the softmax function applied on the scaled dot-product between Q and K matrices, as shown in Eq. 3.20. The rows of B are averaged to obtain $B' \in \mathbb{R}^{|X|}$. This vector B' contains the attention that head k gives to each word in X, computed as the average of the self-attentions in the head. Finally, the attention of each word in each head, α_{wk}, is normalized by dividing it by the number of times that the word w appears in all the samples, c_w.

Once the matrix α is computed, it can be determined what are the most attended words by the heads in H_{ironic}. If the attention heads in H_{ironic} focus on more polarity words than the heads of $H_{non-ironic}$, then the polarity words should be more useful for detecting the ironic class than the non-ironic class. Furthermore, the more polarity words focused by H_{ironic}, the more discriminant they should be for detecting the ironic class. To do this study, we determine the most attended words w for each head k by using a threshold ε, i.e. a word w is highly attended by an attention head k if $\alpha_{wk} > \varepsilon$. We used an $\varepsilon = 0.45$ to take into account a considerable number of highly attended words to do the analysis. To determine the polarity of the most attended words, some polarity lexicons were used. For the English language, we used NRC [188], MPQA [185], AFINN [184], and BingLiu [108]. For the Spanish language,

Algorithm 3 Compute the average word attention, for each head, captured by the model on a set of samples.

Input: $\mathscr{V}$ vocabulary, set of samples $\mathscr{D}$, trained Transformer Encoder f
Result: α_{wk} the average attention of head k for word w

procedure COMPUTEWORDATTENTIONS($\mathscr{D}, f$)
  $\alpha_{wk} \leftarrow 0,\ \forall w \in \mathscr{V} \wedge \forall k \in H_{ironic}$
  for $X \in \mathscr{D}$ **do**
    for $k \in H_{ironic}$ **do**
      $B \leftarrow softmax(\frac{f(X)_{Q_k} f(X)_{K_k}^{\top}}{\sqrt{d_k}})$
      $B' \leftarrow \frac{1}{|X|} \sum_{i=1}^{|X|} B_{ij}$
      $\alpha_{wk} \leftarrow \alpha_{wk} + B'_w,\ \forall w \in X$
    end for
  end for
  $c_w \leftarrow 0,\ \forall w \in \mathscr{V}$
  $c_w \leftarrow c_w + 1,\ \forall w \in X \wedge \forall X \in \mathscr{D}$
  $\alpha_{wk} \leftarrow \frac{\alpha_{wk}}{c_w},\ \forall w \in \mathscr{V} \wedge \forall k \in H_{ironic}$
end procedure

we used ElHPolar [119], iSOL [120] and NRC translated to Spanish. Tables 3.20 and 3.21 show the 5 most attended polarity words for each attention head in H_{ironic} and in $H_{non-ironic}$ respectively, to illustrate the vocabulary considered in the lexicons and the attention that these words receive. Also, with the aim of showing that the attention heads attend mostly to content words, we include the Tables 3.22 and 3.23, that show the 5 most attended words for each attention head, regardless of whether they convey polarity or not. Furthermore, Tables 3.24 and 3.25 show the most attended words by each head as well as which of these words are positive or negative for the Spanish and English corpora respectively.

Table 3.20 Top-5 most attended polarity words by the H_{ironic} heads both for Spanish and English languages.

Language	Heads	(w, α_{wk})
Spanish	H_0	(incompetente, 0.93), (soberbia, 0.90), (desleal, 0.88), (vomitivo, 0.87), (indignacion, 0.86)
	H_1	(laberinto, 0.88), (recomendable, 0.85), (defensor, 0.84), (conspiraciones, 0.81), (maestro, 0.79)
	H_2	(absurda, 0.79), (desinformacion, 0.77), (cobardia, 0.75), (ambicion, 0.67), (mentirosa, 0.64)
	H_3	(salvajismo, 1.0), (corruptos, 0.99), (indecencia, 0.99), (brutal, 0.99), (vomitivo, 0.99)
English	H_0	(persuasive, 1.0), (universal, 0.99), (socialist, 0.99), (supremacy, 0.99), (loon, 0.99)
	H_2	(exhausted, 1.0), (stupidest, 0.99), (president, 0.99), (heck, 0.99), (sensitive, 0.99)
	H_3	(heck, 1.0), (desperately, 1.0), (humid, 1.0), (permission, 1.0), (fault, 1.0)
	H_7	(inspiring, 1.0), (ouch, 1.0), (manic, 1.0), (eventful, 1.0), (sweets, 0.99)

Table 3.21 Top-5 most attended polarity words by the $H_{non-ironic}$ heads both for Spanish and English languages.

Language	Heads	(w, α_{wk})
Spanish	H_4	(unanimidad, 0.87), (cascada, 0.84), (incidencia, 0.81), (colapsado, 0.79), (letrinas, 0.76)
	H_5	(falla, 0.82), (quejica, 0.62), (mola, 0.62), (retenciones, 0.58), (biblioteca, 0.54)
	H_6	(autorizado, 0.85), (burro, 0.81), (mola, 0.81), (pirata, 0.81), (matao, 0.80)
	H_7	(droga, 0.93), (deficiencias, 0.88), (formula, 0.86), (ortiga, 0.79), (transparentes, 0.71)
English	H_1	(inspiring, 0.99), (rewarding, 0.92), (balance, 0.88), (ple', 0.87), (surprising, 0.85)
	H_4	(forgive, 1.0), (quit, 1.0), (obsessed, 1.0), (pleased, 1.0), (oversized, 1.0)
	H_5	(promises, 1.0), (persuasive, 1.0), (practical, 1.0), (subtle, 0.99), (noted, 0.99)
	H_6	(smear, 1.0), (credible, 1.0), (frantic, 1.0), (unwanted, 1.0), (fainting, 1.0)

Table 3.22 Top-5 most attended words by the H_{ironic} heads both for Spanish and English languages.

Language	Heads	(w, α_{wk})
Spanish	H_0	(cocinillas, 0.95), (incompetente, 0.93), (iluso, 0.92), (soberbia, 0.9), (desleal, 0.89)
	H_1	(punset, 0.96), (reducto, 0.93), (novelon, 0.92), (paleolitico, 0.92), (emperador, 0.92)
	H_2	(absurda, 0.79), (mediatica, 0.78), (desinformacion, 0.77), (cobardia, 0.75), (ambito, 0.73)
	H_3	(salvajismo, 1.0), (corruptos, 0.99), (indecencia, 0.99), (vascos, 0.99), (brutal, 0.99)
English	H_0	(substitute, 1.0), (hollywood, 1.0), (persuasive, 1.0), (economics, 1.0), (civilised, 1.0)
	H_2	(jelouse, 1.0), (exhausted, 1.0), (dentists, 1.0), (stupidest, 1.0), (president, 1.0)
	H_3	(shakespeare, 1.0), (noises, 1.0), (librarian, 1.0), (calculations, 1.0), (timing, 1.0)
	H_7	(ouch, 1.0), (eventful, 1.0), (yaayy, 1.0), (greaat, 1.0), (manic, 1.0)

Table 3.23 Top-5 most attended words by the $H_{non-ironic}$ heads both for Spanish and English languages.

Language	Heads	(w, α_{wk})
Spanish	H_4	(ayudarnos, 0.99), (emprendedores, 0.95), (surtido, 0.92), (cemento, 0.92), (tazas, 0.88)
	H_5	(macario, 0.97), (vomito, 0.93), (leas, 0.89), (falla, 0.82), (habriais, 0.76)
	H_6	(roba, 1.0), (comes, 0.96), (joderse, 0.93), (suena, 0.93), (cuece, 0.92)
	H_7	(pla, 1.0), (hammond, 1.0), (droga, 0.93), (fumar, 0.92), (deficiencias, 0.88)
English	H_1	(dip, 1.0), (blessed, 1.0), (inspiring, 1.0), (sleepy, 0.99), (rted, 0.97)
	H_4	(lov, 1.0), (forgive, 1.0), (obssessed, 1.0), (lovee, 1.0), (unfollowed, 1.0)
	H_5	(promises, 1.0), (shifts, 1.0), (workday, 1.0), (persuasive, 1.0), (hygiene, 1.0)
	H_6	(smear, 1.0), (sinus, 1.0), (credible, 1.0), (shift, 1.0), (frantic, 1.0)

Table 3.24 Number of positive and negative words for each attention head, along with the number of highly attended words and the ratio of polarity words for the Spanish language.

Head Set	Heads	$\lvert\alpha_w > \varepsilon\rvert$	Negative	Positive	Ratio
H_{ironic}	H_0	240	102	24	52.50%
	H_1	221	12	18	13.57%
	H_2	73	22	8	41.09%
	H_3	603	140	47	31.01%
	Σ	1137	276	97	32.80%
$H_{non-ironic}$	H_4	276	14	28	15.21%
	H_5	116	6	9	12.60%
	H_6	281	41	11	18.50%
	H_7	237	14	18	13.50%
	Σ	910	75	66	15.50%

Table 3.25 Number of positive and negative words for each attention head, along with the number of highly attended words and the ratio of polarity words for the English language.

Head Set	Heads	$\lvert\alpha_w > \varepsilon\rvert$	Negative	Positive	Ratio
H_{ironic}	H_0	765	92	139	30.20%
	H_2	261	54	45	37.93%
	H_3	544	111	62	31.80%
	H_7	317	29	123	47.94%
	Σ	1887	286	369	34.71%
$H_{non-ironic}$	H_1	159	8	20	17.61%
	H_4	132	14	37	38.63%
	H_5	623	72	149	35.47%
	H_6	817	180	130	37.94%
	Σ	1731	264	336	34.66%

Regarding the Spanish corpus, no heads are reacting in a higher extent to positive words than to negative ones, suggesting that the irony in IroSVA corpus is made by conveying more negative than positive feelings. Furthermore, the heads in H_{ironic} have the highest ratio of polarity words attended, meaning that many of the words highly attended by these heads convey some kind of polarity. The main differences of the English corpus with respect to the Spanish corpus are that in the English corpus a higher attention is given to positive words and a higher ratio of polarity words are attended by all the attention heads, both from H_{ironic} and $H_{non-ironic}$. Moreover, the attention given by the heads from H_{ironic} to polarity words is also more scattered in the English corpus, although, there are some heads mostly specialized on detecting negative (H_3) and positive (H_7) words. All these results suggest that, in addition to the language, both corpora are quite different because of the type of irony present in them.

Perhaps, the fact that the irony of the IroSVA corpus is contextualized (each sample has a certain context) makes its irony different from that of the SemEval corpus.

In addition to the previous analyses, it is also important to study the role of individual words for the irony, especifically, if there are specific words with a high impact on the model when it decides if a sample is ironic or not. Thus, the objective is to determine which words, if any, are more relevant in the decision of the model without taking into account their relationships with other words. This analysis has been addressed from two different perspectives. On the one hand, from the point of view of the attention mechanisms of the TE-NoPos model, by inspecting the attention matrices. To do this, the matrices B for all the attention heads in H_{ironic} are computed and averaged element-wise to obtain a matrix $\hat{B}$, finally the vector B' is computed as the average of the rows of $\hat{B}$. Therefore, in this case, the vector B' contains the averaged attention that each word w receives from all the other words in a tweet, averaged for all the attention heads in H_{ironic}. On the other hand, from the perspective of the gradients of the loss function $\mathscr{L}$ with respect to the input X, $\nabla_X \mathscr{L}(f(X;\theta),y) \in \mathbb{R}^{T\times d}$. This concept is extensively used in the field of explainable AI [230, 231] and for generating adversarial examples [232]. We have used this information to determine the relevance of the words in the decision of our model when ironic samples are correctly classified as ironic. Thus, from a correctly classified ironic sample $X : y = 1 \wedge f(X) = y$ it is possible to compute $\nabla_X \mathscr{L}(f(X;\theta),y = 1)$ to observe what words of X have gradients with higher Euclidean norm. Figure 3.13 shows some examples of ironic tweets. For each example we show, at word level, the Euclidean norm of the gradients ($\nabla_X \mathscr{L}(f(.))$) and the averaged attention vector (B'). The Spanish examples translated to English are: "si la tierra fuera plana se habría caído con el lado de la mantequilla hacia abajo" → "if the earth was flat it would has fallen with the butter side down" and "el libro de pedro @user me parece la inocentada de este año en version anticipada [clap emoji]" → "pedro's @user book seems to me to be an April Fool's joke of this year in advance [clap emoji]".

The examples shown in Figure 3.13 illustrate how the most relevant words are identified in a similar way with both techniques. It is possible to see that, in spite of relationships among words are not considered, the most relevant words seem to be part of dependencies that involve irony e.g., "shopping | sleep | fun" or "oh | look | another | storm" for the English examples and "tierra (earth) | plana (flat) | lado (side) | mantequilla (butter)" or "libro (book) | inocentada (April Fool's joke)" for the Spanish examples. This last example also illustrates the fact that, mainly in the Spanish corpus, there are some words which bias the decisions of the model to the ironic class. In this case, most of the relevance is assigned to the word "book", hinting the topic about the book of Pedro Sánchez.

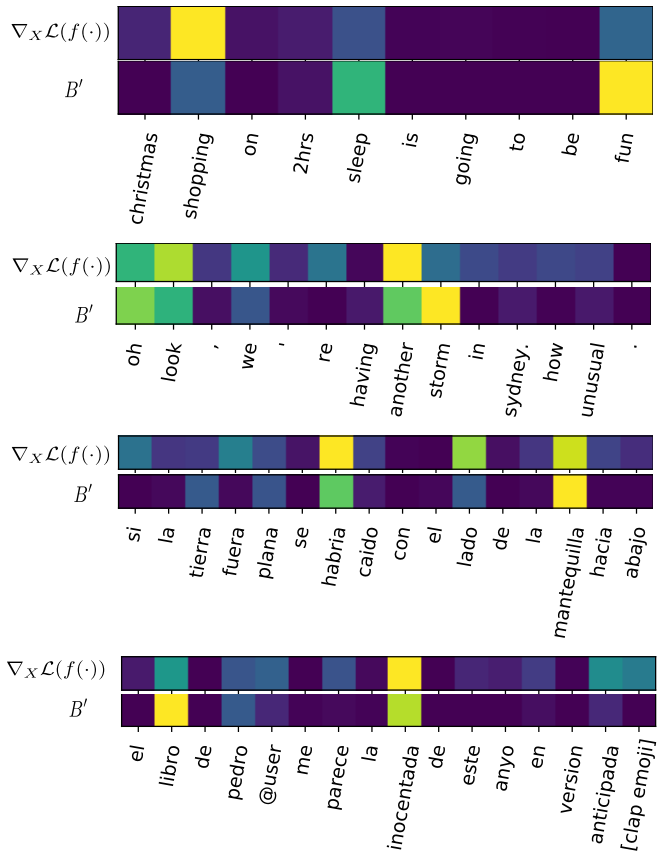

Fig. 3.13 Examples of the word relevance measured by the Euclidean norm of the gradients and the average attentions respectively (the lighter the more relevant)

Finally, we intended to detect specific word relationships that indicates the presence of irony. In some text classification tasks, such as sentiment analysis, some individual words tend to bias the decisions of the models. However, in irony detection tasks, the factor that generally determines the presence of irony is the relationship among words instead of the relevance of individual words. In order to analyze these relationships, the average of the attention matrices of all the heads in H_{ironic} was computed, i.e. $A_{ij} = \frac{1}{|H_{ironic}|} \sum_{k \in H_{ironic}} B_{ij}$, in order to determine the ironic relationships between two words w_i and w_j by observing the attention that the word w_j receives from the word w_i. Thus, the maximum values of the matrix A refer to important ironic relationships between words. Figure 3.14 shows the matrices A for the examples of the previous section, where the first row refers to the English examples and the second refers to the Spanish examples.

In the first English example, it is remarkable the high attention that the word “fun” receives from all the other words, as well as the relationship among the segment “going to

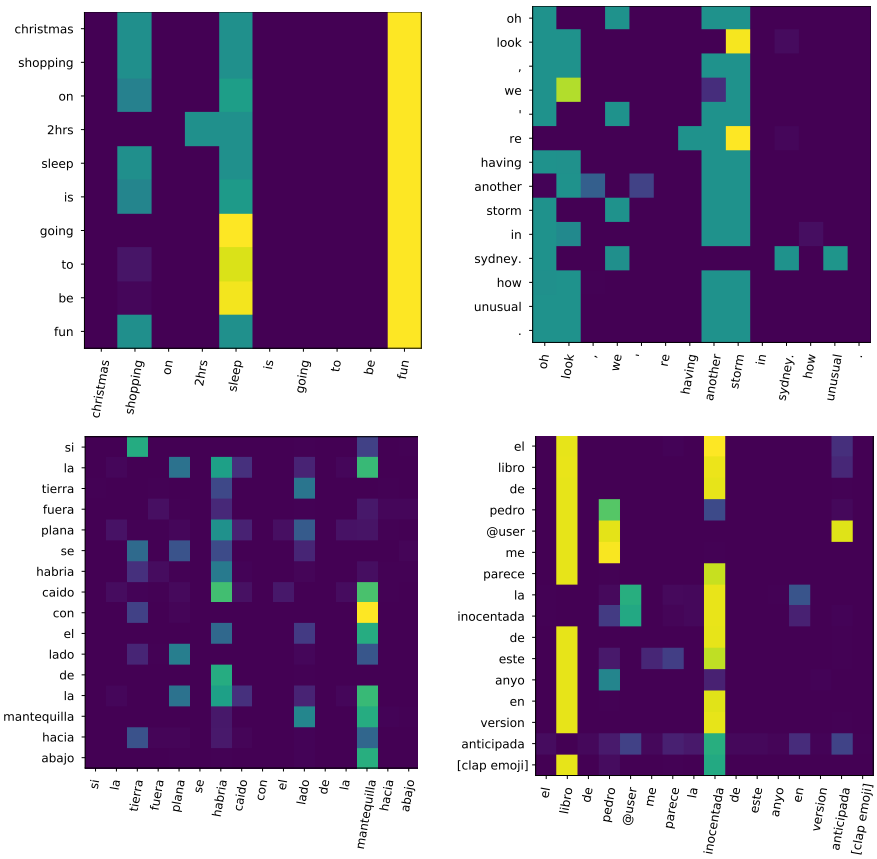

Fig. 3.14 Attention matrices for some ironic examples in both languages (the lighter the more relevant).

be", that precedes the word "fun", and the word "sleep". Furthermore, it is interesting to observe how the words "christmas", "shopping", "sleep", and "fun" attend the same words ("shopping", "sleep", and "fun") with similar attentions. In the second English example, it can be observed how the model relates "look" and "storm" and how the most relevant words "oh", "look", "another" and "storm" are attended highly by all the other words. Regarding the Spanish language, in the first example the attentions are highly scattered, and, the highest attention is given on the word "mantequilla (butter)" by the word "con (with)". Moreover, it is possible to see how the words of the segment "con el lado (with the side)" place their highest attention in the word "mantequilla (butter)". Also, the attention that the words "caído (fallen)" and "plana (flat)" put on the word "habría (would have)" are also high. In the second Spanish example, the two words mostly related with the irony, "libro (book)" and "inocentada (April Fool's joke)", are the most attended by all the other words, highlighting the relationship among the words of the segment "el libro de (the book of)" with "inocentada (April Fool's joke)".

In order to observe in more detail the relationships between words captured by the model, we extracted word pairs with the highest attention from each attention matrix. This pairs excludes those relationships where one of the words is a stopword as well as the relationships where both words are the same. Table 3.26 shows the top-5 highest attended word pairs for the four previous examples. It can be seen that the captured relationships are highly related with the presence of irony e.g., (shopping, fun), (unusual, oh) or (book, April Fool's joke).

Table 3.26 Top-5 relationships between pair of words for the previous ironic examples.

Language	Example	Top-5 Relationships
English	1	(sleep, fun), (christmas, fun) (going, fun), (2hrs, fun), (shopping, fun)
	2	(look, storm), (sydney, unusual), (', oh), (, , storm), (unusual, oh)
Spanish	1	(fallen, butter), (down, butter), (butter, side), (side, flat), (earth, side)
	2	(book, April Fool's joke), (pedro, book), ([clap emoji], book), (year, book), (seems, book)

Part of the research shown in this chapter was published in three papers by the author:

- *José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Self-attention for twitter sentiment analysis in spanish.* Journal of Intelligent & Fuzzy Systems*, 39:2165–2175, 2020*
- *Lluís-F Hurtado, José-Ángel González, and Ferran Pla. Choosing the right loss function for multi-label emotion classification.* Journal of Intelligent & Fuzzy Systems*, 36(5):4697–4708, 2019*
- *José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter.* Information Processing & Management*, 57(4):102262, 2020*

Chapter 4

Pre-trained Deep Bidirectional Transformers for Spanish Twitter

In this chapter we present one of the main contributions of this thesis. This contribution is an adaptation of BERT to the Twitter domain and the Spanish language (TWilBERT), proposed with the aim of boosting the state of the art in text classification tasks with Spanish tweets [30]. Thus, we intend to establish a competitive and easy-to-use baseline that allows the research community to focus on the development of new architectures for text classification tasks with Spanish tweets, potentially built by means of this language/domain specialized representation model.

In recent years, the Natural Language Processing community has been moving from uncontextualized word representations [80, 81, 228] towards contextualized word representations [57, 68, 82–84, 98, 233]. In the first case, each word is represented by one embedding that condenses information of all the contexts where the word appears. While in the second case, each word is represented by different embeddings depending on the context of the word. This allows to model complex features of the words e.g., coreference or polysemy. Among these contextualized architectures, BERT [57] stands out due to its capacity to compute bidirectional contextualized word representations. BERT is a neural bidirectional language model which uses Transformer Encoders [11] as backbone. It is able to compute bidirectional word representations due to the use of a Masked Language Model (MLM) as pretraining objective. MLM is based on Cloze tasks, where tokens are randomly masked, forcing the model to learn the bidirectional context of a token to predict it. Furthermore, the authors of BERT considered sentence coherence as an important aspect of language understanding. For this reason, they proposed the Next Sentence Prediction (NSP) signal with the aim of learning coherence by means of determining if a text segment A precedes a text segment B in the source.

Due to the competitive performance of this model in English downstream tasks, the authors of BERT provided a multilingual version (M-BERT), trained with the Wikipedia dumps of 104 different languages. However, this multilingual model exhibits systematic deficiencies that affect certain language pairs [234]. Furthermore, the competitive performance of BERT in English downstream tasks is not achieved by M-BERT when it is applied to tasks on other languages. or this reason, specializations of BERT for several languages have proliferated [235–239]. In addition to the language, the domain of the downstream tasks is also a key aspect that degrades the performance of this kind of models. The more different the target domain is compared to the source domain, the more remarkable is the degradation. This is especially true for the Twitter domain in which we are interested, where, usually, users communicate with each other informally, and using social network slang.

The competitive behavior of BERT-based models in downstream tasks whose features are similar to the dataset used for pretraining has encouraged the scientific community to use BERT ubiquitously in a broad range of tasks. However, its performance is drastically reduced when this kind of models, especially M-BERT, are used in non-English tasks [236, 237, 239] where some properties like syntaxis and grammar are different from those in which the models were trained. This is the case of the Twitter domain, where M-BERT has to deal with Spanish tweets [225, 240–242]. Typically, these proposals have obtained worse results than other Deep Learning architectures based on the use of incontextual word embeddings trained with Spanish Twitter datasets [16, 24]. Our motivation in this research is to adapt and to improve the language modeling capacity of the BERT architecture to boost the state of the art in text classification tasks in the domain of Twitter for the Spanish language. To achieve this goal, it is necessary to tackle two main challenges.

The first challenge is language dependency. Although the authors of [57] provided multilingual models pretrained with large amounts of texts in many languages (M-BERT), which presupposes that all these languages share structural properties e.g., typological (similar subwords) or grammatical properties. However, even though that M-BERT provides a deeper representation than simply memorizing vocabulary, contextual representations exhibit systematic deficiencies that affect certain language pairs, as shown in [234]. This entails a reduction in the results when fine-tuning is performed on some languages. This is so much so that, in order to obtain more competitive performance, usually, it is better to use simpler models trained in the target language than the M-BERT model.

The second challenge we must tackle is domain dependency. M-BERT was trained using the Wikipedia dataset from 104 different languages. Consequently, the use of M-BERT in other domains can degrade the performance if the target domain is very different from the domain used for pretraining. This is the case of Twitter, where users communicate with

each other informally, using typical expressions of social network slang, many times with lexico-syntactic errors, or adding special tokens such as hashtags, user mentions, and emojis. Therefore, there is a great mismatch between Twitter (target domain) and Wikipedia (source domain).

Another problem related to the domain is the strategy used to learn coherence. We consider that, as discussed in [82], the inter-sentence modeling is an important aspect of language understanding, and we want to take it into account for learning coherence in Twitter. To learn coherence in M-BERT, the self-supervised Next Sentence Prediction (NSP) was used, which allows improving the performance in downstream tasks which require reasoning between pairs of sentences. However, the benefits of the NSP signal have been a controversial topic in the literature [92, 99]. To address the NSP problems, the SOP signal was proposed in [82]. Besides, in the Twitter domain, this signal cannot be used directly, due to there is no sequentiality between sentences like in a document (or tweets in the history of tweets from a given user). Nevertheless, there is a sequentiality between a given tweet and a reply to this tweet in Twitter conversations. For these reasons, in this work, we propose the Reply Order Prediction (ROP) signal, which is an application of SOP to learn coherence between (tweet, reply) pairs in order to improve the performance in downstream tasks that requires reasoning on pairs of tweets. The definition of this signal is identical to SOP, but using positive and negative pairs extracted from Twitter conversations instead of subsequent sentences of a document.

We propose a specialization of BERT both for the Spanish language and the Twitter domain, which we called TWilBERT. This specialization consists of training a BERT model from scratch to obtain coherent contextualized embeddings of Spanish tweets. In order to learn inter-sentence coherence, we propose Reply Order Prediction (ROP), an adaptation of the NSP signal, similar to [82], to Twitter conversations. To our knowledge, this is the first work that proposes a full specialization of BERT for the Twitter domain, taking coherence into account. In addition, we implemented and freely released a Keras [243] framework to train, evaluate and fine-tune TWilBERT models. The pretrained TWilBERT models are released and can be easily used in the provided framework. The main goals of this chapter are: to propose an adaptation of BERT to address text classification tasks in Spanish Twitter; to adapt the Next Sentence Prediction signal for learning coherence between pairs of tweets inside Twitter conversations; to perform an extensive analysis in order to study the performance of TWilBERT, and to provide a framework [1] for training, evaluating, and fine-tuning TWilBERT models, along with some pretrained TWilBERT models.

[1] `https://github.com/jogonba2/TWilBERT`

4.1 Related Work

BERT and several variants of its underlying structure are the state of the art for learning contextual representations that are useful in many Natural Language Processing tasks. In this section, we discuss some of these variants of the BERT architecture and its hiper-parameters, recently published in the literature, that improved BERT in several directions [82, 83, 92].

In SpanBERT [92], several masking strategies were proposed. Their best results were obtained by masking contiguous random token spans instead of single tokens, and using a span boundary objective for predicting each token in a masked span using the tokens on its boundary. The performance of several pretraining masking schemes in span selection tasks such as question answering and coreference resolution, was also studied. They found that using a geometric distribution for sampling random spans provides substantial gains on span selection tasks.

The authors of RoBERTa [83] made a careful measurement of the impact of BERT hyper-parameters and training corpora on the performance of the model. Specifically, they found three interesting aspects that had a great impact on the BERT performance: the NSP signal, the masking strategy, and the batch size. First, the NSP signal consistently degrades the results on downstream tasks, showing that this signal does not provide additional information to the MLM. Regarding the masking, they found that a dynamic masking strategy achieved better results than static masking, i.e. it is better to use different maskings rather than use a small fixed set of masks for each sample during training as in BERT. With respect to the batch size, they found that using large batch sizes improves the perplexity of the MLM objective, as well as the performance on downstream tasks.

In AlBERT [82], the authors found that there is some point where further increasing the model size degrades the behavior of the system in downstream tasks. This degradation was observed empirically when a BERT model with $L = 24$ layers and hidden size $H = 2048$ was trained and fine-tuned for the ReAding Comprehension from Examinations dataset [244], obtaining significantly lower results than another model trained with $L = 24$ and $H = 1024$ (BERT large [57]). To overcome this degradation when the model size increases, while maintaining the training time and the memory consumption, the authors of AlBERT proposed three different strategies. Firstly, the factorized embedding parameterization. This strategy was proposed because of, usually, in this kind of models it is required a higher dimensionality for the contextualized representations than for the subword embeddings. In BERT, increasing the dimensionality of the contextual embeddings forces to increase also the dimensionality of the incontextual subword embeddings due to the residual connections between each pair of subsequent layers. Nevertheless, factorized embedding parameterization allows to decouple the dimensionality of both kinds of embedding, reducing considerably

the number of parameters of the model. Secondly, cross-layer parameter sharing, to improve the parameter efficiency by means of tying the weights among a pre-defined set of layers. The authors of the research shown that this strategy was able to smooth the transitions from layer to layer, thus stabilizing the network parameters. Thirdly, they proposed an alternative to the NSP signal, the so-called Sentence-Order Prediction (SOP). The SOP signal is a reformulation of NSP where pairs of unordered sentences are used as negative samples. The benefits from the NSP signal have been a controversial issue in the literature [92, 99], it seems that the NSP signal captures only topic coherence which does not provide additional information to the MLM task. For this reason, the authors of [82] proposed SOP as pretraining signal to learn better the inter-sentence coherence.

Recently, several strategies for improving Transformer models have been proposed. These strategies can be used to increase the performance of BERT by means of modifying its underlying architecture. Some works in this regard are the LAMB optimizer [56] and Product Key Memory layers [245]. In [56] the authors proposed a layer-wise adaptative large batch optimization technique which allows the models to be trained with very large mini-batches without any degradation of the performance. This way, the training time of the BERT models was reduced from 3 days to 76 minutes in a TPUv3 Pod. In [245], the authors proposed a novel structured memory layer which can be integrated in any neural network with the aim of increasing the capacity of the models without computational overhead. This mechanism has been especially useful in Transformer language models, where a 12-layered Transformer with only one memory layer, under a specific setup of its hyper-parameters, was able to outperform a 24-layered baseline Transformer.

In order to use BERT in other languages different from the English language, the authors of BERT [57] also provided a multilingual pretrained model (M-BERT). However, the competitive performance of BERT in English downstream tasks is not achieved by M-BERT when it is used on other languages. Several works have focused on training specialized BERT models, from scratch or from pretrained weights, for several languages: Dutch [235], French [236, 237], Finnish [238], and Italian [239]. In FlauBERT [236] and in CamemBERT [237], pretrained BERT-based language models were proposed for the French language, which obtained better results than M-BERT, under similar settings, for a wide range of downstream tasks. In [238], a thorough evaluation of M-BERT compared with a BERT model trained from scratch with Finnish texts, was made. The authors shown that the language-specialized version constitutes the state of the art in several Finnish tasks, systematically outperforming M-BERT, which largely fails to reach competitive performance. In AlBERTo [82], a BERT language model was pretrained with Italian tweets (without coherence signal) and evaluated

in several text classification tasks such as irony detection, sentiment analysis, and subjectivity classification.

In addition to the language, another aspect that degrades the performance of pretrained BERT models is the domain. The more different the target domain is from the pretraining domain, the more remarkable is the degradation of the performance. Several works studied this issue, mainly focusing on training BERT models, from-scratch or on from some pretrained weights, in the target domain [239, 246, 247]. In [246], a BERT-based language model was pretrained by using a large-scale dataset of scientific papers. Their experimentation with the domain specialized model shown significant improvements over BERT in a broad set of tasks. In [247], the authors proposed to use combinations of general and biomedical domain corpora in order to train BERT-based language models specialized on addressing named entity recognition, relation extraction, and question answering downstream tasks.

Our TWilBERT proposal leverages recent modifications of the BERT architecture, published in RoBERTa [83] and AlBERT [82], that shown systematic improvements on the MLM objective and downstream tasks. Specifically, our proposal aggregates the inter-sentence coherence loss of AlBERT, applied on (tweet, reply) pairs, along with most of the hyper-parameter choices of RoBERTa that allow for successfully pretraining BERT models such as: dynamic masking, which is crucial for pretraining on large datasets; the use of large batch sizes for improve the perplexity of the MLM objective and the performance in downstream tasks, and the value of the Adam β_2 hyper-parameter for improving stability with large batch sizes.

Beyond the similarities of TWilBERT, AlBERT and RoBERTa in terms of the underlying architecture and its hyper-parameters, the most related work presented in this section is ALBERTo [239], because of we also attempted to address the Twitter domain. In addition to the specialization language (Spanish language in our case), our systems are different from ALBERTo models in a crucial aspect of the BERT architecture. In ALBERTo, the model does not learn coherence among tweets because the cognition of a flow of tweets cannot be automatically identified on a sequence of tweets from the same author. However, we considered that inter-sentence coherence is an important aspect of language understanding that could improve the performance on downstream tasks that require reasoning on pairs of tweets. For this reason, differently from [239], we propose to use coherence signals in Twitter conversations, where a flow of tweets can be easily identified as (tweet, reply) pairs.

In addition, we implemented and freely released a Keras [243] framework to train, evaluate and fine-tune TWilBERT models. All the techniques and improvements discussed in this section are implemented within the framework. Also, the pretrained TWilBERT models are released.

4.2 Proposed Approach

TWilBERT is provided as a framework that allows training, evaluating, and fine-tuning BERT-based models. It also includes several techniques and improvements published in recent works such as: cross-sharing parameter layers [82], factorized embedding parameterization [82], Product Key Memory layers [245], LAMB optimizer [56], gradient accumulation. In addition, we provided two different pretrained models for the Twitter domain in Spanish.

Similarly to what the authors made in [57], two different TWilBERT models were defined, with different number of Transformer layers and attention heads in the multi-head self-attention mechanism of each layer.

On the one hand, TWilBERT-Base (TW-Base) was defined to have half of the Transformer layers and attention heads than M-BERT. TW-Base has $L = 6$ Transformer layers, $A = 6$ attention heads, $d_q = d_k = d_v = 64$ the dimensionality of the Query, Key and Value projections [11], $E = 768$ the dimensionality for the subword embedding layer and $H = E$ hidden size.

On the other hand, TWilBERT-Large (TW-Large) was defined to have the same number of parameters than M-BERT [57]. Specifically, TW-Large have $L = 12$, $A = 12$, $d_q = d_k = d_v = 64$, and $E = H = 768$. We did not use any kind of dropout [49] in the models, due to it can adversely affect the performance of Transformer-based models, as stated in [82]. As pretraining objectives, we used MLM and ROP for both TWilBERT models, in order to learn coherent bidirectional representations of (tweet, reply) pairs.

We used dynamic masking [83] for generating the MLM targets using n-gram masking [92] with a maximum span of $m = 3$ subwords and a maximum of 15% subwords masked for each sample. The probability for masking a span of length $0 < l \leq m$ is defined following Eq 4.1. The probabilities of each kind of token masking ([MASK] token, random subword and keep subword) are the same as in the BERT model [57].

$$p(l) = \frac{1/l}{\sum_{i=0}^{m} 1/i} \tag{4.1}$$

To build the corpus, a total of 91 million of Spanish tweets were streamed from September 2019 to January 2020. We applied a post-process in order to get the replies for all the tweets collected by the streamer. Those tweets that have not got reply, or are not reply of a tweet, or have less than 3 words, were discarded. The result of this post-process was 47 million of (tweet, reply) pairs (7.65Gb of text and 1.16 billion words) which generates 94 million of positive and negative pairs for the ROP signal. All these tweets were segmented as subwords units by using SentencePiece [248] with a vocabulary size of 30,000 subwords. Furthermore, in order to reduce the number of subwords required for representing the (tweet, reply) pairs, user mentions and urls were replaced by a generic token.

We used Adam [55] with gradient accumulation for minimizing the cross-entropy both for the MLM and ROP signals, with an effective batch size of 2048 samples (64 batch size and 32 accumulation iterations). We also used Noam learning rate annealing [11] with 10,000 warmup steps for TW-Large and 8,000 for TW-Base, due to the faster convergence of TW-Base compared to TW-Large. To deal efficiently with pairs of variable-length sequences, we implemented a bucketing strategy based on the lengths of the (tweet, reply) pairs, with a maximum length of 128, in order to reduce, as much as possible, the amount of padding. The buckets were ordered by length in ascending order to be used as input for the TWilBERT models. Two Nvidia Geforce RTX 2080 Ti were used for training TW-Large and TW-Base during 15 days.

Table 4.1 summarizes the differences among both TWilBERT models and M-BERT. Figure 4.1 shows the results, for the MLM and ROP signals respectively at each training step for both TWilBERT models. It can be seen that TW-Large outperforms TW-Base for both the MLM and ROP tasks.

Table 4.1 Differences among M-BERT, TW-Base, and TW-Large.

	M-BERT	TW-Base	TW-Large
Language	104 languages	Spanish	Spanish
Domain	Wikipedia	Twitter	Twitter
Objectives	MLM+NSP	MLM+ROP	MLM+ROP
Tokenization	WordPiece	SentencePiece	SentencePiece
Vocabulary	110k	30k	30k
Masking	Static subword	Dynamic spans	Dynamic spans
L	12	6	12
A	12	6	12
E	768	768	768
H	768	768	768
d_q	64	64	64
d_k	64	64	64
d_v	64	64	64

4.3 Evaluation

In order to evaluate the performance of our proposal, we selected a broad set of text classification tasks for Spanish language on the Twitter domain. Specifically, we are interested in addressing tasks related with social media analysis such as sentiment analysis, emotion detection, stance detection, hate speech detection and topic detection. Furthermore, to make

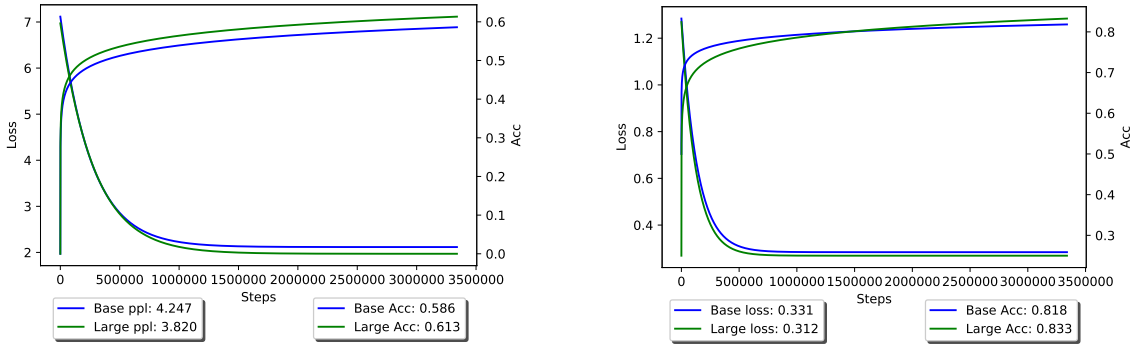

Fig. 4.1 Statistics of MLM (left) and ROP (right) signals. The plots show the evolution in terms of the loss and accuracy during training. The boxes show the final values, after training, of the loss (including perplexity in the case of MLM) and the accuracy.

a fair comparison with the state of the art, we only considered reference corpora provided in three international competitions that are highly relevant in the field: Evaluation of Human Language Technologies for Iberian languages (IberEval), Iberian Languages Evaluation Forum (IberLEF), and International Workshop on Semantic Evaluation (SemEval). Additionally, several tasks that we address in our experimentation provide the corpora in different Spanish variants from Spain, Mexico, Uruguay, etc. We addressed also these cases in order to observe the behavior of our models in specific low-resources variants of the Spanish language. A total number of 14 different corpora have been used for evaluating TWilBERT.

According to the requirements of BERT-based models to operate on a given input and its respective output, all the text classification tasks considered in this section can be divided in the following categories:

- **Single-input single-label**: given as input a sample $X \in \mathscr{V}^T$, a degenerate $(X, \varnothing)$ pair is generated. The pooled token representations are used as input for a softmax output layer that computes a probability distribution over the set of classes $\mathscr{C}$.
- **Single-input multi-label**: in this case, the input is identical to the previous one, however, the last output layer is a sigmoid layer that computes the probability of each class $c \in \mathbb{C}$ as a Bernoulli distribution.
- **Multi-input single-label**: the input is composed by k different text segments. To handle it, all the text segments are concatenated by means of the [SEP] token in order to compose the input for TWilBERT. The output layer is a softmax layer to compute a probability distribution over the set of classes $\mathscr{C}$.

The evaluation metrics used in this experimentation are those considered in the competitions to rank the systems. Specifically, for the single-label tasks, the metrics considered were:

Accuracy (Acc), Macro-Precision (MP, macro-averaged version of Eq. EA.5), Macro-Recall (MR, macro-averaged version of Eq. EA.6), Macro-F_1 (MF_1, Eq. EA.8), and Binary F_1 (Eq. EA.7 when $c = 1$). For the multi-label case, Jaccard Accuracy (JAcc, Eq. EA.1) is considered. All these metrics are discussed in §A.1.

To be able to compare our results with those of the first-ranked system in each task, the training, development, and test partitions provided by the organizers of each task were used. In some tasks the organization did not provide the development partition. In these cases, we have generated them by splitting the train set using a random sampling process. The sampling process selects 20% of the training set as development set, maintaining the original class distribution in both sets. We fine-tuned the models by using a grid search over batch size ([16, 32]), learning rate ([1e-5, 5e-5, 1e-4, 5e-4]), and pooling strategy (averaging the contextualized embeddings or using the embedding of the [CLS] token). Furthermore, weighted cross-entropy was used to tackle with the class imbalance. Each experiment was repeated 3 times, and the best model on the development set was selected to be evaluated on the test set. We perform a comparison among M-BERT, the TWilBERT models and the best system of each competition. Additionally, we consider TW-Large without ROP to observe the behavior of the proposed loss signal, and a BERT model, with the same hyper-parameters than TW-Large, trained only with the Spanish Wikipedia (S-BERT) to observe how much the domain inconsistency between pretraining and fine-tuning affects the performance on downstream tasks. For the sake of simplicity, we added the results of these two systems in all the tables, although their results are discussed in §4.4.

First, for topic classification, we used the dataset of the Classification of Spanish Election Tweets (COSET) task [249]. This task is intended to classify the topic discussed in a tweet into one of five topics related with the Spanish 2015 electoral cycle. The five topics are: Political Issues, Policy Issues, Campaign Issues, Personal Issues, and Other Issues. It is a single-input single-label task, where the MF_1 is used to evaluate and rank the systems. Table 4.2 show the results of M-BERT, TWilBERT models and the best system of the competition.

Table 4.2 Results for COSET task.

System	MP	MR	MF_1	Acc
M-BERT	67.65	64.30	65.25	70.35
S-BERT	63.49	62.07	61.73	64.58
TW-Base	72.03	63.80	**66.58**	71.00
TW-Large	67.84	59.51	61.47	73.20
TW-Large (w/o ROP)	64.77	60.88	62.22	66.18
Best [35]	-	-	64.82	-

The TW-Base system outperforms the best system of the competition by +1.76 MF_1. Also, a large difference of +5.11 MF_1 can be observed between the results of TW-Base and TW-Large. M-BERT is competitive in this task, outperforming also the best system of the competition by +0.43 MF_1. However, TW-Base shows a better behavior than M-BERT, outperforming it by +1.33 MF_1.

For stance detection, we considered two different tasks on the same fact. On the one hand, the Stance Detection in Tweets on Catalan Independence (SDTC) task [250]. This dataset was collected by the organizers during the Catalan elections in September 2015, which have been interpreted by many political actors and citizens as a *de facto* referendum on the independence of Catalonia from Spain. On the other hand, the Multimodal Stance Detection in Tweets on Catalan 1Oct Referendum (MSDTC) task [251]. In this case, the dataset was collected by the organizers during the Catalan Referendum in October 2017. Along with the tweets, a context composed by the previous and the following tweet to each tweet is also provided.

The two competitions were proposed with the aim of detecting the stance of tweets (in favor, against or neutral) towards the target independence of Catalonia in Twitter messages written in Spanish. The SDTC task is a single-input single-label task whereas the MSDTC task can be addressed both as a single-input single-label task (if the context is discarded) or as a multiple-input single-label task. In order to evaluate and rank the systems for the SDTC task, MF_1 discarding the neutral class is used. For MSDTC, the evaluation metric is MF_1 considering the three classes.

Table 4.3 shows the results for the SDTC task. Neither M-BERT nor TWilBERT models outperform the best system of the competition, that is based on a combination of stylistic, structural and contextual features based on n-grams. This can be related with the low performance, observed in this task, obtained by systems based on distributed features in comparison to systems based on categorical features [250]. M-BERT and TW-Base obtained similar results, being both outperformed by TW-Large by +2.01 MF_1 and +1.86 MF_1 respectively.

Table 4.3 Results for SDTC task.

System	MP	MR	MF_1	Acc
M-BERT	57.11	54.61	43.73	69.80
S-BERT	54.07	53.75	43.36	65.77
TW-Base	52.05	55.40	43.88	61.79
TW-Large	54.86	56.27	45.74	66.51
TW-Large (w/o ROP)	52.22	54.10	43.16	64.57
Best [252]	-	-	**48.88**	-

Table 4.4 shows the results for the MSDTC task. In this case, we considered the task both as single-input (*sgl*) and multiple-input (*mpl*). For the *mpl* experiments, the central tweet and the next tweet are joined by means of a [SEP] token to compose the input.

Table 4.4 Results for MSDTC with *sgl* and *mpl* input configurations.

System	MP	MR	MF_1	Acc
M-BERT (*sgl*)	48.71	47.63	47.06	52.53
S-BERT (*sgl*)	50.87	50.06	49.53	55.14
TW-Base (*sgl*)	51.08	50.40	50.18	54.96
TW-Large (*sgl*)	52.62	51.48	51.46	55.23
TW-Large (w/o ROP) (*sgl*)	50.81	48.53	47.39	54.33
M-BERT (*mpl*)	54.12	51.14	50.48	56.68
S-BERT (*mpl*)	54.29	52.84	52.70	57.49
TW-Base (*mpl*)	56.23	52.62	52.10	58.03
TW-Large (*mpl*)	57.48	54.51	**54.53**	59.30
TW-Large (w/o ROP) (*mpl*)	55.72	49.71	48.91	54.06
Best [253]	-	-	28.02	-

In the *sgl* experiments, both TW-Base and TW-Large outperform the M-BERT system by +4.40 MF_1 in the best case. In the *mpl* experiments, the ranking of these systems is the same than for *sgl* experiments, being again the TW-Large the system that obtains the best results, by +4.05 MF_1 in comparison with M-BERT and by +2.43 MF_1 in comparison to TW-Base. It can be observed how the addition of context improves the results of all the systems. This could be favored by the NSP and ROP signals used during the pretraining of the models to learn coherence relationships among pairs of inputs. However, in the case of M-BERT, adding the context do not improve the results of TW-Large even without considering the context. This suggests that the ROP signal is better suited for the Twitter domain than the NSP signal. Both M-BERT and TWilBERT models with *sgl* and *mpl* input configurations, clearly outperform the results of the best system in the competition.

It is interesting to observe that the TW-Large without the ROP signal obtains similar results than M-BERT for the *sgl* experiments, however, when the context is included, M-BERT outperforms it by +1.57 MF_1. This shows that including a coherence signal in the training process, even it is not well suited for the Twitter domain, improves the capability of the models for reasoning with multiple inputs. Additionally, the improvement obtained when the context is considered for TW-Large without ROP is smaller compared with the improvements on the other models (1.52 vs 3.42 MF_1 for M-BERT, 1.52 vs 1.92 MF_1 for TW-

Base and 1.52 vs 3.07 MF_1 for TW-Large). TW-Large without ROP signal is outperformed, in both *sgl* and *mpl* experiments, by both versions of TWilBERT that consider the ROP signal.

We also used the Irony Detection in Spanish Variants (IroSVA) task [146] to evaluate the behavior of our proposal for Irony Detection. The main objective of the IroSVA task is to identify the presence of irony in short messages (tweets and news comments) written in three different Spanish variants from Spain, Mexico, and Cuba. It is a single-input single-label binary classification task where the evaluation metric is the MF_1. Tables 4.5, 4.6, and 4.7 show respectively the results for the Spain, Mexico, and Cuba variants of the IroSVA task.

Table 4.5 Results for the Spain variant of IroSVA task.

System	MP	MR	MF_1	Acc
M-BERT	69.21	69.63	69.40	72.50
S-BERT	69.02	70.88	69.32	71.17
TW-Base	73.17	72.75	**73.00**	76.17
TW-Large	71.89	70.00	70.70	75.00
TW-Large (w/o ROP)	69.43	67.50	68.16	73.00
Best [24]	-	-	71.67	-

Table 4.6 Results for the Mexico variant of IroSVA task.

System	MP	MR	MF_1	Acc
M-BERT	61.92	62.42	62.10	65.66
S-BERT	69.35	67.14	67.86	73.00
TW-Base	68.79	67.91	68.27	72.50
TW-Large	69.30	70.01	**69.61**	72.50
TW-Large (w/o ROP)	64.99	66.06	65.28	68.00
Best [24]	-	-	68.03	-

Table 4.7 Results for the Cuba variant of IroSVA task.

System	MP	MR	MF_1	Acc
M-BERT	69.72	65.87	66.75	73.00
S-BERT	63.94	64.75	64.19	67.17
TW-Base	67.17	67.50	67.32	70.67
TW-Large	70.00	66.88	**67.73**	73.33
TW-Large (w/o ROP)	67.04	66.00	66.40	71.00
Best [254]	-	-	65.96	-

It can be seen that, for all the Spanish variants, the TWilBERT models outperform M-BERT and the best systems of the competition by a margin between +0.98 and +7.51 MF_1. For the Spain variant, TW-Base outperforms TW-Large by +2.3 MF_1, however, for the Mexico and Cuba variants, TW-Large outperforms TW-Base by +1.34 MF_1 and +0.41 MF_1, respectively. Also, these results can be compared with our experimentation in §3.3 (see Table 3.16), where we used non-pretrained. In this case, TWilBERT outperforms our previous proposal in all the Spanish variants by +1.33 in the Spain variant, +1.58 in Mexican and +2.46 in Cuban.

For emotion detection, we used the dataset provided in the SemEval-2018 Task 1: Affect in Tweets task [113]. This task includes an array of subtasks for inferring the affectual state of a person from a given tweet. In our case, we only focused on the subtask E-c (SemEval-Ec). It is a single-input multi-label task with 11 different classes $\mathscr{C} = \{$anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust$\}$ where the evaluation metric is JAcc.

Table 4.8 shows the results for the SemEval-Ec task. M-BERT obtained worse results than the best system of the competition, being outperformed by +2.05 JAcc. TW-Base outperforms M-BERT by +1.26 JAcc, but it showed a lower performance in comparison to the best system. TW-Large is the system that obtained the most competitive results, outperforming the best approach in the competition by +1.70 JAcc. Again, the results can be compared with those shown in §3.2 (see Table 3.10). In this case, TWilBERT outperforms our previous proposal by +1.27 JAcc.

Another task we also considered for evaluating TWilBERT is hate speech detection. We used the dataset provided in the SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter task (HatEval) [256] to evaluate our proposal. Specifically, we focused on the Subtask A, that is a single-input single-label binary classification, where the systems have to predict whether a tweet in Spanish with a given

Table 4.8 Results for SemEval-Ec task.

System	JAcc
M-BERT	44.85
S-BERT	40.40
TW-Base	46.11
TW-Large	**48.60**
TW-Large (w/o ROP)	47.48
Best [255]	46.90

target (women or immigrants) contains hate speech. The evaluation metric used in this task is MF_1.

Table 4.9 shows the results for the HatEval task. It can be seen that, the M-BERT model and the two TWilBERT models outperform the best system of the competition. Specifically, M-BERT showed the best behavior, with an improvement of +2.98 MF_1 compared to the best system. M-BERT also outperformed TW-Large by +2.83 MF_1. The results of TW-Base and M-BERT are similar, being +0.68 MF_1 higher for the M-BERT model.

Table 4.9 Results for HatEval task.

System	MP	MR	MF_1	Acc
M-BERT	75.82	76.25	**75.98**	76.50
S-BERT	71.48	71.23	71.34	72.38
TW-Base	74.68	75.45	75.30	74.44
TW-Large	73.19	73.90	73.15	73.44
TW-Large (w/o ROP)	70.40	70.99	70.39	70.75
Best [257]	-	-	73.00	-

The last task we considered is sentiment analysis. For evaluating the performance in sentiment analysis, we used the datasets provided in the 2019 edition of the Workshop on Semantic Analysis at SEPLN (TASS). It is a single-input single-label task on four classes $\mathscr{C} = \{Negative, Neutral, None, Positive\}$ where the *None* class refers to tweets that do not express sentiment and the *Neutral* class refers to tweets where both *Positive* and *Negative* sentiments are expressed with the same intensity. The organizers of the task provided five different corpora, considering five different variants of the Spanish language from Spain, Mexico, Peru, Costa Rica, and Uruguay. The evaluation metric used for evaluating and ranking the systems is the MF_1.

Tables 4.10, 4.11, 4.12, 4.13, and 4.14 show the results on the Spain, Costa Rica, Uruguay, Peru, and Mexico variants respectively. Except the case of Costa Rica variant, always there is a TWilBERT model that obtains better results than the best system of the competition.

For all the variants, both TWilBERT models outperformed M-BERT, obtaining results up to +11.07 MF1.

Table 4.10 Results for the Spain variant of TASS task.

System	MP	MR	MF_1	Acc
M-BERT	49.17	49.36	48.89	59.38
S-BERT	43.08	41.98	42.82	51.82
TW-Base	51.96	50.75	50.84	59.14
TW-Large	52.10	51.94	**51.64**	59.50
TW-Large (w/o ROP)	50.47	48.00	48.55	55.51
Best [16]	50.50	50.80	50.70	-

Table 4.11 Results for the Costa Rica variant of TASS task.

System	MP	MR	MF_1	Acc
M-BERT	46.85	46.49	46.20	50.52
S-BERT	45.33	43.16	43.37	52.06
TW-Base	49.40	50.51	49.46	57.20
TW-Large	50.05	51.06	50.24	59.52
TW-Large (w/o ROP)	46.30	46.95	46.21	50.85
Best [242]	58.88	45.40	**51.20**	-

Table 4.12 Results for the Uruguay variant of TASS task.

System	MP	MR	MF_1	Acc
M-BERT	46.80	46.01	45.14	56.58
S-BERT	46.95	47.81	46.95	53.29
TW-Base	53.76	56.40	54.56	63.00
TW-Large	55.49	60.12	**56.21**	62.88
TW-Large (w/o ROP)	49.74	48.25	48.35	54.41
Best [16]	49.70	53.60	51.50	-

Table 4.13 Results for the Peru variant of TASS task.

System	MP	MR	MF_1	Acc
M-BERT	46.58	40.60	37.90	37.43
S-BERT	38.75	39.51	38.48	42.14
TW-Base	45.83	45.36	**45.49**	48.22
TW-Large	48.40	46.28	45.01	44.06
TW-Large (w/o ROP)	47.45	41.46	39.30	39.48
Best [258]	46.20	44.60	45.40	-

Table 4.14 Results for the Mexico variant of TASS task.

System	MP	MR	MF_1	Acc
M-BERT	49.78	47.97	46.71	64.80
S-BERT	45.60	46.66	45.57	58.26
TW-Base	47.37	52.13	47.75	62.73
TW-Large	51.39	50.64	**50.38**	63.93
TW-Large (w/o ROP)	47.80	48.57	48.13	63.67
Best [16]	49.00	51.20	50.10	-

These results show the lack of specialization of M-BERT in almost all the Spanish variants as those from Uruguay, Peru, Costa Rica, or Mexico. Besides that, the results of M-BERT in the Spain variant are more competitive than in the other variants. This may be due to the Spanish Wikipedia dataset used for training M-BERT does not include expressions from the Latin American variants. Also, we can compare these results with those obtained in §3.1 (see Table 3.5). Again, for all the Spanish variants, TWilBERT outperforms our previous proposal: improvement of +0.96 M-F_1 in the Spain variant, +0.66 in Costa Rican, +4.67 in Uruguayan, +0.75 in Peruvian and +0.28 in Mexican.

4.4 Analysis

In the previous section, we have studied the behavior of several TWilBERT and BERT models on a set of 14 different text classification datasets. The average results obtained in these 14 datasets are shown in Table 4.15. It can be seen how the TWilBERT models that consider the ROP signal outperform the M-BERT model by +3 points on average. By contrast, if the ROP signal is not used during pretraining, the results are very similar to those obtained by M-BERT. Also, if BERT is pretrained only with the Spanish Wikipedia, the

results obtained are 1 point lower on average than those obtained by M-BERT. These results suggest that the language, the domain and the coherence are relevant for obtaining better results on downstream tasks. According to the results (S-BERT < TW-Large (w/o ROP) < TW-Base < TW-Large) the language seems to be the less relevant aspect, followed by the domain consistency and the coherence. Regarding M-BERT, the multilingual pretraining shows a great capability for generalizing both for the language and the domain, however, the TW-Large (w/o ROP), which is pretrained with substantially less data and does not consider inter-sentence coherence, obtains the same results. As shown in 4.15, is the combination of the specific language, domain and coherence signal that makes the difference.

Table 4.15 Averaged results on all the text classification datasets.

	M-BERT	TW-Base	TW-Large	TW-Large (w/o ROP)	S-BERT
Avg	53.60	56.49	**56.89**	53.57	52.68

We hypothesized that these improvements are possibly due to three main factors, related with the Twitter domain: the performance of the language model on tweets, the coherence between tweets learned by means of ROP (especially useful in multi-input tasks) and a lower redundancy among the patterns captured by the attention heads of TWilBERT in comparison to those of M-BERT. These factors are analyzed below.

First, we study the specialization of the **language models** of M-BERT and TWilBERT to the Twitter domain. To do this, we built a dataset, $\mathscr{D}$, that contains all the tweets of the 14 datasets used in the previous section. This dataset is composed by 86,542 tweets. The aim of this analysis is to compute the probability that each language model assigns to $\mathscr{D}$, because the more probability a model assigns to the elements of $\mathscr{D}$, the more specialized this model is in $\mathscr{D}$. However, it is not easy to compute a probability for a text sequence using BERT-based language models due to they are bidirectional. Nevertheless, as shown in [259], BERT-based models can be interpreted as Markov Random Field language models. This interpretation can be used to compute unnormalized log-probabilities, which allow us to find the model that assigns a higher score to the tweets of $\mathscr{D}$. Eqs. from 4.2 to 4.5 show the process to compute the averaged unnormalized log-probabilities assigned by a model f_θ to the dataset $\mathscr{D}$.

$$\gamma(\mathscr{D}) = \frac{1}{N}\sum_{i=1}^{N}\alpha(X_i) \tag{4.2}$$

$$\alpha(X) = \frac{1}{|X|}\sum_{t=1}^{|X|} log\ \phi_t(X) \tag{4.3}$$

$$\phi_t(X) = f_\theta(X_{\setminus t})_{x_t} \tag{4.4}$$

$$X_{\setminus t} = \{x_1, ..., [\text{MASK}], x_{t+1}, ..., x_{|X|}\} \tag{4.5}$$

where N is the number of samples in $\mathscr{D}$, $\phi_t(X)$ is the probability assigned by the model f_θ to the token t in the sample $X \in \mathscr{D}$, $\alpha(X)$ is the average of unnormalized log-probabilities for all the tokens in X, $\gamma(X)$ is the average of α for all the samples X, and $X_{\setminus t}$ is the tweet X where the token t is masked using the token [MASK], used as input for the model f_θ. The higher the probability assigned to the sample X, the closer to zero is $\alpha(X)$. Therefore, the more fitted a model f_θ is to the dataset $\mathscr{D}$, the closer to zero is $\gamma(\mathscr{D})$. Table 4.16 shows $\gamma(\mathscr{D})$ for M-BERT, S-BERT and the TWilBERT models.

Table 4.16 $\gamma(\mathscr{D})$ results for each model.

	M-BERT	TW-Base	TW-Large	TW-Large (w/o ROP)	S-BERT
$\gamma(\mathscr{D})$	-4.16	-1.26	**-1.19**	-1.21	-3.19

It can be observed a correspondence between the results shown in Tables 4.15 and 4.16 for the M-BERT, TW-Base and TW-Large models, where the ranking among them is the same in both tables. However, in spite of $\gamma(\mathscr{D})$ for TW-Large without ROP is very similar to TW-Large, it obtains similar results to M-BERT. These results suggest that, although the performance of the language model is relevant for improving the performance on downstream tasks, there are other aspects that affect to the performance. A deeper study will be necessary in order to explain these results. The results of TW-Base and TW-Large are also very similar (difference of 0.07 in terms of $\gamma(\mathscr{D})$), being also similar their results averaged for the downstream tasks (difference of 0.40 points in average). It is interesting to see that $\gamma(\mathscr{D})$ is significantly higher for those models trained with tweets, in comparison to those trained with a general domain. This suggests that the domain inconsistency between pretraining and fine-tuning affects negatively to the language modeling task on the downstream domain. TW-Large is the language model which best fits the dataset $\mathscr{D}$.

Second, we analyze the **coherence between tweet pairs** captured by ROP (TWilBERT models) and NSP (M-BERT). To do this, we crawled a new dataset $\mathscr{D}'$ that contains 15,000 (tweet, reply) pairs unseen during the training phase. Following [82], we considered two different levels of coherence: topic prediction and inter-sentence coherence. From $\mathscr{D}'$, we generated two new datasets, $\mathscr{D}'_1, \mathscr{D}'_2$. Both datasets are composed of 15,000 positive pairs and 15,000 negative pairs. The purpose of both datasets is to evaluate M-BERT and the TWilBERT models in two binary classification tasks to classify positive and negative pairs with respect to the aforementioned levels of coherence. The positive instances of $\mathscr{D}'_1$ and $\mathscr{D}'_2$ are the samples of the dataset $\mathscr{D}'$, while the negative samples are built following the

coherence level to study. On the one hand, the negative samples of $\mathscr{D}'_1$ are (tweet, reply) pairs where the reply of a tweet is randomly sampled among the replies of all the other tweets in $\mathscr{D}$. Thus, this coherence level is focused on topic relationships among tweets and their replies [82]. On the other hand, the negative instances of $\mathscr{D}'_2$ are (reply, tweet) shifted pairs, thus breaking sequentiality of the conversations to force models to detect inter-sentence coherence. Table 4.17 shows the Accuracy for M-BERT and for the TWilBERT models in both datasets.

Table 4.17 Accuracy of M-BERT and TWilBERT models for the two levels of coherence.

	$\mathscr{D}'_1$	$\mathscr{D}'_2$	Avg
M-BERT	47.68%	49.41%	48.55%
TW-Base	**55.43%**	86.15%	70.79%
TW-Large	54.57%	**91.27%**	**72.92%**

As it can be seen in Table 4.17, M-BERT behaves like a random system in both datasets, thus showing a lack of specialization in the two levels of coherence when it is applied to the Twitter domain. The TWilBERT models better capture the coherence between pairs of tweets, obtaining statistically significant improvements both for topic prediction ($\mathscr{D}'_1$) and for inter-sentence coherence ($\mathscr{D}'_2$) in comparison to M-BERT. The same behavior was also observed in [82]. It is interesting to highlight that, in spite of M-BERT was trained over pairs of sentences by using the NSP signal [2], this system obtained up to -7.75% of accuracy less than the TWilBERT models on $\mathscr{D}'_1$ dataset. Both TWilBERT models obtained similar results in $\mathscr{D}'_1$, without significant differences. However, TW-Large obtained significant improvements on $\mathscr{D}'_2$ in comparison to TW-Base. TW-large is the system which better captures the coherence, in average, for the two coherence levels.

Finally, we study the **redundancy** of the attention heads of each model. The aim of this study is to determine if some attention heads detect similar patterns than other attention heads in the same model. This way, a low redundant system must be specialized in detecting a wide variety of patterns in each abstraction level, thus improving the performance in downstream tasks [260]. To do this analysis, we computed the Jensen-Shannon Divergence (JSD) among all the pairs of attention heads in all the Transformer layers, in the same way that [261]. We computed the distance between the attention distributions of two heads, H_i and H_j, as shown in Eq. 4.6. We applied multidimensional scaling to project the JSD among the attention

[2]These pairs were built by the authors of [57] in the same way that we built the $\mathscr{D}'_1$ dataset. However, they used sentences from Wikipedia.

heads in two dimensions. This projection can be seen in Figure 4.2 for TW-Large, TW-Base and M-BERT respectively; where L indicates the layer to which each head belongs.

$$J = \sum_{X \in \mathscr{D}'} \sum_{t \in X} \text{JSD}(H_i(t), H_j(t)) \tag{4.6}$$

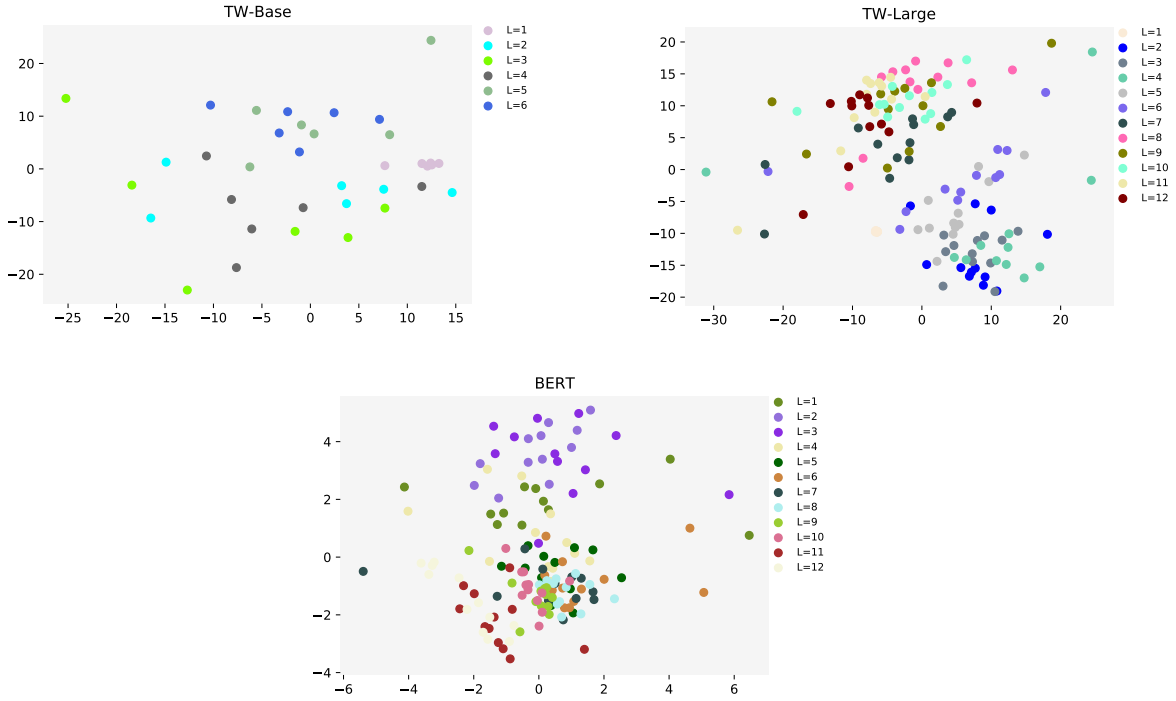

Fig. 4.2 Visualization of JSD divergences among TW-Large, TW-Base and M-BERT attention heads embedded in two dimensions.

As it can be observed, for all the models, there are several clusters of heads that behave similarly. Specifically, attention heads in the same layers tends to get closer, which was also observed by the authors of [261]. Furthermore, they mentioned that “one possibility for the apparent redundancy in BERT’s attention heads is the use of attention dropout, which causes some attention weights to be zeroed-out during training". However, the TWilBERT models, that do not use any kind of dropout, also shown this inner-layer redundancy. The system that shows less redundancy is TW-Base, possibly because the reduced number of attention heads forces a higher specialization of these heads. M-BERT and TW-Large show a similar behavior. However, the JSD among the M-BERT heads is more concentrated in a reduced space ($x_1 \in [-6,6], x_2 \in [-6,6]$) than in TW-Large ($x_1 \in [-20,20], x_2 \in [-30,20]$). It can be also observed that the attention heads are grouped in two different clusters. The first cluster is composed by the first half of the heads ($\text{L} \in \{1,6\}$) and the second cluster is composed by the second half ($\text{L} \in \{7,12\}$). The inter-class and the intra-class distances between the

two clusters are higher in TW-Large than in M-BERT, which suggests that TW-Large is less redundant than M-BERT and, thus, more specialized in computing different patterns at each abstraction level.

From the two aforementioned clusters, we randomly selected three attention heads to observe what patterns they capture. Specifically, we selected heads 2, 4, and 5 (from the first cluster) and heads 9, 10, and 11 (from the second cluster). Figure 4.3 shows the attention weights of these heads for a given sample. The first row refers to the heads 2, 4, and 5, and the second row refers to the heads 9, 10, and 11.

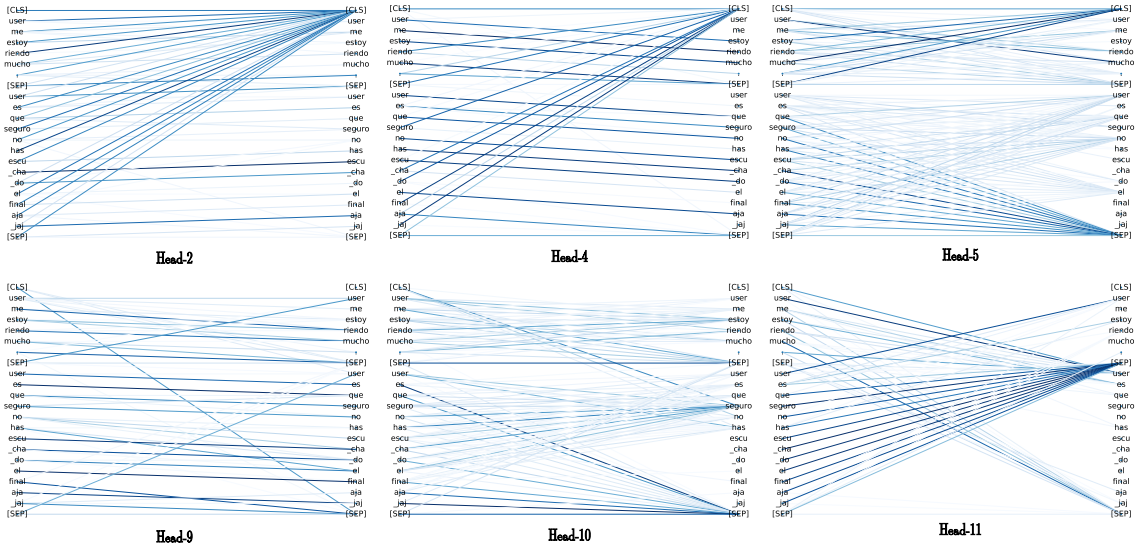

Fig. 4.3 Attention weights for the sample "*[CLS] @user me estoy riendo mucho . [SEP] @user es que seguro no has escu _cha _do el final aja _jaj [SEP]*" (tweet and reply are separated by the intermediate [SEP] token). The English translation of this pair is: "*[CLS] @user I'm laughing a lot . [SEP] @user is that surely you have not heard the end ahaha [SEP]*"

Several surface-level patterns can be observed in Figure 4.3. Heads 2, 9, and 4, attend to the previous, next, and 2 position next token, respectively. Heads 5 and 10 are focused on the separation between the tweet and its reply. This separation is clearer in head 5, where the tweet attends to the [CLS] token and the reply attends to the last [SEP] token. In head 10, the attentions are more scattered than in the head 5. In general, it is also observed a large amount of attention to the tokens [CLS] and [SEP], especially in the head 11, where all the tokens attend to the intermediate [SEP] token.

Part of the research shown in this chapter was published in one paper by the author:

- *José Ángel González, Lluís-F. Hurtado, and Ferran Pla. TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter.* Neurocomputing, *426:58 – 69, 2021*

Chapter 5

Automatic Summarization

Nowadays, the need for automatic summarization systems is directly proportional to the amount and the complexity of the unstructured information published in digital media such as news articles, blogs, discussions in social media, e-books, etc. This unstructured information grows exponentially, which makes it difficult for the users of such digital media platforms to focus on the most relevant contributions that most concern them. This is an interesting effect, since, although people have access to almost all the information of the world, they have time constraints to extract relevant knowledge from such a vast amount of content. Furthermore, the number of people with Internet access also has experimented an exponential growth, so, presumably, not all of them have the same cognitive capabilities (special needs, intellectual disabilities, etc.) or the same background knowledge. In this way, the automatic summarization problem also covers a social dimension, posing as an effective solution for this type of users to understand the key content of the resources [262]. These resources may be in different formats like video, audio, text, or even multimodal combinations among them, and they may pertain also to different domains e.g., newspapers, scientific articles, medical reports, tv programs, etc.

In any case, the action of summarizing is well defined in the abstract, following the Royal Spanish Academy: "reduce to short and precise terms, or just consider and briefly repeat the essentials of a subject or matter". However, in spite that it is well defined for being understood by humans, it is very frequent that different people have different ideas about what is a valid summary for a given resource. This is especially due to aspects like preferences for specific ideas (lack of subjectivity) [263, 264] or an imprecise understanding of the summarization problem e.g., what is, in the previous definition, the "essentials of a subject or matter"? isn't it different, depending on the goal of the people when processing a resource?.

These issues affect directly to automatic summarization systems, both to supervised and unsupervised systems, either because they depend on supervised (document[1], summary) pairs or because humans have to design the systems in terms of what is for them a valid definition of a summary. Furthermore, they are especially harmful to evaluating the systems, since, the definition of a valid summary for a document is subjective. Although it is clear that the goal is to measure the quality of the summaries, no measure highly correlated with that goal has yet been found in current summary environments.

Historically, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [265] has been used as the *de facto* standard proxy to evaluate automatically the quality of summaries. This metric was introduced in the Document Understanding Conference (DUC) 2004, with the aim of reducing the cost of manual evaluation by means of automatic comparisons in terms of the overlapping among token subsequences. A more formal introduction of this metric is shown in §5.2.3. In spite of a high correlation was originally found between ROUGE and human judgements, this metric has received criticism for two main reasons. First, ROUGE is less correlated with human judgements than was originally claimed [266–268], due to summarization environments (in terms of data and models) have changed substantially with respect to the first environments. The second reason is implicitly related to the first one: ROUGE is based only on form overlapping, so, it could assign higher scores to summaries with similar forms, independently if they convey the same meaning. This effect can be easily detected by human judges, which causes the main discrepancy with the ROUGE metric. Figure 5.1, shows a document[2], its reference summary, and three summary candidates. It can be seen how a completely invalid summary (factually inconsistent) is higher scored than the other two better summaries. The factually consistent summary extends the reference, specifying "disease" and "antiviral" (two true facts), while our summary abbreviates "United States", adds information about the "hospitalization", and changes the syntactic structure of the reference summary.

Several protocols for manual evaluation have been proposed, mainly based on dimensions related to the quality of the summaries such as relevance, consistency, fluency, coherence, or readability [268, 269]. However, it is prohibitive to manually estimate the quality of summaries generated by automatic systems. A very promising strategy to solve this was recently published [270], and it is based on learning a quality function from human comparisons between summaries. Furthermore, [270] shows that it is possible to integrate the learned quality function to build a human-in-the-loop approach, by using the quality function as

[1]For the sake of clarity, we use the term *document* to refer any kind of resource.

[2]Extracted from `https://www.eldiario.es/internacional/elecciones-eeuu-2020/trump-permanecera-hospitalizado-durante-dias-dar-positivo-coronavirus_1_6265556.html` and translated to English.

Document: Donald Trump will remain hospitalized "for the next few days" after his transfer by helicopter to the Walter Reed military medical center, hours after being diagnosed with COVID-19, as reported by the White House. This Friday night, the US president began therapy with the antiviral Remdesivir. [...]

Reference summary: The President of the United States has begun Remdesivir treatment for COVID-19.

Factually inconsistent summary *(ROUGE-1: 91.66, ROUGE-2: 81.66)*: The President of the United States has begun bleach treatment for COVID-19.

Factually consistent summary *(ROUGE-1: 78.57, ROUGE-2: 53.84)*: The President of the United States has begun treatment with the antiviral Remdesivir for Coronavirus disease.

Our summary *(ROUGE-1: 15.38, ROUGE-2: 0.00)*: The US President, who has been hospitalized with COVID-19, is being treated with Remdesivir.

Fig. 5.1 ROUGE metric scoring higher a factually inconsistent summary than other two valid summaries. The generation of summaries unfaithful to the documents is a frequent and known issue of supervised models trained on likelihood training and approximate decoding objectives [1] (and most of the current systems are based on this).

a reward to finetune a summarization policy. In parallel to the evaluation approaches that require reference summaries, other automatic evaluation approaches have been proposed to dispense with references [271–273], however, these approaches have fallen into disuse due to the widespread success of supervised models.

In practice, all the summarization systems can be classified as extractive, abstractive or hybrid systems[3]. On the one hand, the extractive systems copy spans of text from documents, typically sentences or word n-grams, in order to build the summaries. These systems have the peculiarity of generating grammatically correct, fluent, and readable summaries, as long as the source document is also correct, and the length of the summary is sufficient. On the other hand, abstractive systems are commonly known in the literature as those that have the capability of rewriting source documents, and consider novel phrases not present in them [268, 274, 275]. It should be noted that rewriting and considering novel phrases are only proxies to the abstractiveness, in the sense of *abstraction* as a semantic generalization that extracts general concepts from more specific concepts [276]. In spite of this, the term *abstractive* is used to refer to those systems based on rewriting source documents, even

[3]There are other taxonomies, depending on the summarization problem addressed e.g., single-document summarization or multi-document summarization. In this thesis, we have only focused on single-document summarization.

if they do not explicitly consider abstraction properties. Finally, hybrid models combine extractive and abstractive strategies. They are typically based on first extracting a set of relevant sentences and later adapting them to the reference summaries e.g., compressing or paraphrasing performed in a decoupled way or simultaneously during the training of the models [277–279]. A more detailed description of systems that fall under these categories can be seen in §5.2.1.

The progress in summarization research has been influenced by the organization of evaluation conferences and the collection of corpora for training and evaluation purposes. It can be highlighted DUC[4] which was integrated later in the Text Analysis Conference (TAC)[5]. These conferences were mainly oriented to evaluation tasks, therefore they provide corpora that were not large enough to be used in the estimation of some corpus-based models. This is especially harmful in the case of deep learning models, that are based on supervised learning techniques. Initial works on automatic summarization were based on unsupervised learning approaches by considering statistical word features [280], topic modeling such as Latent Semantic Analysis [281], graph-based approaches such as LexRank [282] and TextRank [283], among others [284] [285]. There are also systems based on supervised learning techniques such as Conditional Random Fields [286] and Support Vector Machines [287].

Modern supervised approaches to automatic summarization take advantage of the success of neural network architectures and their ability to learn continuous features, without the use of hand-crafted features, by means of adjusting millions of parameters [26, 28, 95, 96, 269, 288–297]. Unfortunately, the construction of a high-quality corpus written by humans for all the possible domains of application is not an easy task, even worse if they have to generate thousands of manual summaries for training supervised systems. Fortunately, there are some strategies to reduce the efforts required to build summarization corpora (at least for the research community). On the one hand, to leverage the large amount of information available on the web. Nowadays, for many summarization tasks, the construction of corpora to train supervised systems is done automatically by extracting (document, summary) pairs e.g., by means of metadata like summary bullets ([274, 275, 289, 298]); document fragments that denote the presence of summaries inside documents, like TLDR [299]; or even reference summaries like abstracts of scientific articles [300, 301]. These automatic approaches have received some criticism, mainly due to the summarization task is underconstrained, and they may contain detrimental noise for the systems [268]. On the other hand, recent models, pretrained with self-supervised objectives and vast amounts of data, require fewer data to

[4] http://www-nlpir.nist.gov/projects/duc/

[5] http://www.nist.gov/tac/

better fit a specific problem [302–304]. To our knowledge, there is not research about the effect of the training size for finetuning this kind of models for automatic summarization. However, we consider that this is important to know what is the correct direction in the corpora design for these new and ubiquitous architectures.

In summary, automatic summarization is a problem difficult to model, subject to human interpretation, and difficult to evaluate objectively, but, if it is still researched and applied correctly, it can be very useful for society. Can you imagine summarizing a book, medical records, legislation, or generating scientific surveys automatically?. Automatic summarization is the door for all of these objectives and we have to find the key.

In this chapter, we discuss the proposals for automatic summarization that have been done in this thesis. Specifically, we proposed a theoretical framework for extractive summarization, based on siamese hierarchical networks with attention mechanisms (§5.1). It allows to developing models that dispense with extractive oracles and Reinforcement Learning (RL) techniques based on ROUGE to fit the task into a sequential binary classification problem. Under this framework, we propose two different models, based on different attentional encoders: Siamese Hierarchical Attention Networks (SHA-NN §5.1.1) and Siamese Hierarchical Transformer Encoders (SHTE §5.1.2). These systems have been successfully applied for summarization of news articles (§5.2) and TV talk shows (§5.3).

5.1 Attentional Extractive Summarization

Typically, extractive summarization has been addressed as a sequential binary sentence classification problem [288, 290, 292–294, 305–307]. However, the available corpora do not provide directly this kind of labeling for training purposes, since in general, corpora only consist in (document, summary) pairs. In order to label the document sentences, previously to the training of the model, the most common strategy consists of using suboptimal ROUGE-based extractive oracles [288, 290, 294]. Recently, RL strategies have been extensively applied [292, 293, 305, 306] in order to dispense with the sentence labeling and optimizing directly the ROUGE evaluation metric.

As pointed out before, approaches that do not rely on RL strategies to optimize directly the ROUGE evaluation metric, are mainly based on the use of suboptimal oracle algorithms due to they require a binary sentence labeling in order to be trained. These approaches typically consist in using oracle systems to label the sentences by following some evaluation measures like ROUGE. In [292], two types of oracles are distinguished: individual oracles, that label each sentence independently (e.g., semantic similarity above a threshold) and collective oracles that consider dependencies among sentences (e.g., greedy algorithms to

search combinations of document sentences that maximize the ROUGE with respect to the reference summary). As stated in [292], the problem of the first type of oracles is that they often generate too many positive labels, causing the model to overfit the data. In the other case, the main problem is related to the underfitting, due to the models trained with cross-entropy loss on collective labels will only maximize probabilities for the sentences in the selected sets. Collective oracles are the most common strategy in the literature [278, 290, 294, 308, 309].

To require a sentence labeling for training the systems has several drawbacks. First, the labeling is suboptimal, leading the model to be trained with non-relevant sentences [305]. Second, this problem becomes more complex for large corpora, where obtaining oracles can be computationally intensive if near-optimal solutions are preferred. Furthermore, the sequential classification, where each sentence is classified taking into account its dependencies with all the other sentences in the document, is a complex problem that can be simplified for summarization purposes. For these reasons, we propose a strategy to avoid the need of a sentence labeling and to simplify the sequential classification process.

Our framework allows the summarization systems to learn by themselves relationships among the sentences of documents and reference summaries, thus allowing the design of end-to-end neural extractive summarization systems. These relationships are learned by attention mechanisms, that are interpreted to extract the most relevant document sentences. In order to learn these relationships, we propose to address the summarization task as a binary classification problem where correct summaries are distinguished from incorrect summaries for documents [6]. Therefore, our proposal dispenses with the sentence labeling, avoiding the large computational cost required to compute near-optimal solutions and allowing to address the problem in a simpler way than RL techniques. Furthermore, this framework generalizes our previous proposals in extractive summarization and hopefully, it will be useful for improve them and to continue the research in this kind of approaches. Specifically, our approach is based on two main assumptions:

1. If y is a correct summary for a document x, then y and x must have similar semantics (or similar representations) whereas if w is an incorrect summary for a document x, then w and x must have less similar semantics than in the previous case (or less similar representations).

2. If we can say if a summary y is correct for a document x and we can look at the relevant sentences in x that lead the system to take that decision, then it is possible to build a summary $\hat{y}$, composed by the relevant sentences in x, that is similar to the reference y.

[6] We consider as incorrect summaries, for a given document, the reference summaries of other documents in the corpora.

The two assumptions allow us to address the extractive summarization as a binary classification problem where correct summaries are distinguished from incorrect summaries for documents. Additionally, if the system is able to distinguish the correct summary for a document, then it can be interpreted to extract the sentences in which it focused.

From these assumptions, it is possible to identify the required mechanisms for designing systems based on the proposed framework. First, it is required to learn representations for documents and summaries that can be used to distinguish if a summary is correct for a given document. Regarding this point, we use hierarchical models in order to compute document-level representations from the sentence-level representations, which are built from the word-level representations. Second, a mechanism to distinguish correct summaries for documents, from the document-level representations, has to be designed. In our framework, this mechanism is based on siamese networks, which use the document-level representations to address the summarization task as a binary classification problem, where a probability distribution of the summary correctness is computed. Finally, it is required an interpretable mechanism to compute relationships among document and summary sentences. In our proposal we focus on the attention mechanisms of the hierarchical models at document level in order to compute the relevance of the document sentences. By this way, it is possible to assign a score to each sentence (based on its relevance when distinguishing correct and incorrect summaries) and rank these scores to extract the k most relevant sentences.

The definition of our framework for extractive summarization is as follows. Let $\mathscr{D} = \{(X_k, X'_k)\}_{k=1}^{M}$ be a corpus of M (document, summary) pairs, where all documents and summaries are defined according to a vocabulary $\mathscr{V}$, let $X_k = \{\{x_i\}_{i=1}^{W}\}_{j=1}^{T}$ be a document composed by T sentences of W words, $X_k \in \mathscr{V}^{T\times W}$, let $X'_k = \{\{x'_i\}_{i=1}^{V}\}_{j=1}^{Q}$ be a summary composed by Q sentences of V words, $X'_k \in \mathscr{V}^{Q\times V}$ and let $f : \mathscr{V}^{T\times W} \times \mathscr{V}^{Q\times V} \rightarrow \mathbb{R}^2$ be a model whose input is a (document, summary) pair and whose output is a probability distribution of the summary correctness over $\mathbb{C} = \{0,1\}$, where 0 is for incorrect summaries and 1 is for correct summaries.

The objective is that the model $f(\cdot,\cdot;\Theta)$ has to be able to determine if a (X, X') pair is correct or incorrect. This way, the output of the model for the (X_k, X'_k) pair will be $f(X_k, X'_k) = 1$ (X'_k is the reference summary for the document X) and for the $(X_k, X'_{j\neq k})$ pair, $f(X_k, X'_{j\neq k}) = 0$ ($X'_{j\neq k}$ is the reference summary for another document from the corpus $\mathscr{D}$, different from X). In order to do that, the model must represent documents and summaries in a proper way to distinguish each case. Thus, $f(\cdot,\cdot;\Theta)$ relies on a document encoder $g : \mathscr{V}^{T\times W} \rightarrow \mathbb{R}^{d_g}$ and in a summary encoder $g' : \mathscr{V}^{Q\times V} \rightarrow \mathbb{R}^{d_{g'}}$. A scheme of this framework can be seen in Fig. 5.2.

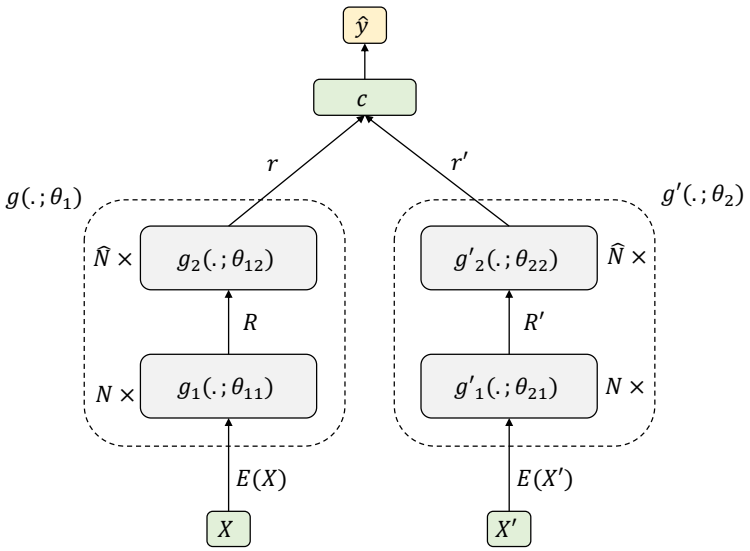

Fig. 5.2 General scheme of Attentional Extractive Summarization framework.

These encoders have to be able to model the hierarchical structure of documents and summaries, so that $g(\cdot;\theta_1)$ and $g'(\cdot;\theta_2)$ are decomposed in two different levels. First, $g_1 : \mathscr{V}^{T\times W} \to \mathbb{R}^{T\times d_{g_1}}$ and $g'_1 : \mathscr{V}^{Q\times V} \to \mathbb{R}^{Q\times d_{g'_1}}$ that are applied independently on each sentence (of documents and summaries respectively) to obtain sentence-level representations from the word-level representations. The encoders can be composed by N hidden layers.

In practice, the words are represented by means of a d_e-dimensional embedding model $\mathrm{E} : \mathscr{V} \to \mathbb{R}^{d_e}$, typically pretrained and applied to arbitrary-length (P) word sequences i.e. $\mathrm{E} : \mathscr{V}^P \to \mathbb{R}^{P\times d_e}$. Therefore, $g_1 : \mathbb{R}^{T\times W\times d_e} \to \mathbb{R}^{T\times d_{g_1}}$ and $g'_1 : \mathbb{R}^{Q\times V\times d_e} \to \mathbb{R}^{Q\times d_{g'_1}}$.

Second, in order to represent documents from the representation of their sentences, $g_2 : \mathbb{R}^{T\times d_{g_1}} \to \mathbb{R}^{d_g}$ and $g'_2 : \mathbb{R}^{T\times d_{g'_1}} \to \mathbb{R}^{d_{g'}}$ are defined. These encoders can have $\hat{N}$ hidden layers.

Therefore, the encoders $g(\cdot;\theta_1)$ and $g'(\cdot;\theta_2)$ are defined as a composition of two levels, $g = g_2(R;\theta_{12})$ and $g' = g'_2(R';\theta_{22})$, where $R = g_1(\cdot;\theta_{11})$ and $R' = g'_1(\cdot;\theta_{21})$. Due to both documents and summaries come from the same domain, they can be represented in the same way through the use of the same set of parameters in both cases, $\theta_{11} = \theta_{21}$ and $\theta_{12} = \theta_{22}$, leading to siamese architectures. The parameters of the documents and summaries encoders are defined as $\theta_1 = [\theta_{11}, \theta_{12}]$ and $\theta_2 = [\theta_{21}, \theta_{22}]$.

As stated before, the document encoder $g(\cdot;\theta_1)$ must be interpretable so that it must assign relevance scores both to words, in order to compute sentence representations, and to sentences, in order to compute document representations. Our approach consists in designing these encoders by means of attention mechanisms that assign scores to words and sentences. Then, document representations are computed as an average of their sentence representations,

using the document level attention mechanism. At the same time, the sentence representations are computed as an average of their words, using the sentence level attention mechanism. The application of these mechanisms is diverse and they can be applied as auxiliary functions on top of the encoders [42, 43] as in [28] or as main mechanisms to compute representations [11] as in [26].

Let $r = g(\cdot;\theta_1)$ and $r' = g'(\cdot;\theta_2)$ be the representations of document and summary respectively, the system must be able to determine if the summary is correct for the document, by using r y r'. In order to do this, a classifier $c(\cdot,\cdot;\theta_3)$ whose output is a probability distribution over $\mathbb{C}$, $c : \mathbb{R}^{d_g} \times \mathbb{R}^{d_{g'}} \rightarrow \mathbb{R}^2$, is applied. Therefore, the model $f(\cdot,\cdot;\Theta)$ can be seen as a classifier $c(\cdot,\cdot;\theta_3)$ applied on top of the encoder outputs, both for document, r, and summary, r', i.e. $f(\cdot,\cdot;\Theta) = c(r,r';\theta_3)$. The parameters of the model are determined by the parameters of each subpart: encoders for documents and summaries and the classifier, $\Theta = [\theta_1,\theta_2,\theta_3]$.

The objective is that the model $f(\cdot,\cdot;\Theta)$ must be able to classify correctly the largest number of pairs, both the positives (extracted directly from the corpora) and the negatives (for a given document, reference summaries from all the other documents in the corpora, sampled by following a distribution p). Therefore, the objective is determined by the minimization of the Eq. 5.1.

$$\mathscr{L}(\Theta) = \sum_{k=1}^{|\mathscr{D}|} \mathrm{L}(f(X_k, X'_k;\Theta), y=1) + \mathop{\mathbb{E}}_{p(X_{j\neq k}|X_k)}[\mathrm{L}(f(X_k, X'_j;\Theta), y=0)] \tag{5.1}$$

where L is a loss function, and $\mathbb{E}_{p(X_{j\neq k}|X_k)}$ denotes expectation with respect to the negative sampling distribution p.

It is interesting to highlight that, once the system is trained for minimizing the training objective, the encoders $g(\cdot;\theta_1)$ and $g'(\cdot;\theta_2)$ must compute proper representations of documents and summaries respectively. By this way, the document representations, computed from their sentences by using the attention mechanism of $g_2(\cdot;\theta_{12})$, are useful to distinguish correct and incorrect (document, summary) pairs. Moreover, this attention mechanism is able to assign a relevance score to each document sentence. Thus, it is possible to determine, focusing on the $g_2(\cdot;\theta_{12})$ attentions, which document sentences have a greater impact on the document representation, being these sentences the most related with the reference summary.

Finally, it is also interesting to highlight that the attention mechanism of $g_1(\cdot;\theta_{11})$ can be used to extract keywords from the documents, being the most attended words inside a sentence those mostly related with respect to the reference summary. We have not experimented in

this work with these attentions, but it opens the door for future improvements by considering the words along with the sentences during the summarization process.

From the definition of the general framework, presented in this section, it is possible to design systems based on it for extractive summarization. To do this, it is necessary to define the encoders both for documents and summaries and both at sentence ($g_1(\cdot;\theta_{11})$ and $g'_1(\cdot;\theta_{21})$) and document level ($g_2(\cdot;\theta_{12})$ and $g'_2(\cdot;\theta_{22})$). Furthermore, it is also required to define a strategy for sentence scoring based on the attention mechanisms of document encoder $g_2(\cdot;\theta_{12})$. In the following subsections, the systems proposed by our research group inside the framework of *Attentional Extractive Summarization* [26, 28] are formalized.

5.1.1 Siamese Hierarchical Attention Networks

Siamese Hierarchical Attention Neural Networks (SHA-NN[7]) [28] is the instance of the general attentional framework when the encoders are Hierarchical Attention Networks [41] based on Bidirectional Long Short-Term Memory (BLSTM) [10] [67] with attention mechanisms, i.e. $g_1(\cdot;\theta_{11}) = \text{BLSTM}_1(\cdot;\theta_1)$, $g'_1(\cdot;\theta_{21}) = \text{BLSTM}_1(\cdot;\theta_1)$, $g_2(\cdot;\theta_{12}) = \text{BLSTM}_2(\cdot;\theta_2)$ and $g'_2(\cdot;\theta_{22}) = \text{BLSTM}_2(\cdot;\theta_2)$. The BLSTM layers are shared for documents and summaries, both at sentence level (BLSTM_1 with dimensionality d_w) and at document level (BLSTM_2 with dimensionality d_s). However, the attention mechanisms for both branches of the siamese model are not shared. Regarding the classifier c, it is a feed-forward network. The architecture can be seen in Figure 5.3.

For this approach, $R \in \mathbb{R}^{T\times d_w}$ and $R' \in \mathbb{R}^{Q\times d_w}$ are computed, following Equations 5.2 and 5.4, as proposed in [42]. They are the output from the sentence level d_w-dimensional BLSTM_1 with attention, where each row i is computed as the average of the hidden vectors of the sentence i attended by $\alpha \in \mathbb{R}^{T\times W}$ (Equation 5.3) and $\beta \in \mathbb{R}^{Q\times V}$ (Equation 5.5) for document and summary respectively. This process, is applied independently to each word embedding matrix that represents each sentence both for document and summary ($R_i : 1 \leq i \leq T$ and $R'_i : 1 \leq i \leq Q$). The following equations show a sentence encoder composed by $N = 1$ BLSTM network.

$$R_i = \sum_{j=1}^{W} \text{BLSTM}_1(\text{E}(X_i))_j \cdot \alpha_{ij} \tag{5.2}$$

$$\alpha_{ij} = \frac{e^{\tanh(W_u \text{BLSTM}_1(\text{E}(X_i))_j + b_u)}}{\sum_{k=1}^{W} e^{\tanh(W_u \text{BLSTM}_1(\text{E}(X_i))_k + b_u)}} \tag{5.3}$$

[7] https://github.com/jogonba2/SHAN

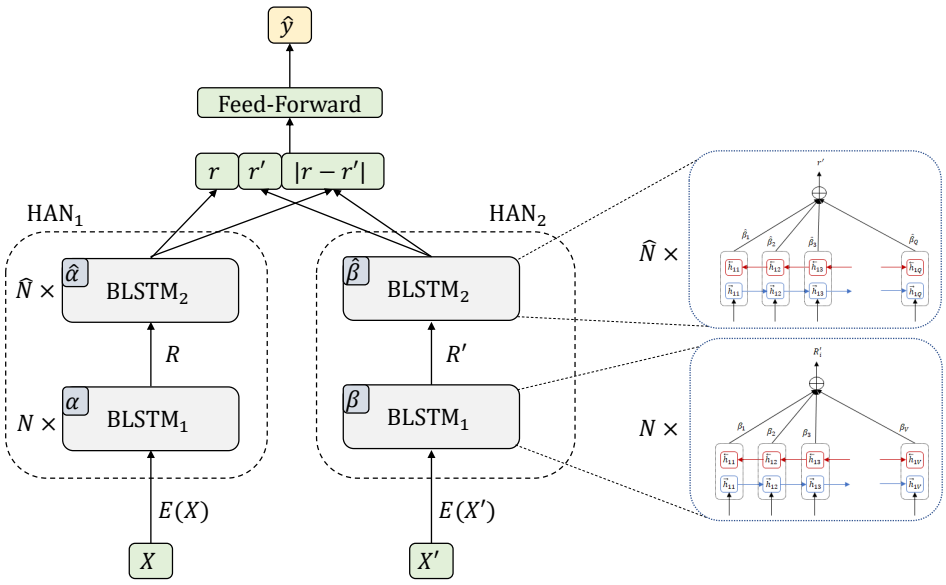

Fig. 5.3 SHA-NN Architecture.

$$R'_i = \sum_{j=1}^{V} \text{BLSTM}_1(\text{E}(X'_i))_j \cdot \beta_{ij} \tag{5.4}$$

$$\beta_{ij} = \frac{e^{\tanh(W_v \text{BLSTM}_1(\text{E}(X'_i))_j + b_v)}}{\sum_{k=1}^{V} e^{\tanh(W_v \text{BLSTM}_1(\text{E}(X'_i))_k + b_v)}} \tag{5.5}$$

where $W_u \in \mathbb{R}^{d_w}$, $W_v \in \mathbb{R}^{d_w}$, are the weights of the attention mechanism for document and summary at word level.

From R and R', $r \in \mathbb{R}^{d_s}$ and $r' \in \mathbb{R}^{d_s}$ can be obtained, following Equations 5.6 and 5.8, similarly to the sentence level but using BLSTM_2 and the attentions $\hat{\alpha} \in \mathbb{R}^T$ and $\hat{\beta} \in \mathbb{R}^Q$ for document and summary respectively. The following equations show a document encoder composed by $\hat{N} = 1$ BLSTM network.

$$r = \sum_{j=1}^{T} \text{BLSTM}_2(R)_j \cdot \hat{\alpha}_j \tag{5.6}$$

$$\hat{\alpha}_j = \frac{e^{\tanh(W_{\hat{u}} \text{BLSTM}_2(R)_j + b_{\hat{u}})}}{\sum_{k=1}^{T} e^{\tanh(W_{\hat{u}} \text{BLSTM}_2(R)_k + b_{\hat{u}})}} \tag{5.7}$$

$$r' = \sum_{j=1}^{Q} \text{BLSTM}_2(R')_j \cdot \hat{\beta}_j \tag{5.8}$$

$$\hat{\beta}_j = \frac{e^{\tanh(W_{\hat{y}}\text{BLSTM}_2(R')_j + b_{\hat{y}})}}{\sum_{k=1}^{Q} e^{\tanh(W_{\hat{y}}\text{BLSTM}_2(R')_k + b_{\hat{y}})}} \tag{5.9}$$

where $W_{\hat{u}} \in \mathbb{R}^{d_s}$, $W_{\hat{u}} \in \mathbb{R}^{d_s}$, are the weights of the attention mechanism for document and summary at document level.

These vector representations r and r', capture bidirectional relationships among the sentence representations, which are obtained from the representations of their words. Then, they can be used to distinguish correct summaries for documents by forcing the attention mechanisms of the document branch to focus on the most relevant sentences. In order to do this, the vector representations of the document r, the summary r', and the difference between them $|r - r'|$ are concatenated and used as input to a feed-forward network with one softmax fully-connected layer, as defined in Equation (5.10), to compute a probability distribution over $\mathbb{C} = \{0, 1\}$.

$$\hat{y} = \text{softmax}(W_{\hat{y}}[r; r'; |r - r'|] + b_{\hat{y}}) \tag{5.10}$$

where $\hat{y}$ is the output of the classifier, $W_{\hat{y}} \in \mathbb{R}^{3d_s \times 2}$ is the weight matrix of the fully connected layer and $b_{\hat{y}} \in \mathbb{R}^2$ is the bias.

Once the network has been trained to distinguish correct summaries for documents, to carry out document summarization with SHA-NN, the attention mechanisms at document level can be directly used to rank sentences and then, to select the most salient ones based on this rank. Specifically, for the summarization process, given a document X, a forward pass is performed on the document branch (left branch) of the siamese network (HAN_1 in Figure 5.3) to obtain the attention score $\hat{\alpha}_j$ of each document sentence. From the ranking of the document sentences based on those scores, the top-k sentences with higher attention score are selected to build the summary.

5.1.2 Siamese Hierarchical Transformers

Due to the process of assigning scores to document sentences of the SHA-NN system is based on the attention mechanisms, then the capacity of these mechanisms plays a crucial role. The greater the capacity of these attention mechanisms to capture complex relationships among different sentences, the better the SHA-NN system will be extracting the most salient sentences to build the summaries. Moreover, the SHA-NN system, as most of the recent extractive systems, rely on recurrent neural networks to derive a semantic representation of the document. These two modifications are the core idea of Siamese Hierarchical Transformer Encoders (SHTE) [26].

Recently, the attention mechanisms have been developed in such a way that they completely replace convolutional and recurrent methods through multi-head self-attention mechanisms, proposed as part of the Transformer models [11]. These multi-head self-attention mechanisms compute word representations by relating different positions of the words in a sentence. Concretely, to compute the representation for a given word, the self-attention compares it to every other word in the sentence. The result of these comparisons is an attention score for every other word in the sentence that determines how much each of the other words should contribute to the representation of the given word, capturing complex relationships between words in sentences such as anaphora, co-reference, coherence and lexical cohesion [310, 311]. Therefore, it seems interesting to incorporate these attention mechanisms in the SHA-NN framework (both at word and sentence level), in order to extract better representations and scores for each sentence in a given document.

Until now, only the ability of transformers to capture word level relationships has been explored. However, these models have not been previously experimented to integrate sentence level relationships in a hierarchical way from the relationships captured at word level. We propose to extend the transformers in a hierarchical way to also work at sentence level. This way, the model could explain relationships among document sentences such as coreference and paraphrasing and use this information inside the extractive summarization framework. So, the contributions are twofold: first, the integration of the transformer encoders in the extractive summarization framework for jointly learning sentence representations and relevant scores, and, second, a hierarchical generalization of the transformer encoders in order to apply them in hierarchical-processing of documents.

SHTE[8] is the instance of the general attentional framework when the encoders, both for sentence and document levels, are Transformer Encoders (TE) [11] shaped in a hierarchical way, i.e. $g_1(\cdot;\theta_{11}) = \text{TE}_1(\cdot;\theta_1)$, $g'_1(\cdot;\theta_{21}) = \text{TE}_1(\cdot;\theta_1)$, $g_2(\cdot;\theta_{12}) = \text{TE}_2(\cdot;\theta_2)$ y $g'_2(\cdot;\theta_{22}) = \text{TE}_2(\cdot;\theta_2)$. Also, in this case, all the weights are shared between the sentence and document levels of the two branches and the classifier c is a feed-forward network. The scheme of this architecture can be seen in Figure 5.4

The multi-head self-attention mechanism used in the Transformer Encoders is defined in Eqs. from 5.11 to 5.13.

$$\text{MultiHead}(A,B,C) = [\text{head}_1;...;\text{head}_h]W^O \tag{5.11}$$

$$\text{head}_i = \text{Attention}(AW_i^Q, BW_i^K, CW_i^V) \tag{5.12}$$

[8] https://github.com/jogonba2/SHTE

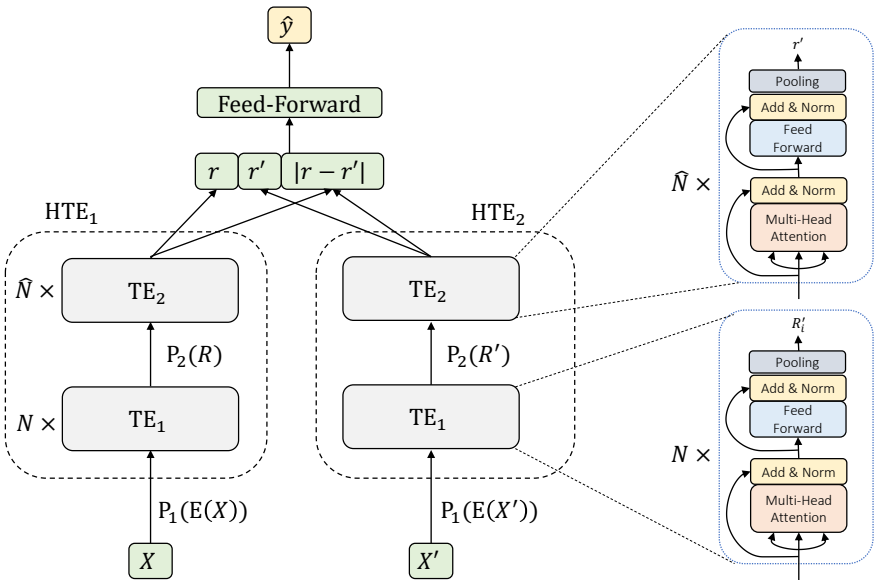

Fig. 5.4 SHTE Architecture.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\intercal}}{\sqrt{d_k}}\right)V \tag{5.13}$$

where A, B and C are the inputs of the multi-head attention, h is the number of attention heads, W_i^Q, W_i^K, W_i^V and W_i^O are the projection matrices for Query (Q), Key (K), Value (V) of the head i, and output (O) of the multi-head attention respectively. This mechanism is used both at sentence and document levels. Also, it is important to highlight that it do not consider the word order and, due to this fact, it is necessary to incorporate positional information to the system.

To do this, we define a function $\mathrm{P}_1 : \mathbb{R}^{d_e} \to \mathbb{R}^{d_e}$ that is applied independently to each word with the aim of identifying its position in the input of the sentence encoder. Our choice for P_1 was the sine-cosine function proposed in [11], that exploits the cyclic nature of sine and cosine functions to represent the positional information. Then, from X and X', $R \in \mathbb{R}^{T\times d_w}$ for article and $R' \in \mathbb{R}^{Q\times d_w}$ for summary are computed by using Transformer Encoders as sentence encoders, following Eqs 5.14 and 5.15.

$$R_i = \frac{1}{W}\sum_{j=1}^{W} \mathrm{TE}_1(\mathrm{P}_1(\mathrm{E}(X_i))) \tag{5.14}$$

$$R'_i = \frac{1}{V}\sum_{j=1}^{V} \mathrm{TE}_1(\mathrm{P}_1(\mathrm{E}(X'_i))) \tag{5.15}$$

where the $N = 1$ layered Transformer Encoder $\mathrm{TE}_1(\cdot;\theta_1)$ is defined in Eq. 5.16. Note that, if $N > 1$ Transformer Encoder layers are used, the output of the TE_1 in the layer i is used as input for the TE_1 in the layer $i+1$.

$$\mathrm{TE}_1 = \mathrm{LayerNorm}(L+F) \tag{5.16}$$

$$F = \max(0, LW_1 + b_1)W_2 + b_2 \tag{5.17}$$

$$L = \mathrm{LayerNorm}(\cdot + \mathrm{MultiHead}(\cdot,\cdot,\cdot)) \tag{5.18}$$

where the weight matrices of the multi-head attention mechanism (Eqs 5.11 and 5.12) are defined for the sentence level as $W_i^Q \in \mathbb{R}^{d_e \times d_k}$, $W_i^K \in \mathbb{R}^{d_e \times d_k}$, $W_i^V \in \mathbb{R}^{d_e \times d_k}$ and $W_i^O \in \mathbb{R}^{(h \cdot d_k) \times d_w}$, and additionally, are shared among the two branches; $W_1 \in \mathbb{R}^{d_w \times d_{fw}}$, $W_2 \in \mathbb{R}^{d_{fw} \times d_w}$, $b_1 \in \mathbb{R}^{d_{fw}}$ and $b_2 \in \mathbb{R}^{d_w}$ are the weights and the bias respectively of the position wise feed-forward network; and LayerNorm refers to Layer Normalization [52]. This process is independently applied to each word embedding matrix that represents each sentence, both for document and summary ($R_i : 1 \leq i \leq T$ and $R'_i : 1 \leq i \leq Q$).

From R and R', $r \in \mathbb{R}^{d_s}$ and $r' \in \mathbb{R}^{d_s}$ can be obtained, following Eqs. 5.19 and 5.20, similarly to the sentence level but using TE_2 for document and summary respectively. Note that, due to Transformer Encoders are applied on top of the sentence representations, it is possible to include positional information also to take into account the position of the sentences both in documents and summaries. To do this, a function $\mathrm{P}_2 : \mathbb{R}^{d_w} \to \mathbb{R}^{d_w}$ is defined, that is also a sine-cosine function like P_1, but applied to each sentence independently, with the aim of incorporating sentence positional information in the input of the encoder at document level.

$$r = \frac{1}{T}\sum_{j=1}^{T} \mathrm{TE}_2(\mathrm{P}_2(R)) \tag{5.19}$$

$$r' = \frac{1}{Q}\sum_{j=1}^{Q} \mathrm{TE}_2(\mathrm{P}_2(R')) \tag{5.20}$$

where $\mathrm{TE}_2(\cdot;\theta_2)$ composed by $\hat{N} = 1$ layer is defined in the same way that TE_1, following Eq. 5.21. If $\hat{N} > 1$, the output of the TE_2 in the layer i is used as input for the next layer $i+1$.

$$\mathrm{TE}_2 = \mathrm{LayerNorm}(\hat{L}+\hat{F}) \tag{5.21}$$

$$\hat{F} = \max(0, \hat{L}\hat{W}_1 + \hat{b}_1)\hat{W}_2 + \hat{b}_2 \tag{5.22}$$

$$\hat{L} = \text{LayerNorm}(\cdot + \text{MultiHead}(\cdot,\cdot,\cdot)) \tag{5.23}$$

where the weight matrices of the multi-head attention mechanism (Eqs 5.11 and 5.12) are defined for the document level as $W_i^Q \in \mathbb{R}^{d_w \times d_k}$, $W_i^K \in \mathbb{R}^{d_w \times d_k}$, $W_i^V \in \mathbb{R}^{d_w \times d_k}$ and $W_i^O \in \mathbb{R}^{hd_k \times d_s}$, and additionally, are shared among the two branches; $\hat{W}_1 \in \mathbb{R}^{d_s \times d_{fs}}$, $\hat{W}_2 \in \mathbb{R}^{d_{fs} \times d_s}$, $\hat{b}_1 \in \mathbb{R}^{d_{fs}}$ and $\hat{b}_2 \in \mathbb{R}^{d_s}$.

From the vectors r and r', the interaction between them is computed as their concatenation with their absolute difference. This interaction is used as input for a feed-forward network whose output is a probability distribution over $\mathbb{C} = \{0, 1\}$, as defined in Eq. 5.10.

It is interesting to note the main difference of SHTE with respect to SHA-NN. In SHA-NN, BLSTM are used to compute the representations, combined with attention mechanisms to average them. As the attention mechanism partly control the impact of each sentence in the final representation, this score can be used to rank the sentences. However, in SHTE the same attention mechanism computes both the representations and the relevance scores. Due to this fact, the relevance of each sentence is implicitly captured by the multi-head self-attention mechanism. This system considers that a document sentence is more relevant the more attended it is by all the sentences of the document. With the aim of building a ranking over the document sentences, we use the attention matrices of the last Transformer Encoder at document level, obtained after a forward pass on the left branch of the network from an input document, following Eqs. from 5.24 to 5.26.

$$G_i = \text{softmax}\left(\frac{Q_i K_i^\intercal}{\sqrt{d_k}}\right) \tag{5.24}$$

$$H_{ij} = \frac{1}{h}\sum_{k=0}^{h} G_{kij} \tag{5.25}$$

$$\alpha_j = \frac{1}{T}\sum_{i=0}^{T} H_{ij} \tag{5.26}$$

where $Q_i, K_i \in \mathbb{R}^{T \times d_k}$ are the Queries and Keys in head i, $G_i \in \mathbb{R}^{T \times T}$ is the attention matrix of head i, $H \in \mathbb{R}^{T \times T}$ is the averaged attention of all the heads, and $\alpha \in \mathbb{R}^T$ is the vector that contains the final score assigned to each sentence j.

The system is composed by h different attentions that explain different relationships among the sentences. As it is shown in Eq. 5.25, we consider that all the relationships

captured by the self-attention mechanism have the same relevance to obtain a score. For this reason, the most attended sentences, in average among the different relationships (attentions), are considered as the most relevant.

After computing the average attention of all the heads, H, the component H_{ij} represents the average attention that the model assigns to the sentence j when it is processing the sentence i. Then, it could be used to compute the relevance of a sentence j in the document, based on the average attention that j receives of all the sentences of the document, following the Eq. 5.26. This process is used to compute the scores for all the sentences, and the scores are used to rank them for selecting the top-k most relevant document sentences in order to compose the summary.

5.2 Summarization of News Articles

Summarizing news articles is particularly interesting as, differently from other domains like scientific papers or medical reports, it is of public interest to the whole society[9]. This kind of resources allow us to connect and to find out current events, and we are constantly exposed to large amounts of this unstructured information by means of TV news, newspapers, or radio programs. Furthermore, some articles are especially hard to read and understand as they might require an in-depth knowledge of the topic discussed, which makes it difficult to identify key aspects for a less experienced and informed reader.

For this reason, almost all the online newspapers that publish news articles, highlight the key aspects that the authors consider relevant in order to make the information more accessible and attractive to any type of public. This is typically done by means of structural components of the articles like headlines and summary bullets, that, typically are composed by a single sentence. So, it should be noted that, as the length of the document increases, the probability that all its relevant aspects can be condensed into a single sentence decreases. This is one of the reasons why the current datasets leave the summarization task underconstrained [268].

Either way, the summarization corpora extracted automatically by using these structural components have been a great revolution in the automatic summarization field, becoming the *de facto* standard for training, evaluating and studying automatic summarization systems [274, 275, 298, 312, 313]. These corpora include: Gigaword [312], built from the annotated Gigaword dataset [313] by means of pairing headlines with the first sentence of the articles;

[9]As it can be seen in the Alexa rank of the most visited websites from Spain, `https://www.alexa.com/topsites/countries/ES`, 5 online newspapers are in the top 25 most visited websites (*Marca*, *El Pais*, *El Mundo*, *As* and *ABC*)

CNN/DailyMail [289], built from the corpus for passage-based question answering [298], where articles are paired with the concatenation of their bullet summaries (it should be noted that, in this case, differently from [313], full articles are considered and the summaries contain several aspects keys in a natural way); NewsRoom [275], built in the same way than CNN/DailyMail but more diverse in terms of summarization strategies and different newspapers; and XSum [274], built following the methodology of [298] from BBC articles.

In this section, we present the experimentation carried out for news articles summarization, by using our systems, proposed under the attentional extractive framework, for the CNN/DailyMail and NewsRoom corpora.

5.2.1 State of the Art

In this subsection we describe the state-of-the-art deep learning approaches in the reference corpora for news articles summarization [274, 275, 298, 312, 313]. For the sake of clarity, we describe the systems by distinguishing them in terms of the strategy they use to address the summarization problem (extractive, abstractive and hybrid).

One the one hand, extractive systems that extract spans of text (typically sentences) to compose summaries. These systems, in turn, can be divided in two different categories: those which use an oracle algorithm, based in ROUGE, to label the sentences of the documents before the training of the models, and those which optimize directly the ROUGE evaluation metric by means of RL strategies. Also, it is convenient to mention a simple heuristic that, although it is not a neural approach, it is especially effective for news articles, Lead-k. It is based on extracting the first k sentences of the documents to compose a summary. Although it seems naive, it is especially robust when it is applied on news articles, generally due to, in this domain, the first sentences (first paragraph) are dedicated to summarize the main ideas of all the document and they are used to grab the attention of the reader. Therefore, it is commonly considered as a lower bound for news summarization [314].

Regarding the extractive systems based on oracles, the first approaches were proposed in [288] and [290]. In [288] a encoder-decoder approach for extractive single-document summarization was proposed. In [290] (SummaRunner), the authors presented two versions of Hierarchical Attention Networks to select sentences from the documents as a binary sequence classification problem. One of these versions is trained using directly the samples provided by the corpus. The other version, requires a greedy algorithm as oracle for labeling the corpus at sentence level, selecting as reference summary the set of sentences from the document that maximize the similarity with respect to the reference summary. Another interesting approach was proposed in [315] where they jointly learn the attention mechanism, to obtain the score of the sentences, and the selection mechanism to extract the

most salient sentences. Recently, the great impact of the Transformer architecture [11] in Natural Language Processing tasks, and particularly in language modeling [57, 82, 83], have boosted the results in extractive summarization by finetuning powerful pretrained language models. The most relevant extractive example is BertSumEXT system [294], which is based on the finetuning of BERT models [57]. The authors of [294] also proposed abstractive and hybrid strategies for generating summaries starting from the pretrained BERT. A novel paradigm for extractive summarization is based on text matching (MatchSum) [295]. This paradigm is highly related to our attentional extractive framework, in the sense that both compute document, reference and distractor representations, and they leverage a siamese approach to represent the references closest to the source documents. However, while for SHA-NN and SHTE the distractors are randomly sampled (easy negatives), in MatchSum, they are sampled from pretrained systems like [294] (hard negatives). Furthermore, they consider a pairwise margin loss to induce ROUGE preferences among distractors (highest ROUGE-ranked distractors have to be closest to the documents, than lowest ROUGE-ranked distractors).

All the oracle-based systems suffer from the ROUGE/cross-entropy mismatch [292], derived from a discrepancy between the task definition and the training objective. This is the main drawback of the summarization systems based on optimizing the cross-entropy instead of the ROUGE measure. For this reason, RL extractive strategies for automatic summarization have received a great interest by the research community. In spite of the first works on RL were intended to perform abstractive summarization [269], recently, these strategies has been widespread for extractive text summarization, optimizing directly the ROUGE evaluation measure [292, 293, 305–307]. In [292] (Refresh), the authors argue about the application of cross-entropy with ground-truth sentence labels to optimize neural summarization models, and they propose the application of the REINFORCE algorithm [316] for extractive summarization in order to train a hierarchical encoder-decoder. In [305], the authors also discussed about the suboptimal nature of the labels obtained by means of oracles. They present a latent variable extractive model which can also be viewed as a RL approach where the reward is defined as a weighted sum of two measures related to the precision and the recall. These measures were computed from the likelihood of a summary sentence and a document sentence, estimated by means of an attention-based sequence-to-sequence sentence compression model. This system can be trained in an extractive (Latent) or in a compressive way (Latent-Comp). In [293] (BanditSum) a theoretically grounded method was proposed for modeling the extractive summarization problem by means of a bandit formalism. The authors proposed a novel structure for computing the conditioned probability of a subset of document sentences given the document, which avoids privileging early sentences over later

ones. In [306] (DQN) an approach based on Deep Q Learning was proposed. This approach is based on an iterative decision problem, where a sentence is selected at each timestep. After each sentence selection, the state of the model is updated and the selected sentence is added to the summary state, which contains the set of selected sentences until the current timestep.

Regarding the abstractive systems, all of them are based on encoder-decoder models with attention mechanisms [95, 96, 269, 289, 291, 296, 297, 312, 317]. First approaches suffered from known problems of the traditional sequence-to-sequence approaches: repetitions, grammatically incorrect generations, lack of coherence, coverage, hallucination (especially factual inconsistency), and the inability of producing words out of the training vocabulary [269, 289, 312]. To address these issues, some systems, capable of selecting or generating a new word at each timestep, were proposed [291, 317] (they can also be seen as a kind of hybrid end-to-end systems). The most relevant example is [291], where an approach based on Pointer Networks and encoder-decoder models with attention mechanisms is proposed. In order to address the repetition problem, the authors enrich their system by using a coverage mechanism based on the attentions of previous timesteps, for each decoder timestep (PointerGen+Cov). This system has been modified by the authors of [317], replacing the backbone architecture (Long Short-Term Memories [10]) by transformers. Most of these issues (grammatically incorrect generations, repetitions, and the inability of producing out of vocabulary words) have been recently overcome by transfer learning, being now more competitive against the extractive approaches. Concretely, this has been reached by means of finetuning transformers, pretrained in a self-supervised way for language generation [95, 96, 296, 297]. In [296], a unified framework for addressing universally all text-based language problems in a text-to-text format was proposed. They pretrained a transformer model with 11 billions of parameters (T5) on a denoising task similar to MLM, and after, they finetuned the model on a wide variety of tasks, including text summarization. The authors of [96] proposed a pretraining self-supervised objective tailored for abstractive summarization, Gap Sentences Generation (GSG). It is based on masking full sentences and concatenate them as a pseudo-summary for being reconstructed by the decoder. Along with a MLM objective, GSG was also used for pretraining a large transformer model on a massive text corpus (Pegasus). In [297], a sparse attention mechanism was proposed to reduce the quadratic dependency of the full attention mechanisms into linear dependencies without loss of generality. It was applied along with Pegasus, obtaining similar results with a lower complexity (BigBird-Pegasus). In [95], a denoising autoencoder for pretraining sequence-to-sequence models was proposed (BART). It is pretrained with a wide variety of denoising tasks: sentence permutation, document rotation, text infilling, token deletion and MLM, where the model is expected to reconstruct the original text. Also, some works have

been proposed in order to increase the faithfulness of the generated summaries, especially by means of guidance mechanisms [279, 318–321].

Nowadays, practically there is not difference among the performances of abstractive and extractive systems in the considered corpora [95, 96, 295], however, the abstractive approaches still have problems related to semantic aspects like hallucination as they are trained on standard likelihood and approximate decoding objectives [1]. Also, the positional bias inherent to the news articles corpora acts as an inductive bias in the training of all these models, which tend to excessively focus on the first sentences [314, 322]. However, as we show in [27], systems with a strong positional bias to the first sentences, like SHA-NN, are able to detect salient sentences that are not at the beginning of the documents, when they are applied on other domains that have not got positional bias.

Due to the recent success of generative approaches, the interest in hybrid strategies have been decreased, but they could be determinant when the improvements of the generative systems reach a saddle point, for example, to reduce factual hallucinations. These approaches are typically based on first extracting a set of sentences and later adapting them to the reference summaries e.g., compressing or paraphrasing [278, 279, 307]. In [278] the authors proposed a compressive approach that removes unnecessary words while keeping the summaries informative, concise and grammatically correct. The model can be trained in an extractive way (ExConSumm-Ext) and in a compressive way (ExConSumm-Comp). Finally, in [307], the authors proposed a sentence-level policy gradient method for first select salient sentences and then paraphrases them (Fast-RL).

5.2.2 Corpora

We carried out the experimentation by using two different corpora for newspaper summarization. On the one hand, the CNN/DailyMail[10] corpus was used in this work. This corpus, which is a set of articles from the CNN and DailyMail news websites, was originally constructed for passage-based question answering [298] and modified for abstractive and extractive summarization [288, 289]. The CNN/DailyMail corpus was partitioned into 287,227 training (article, summary) pairs, 13,368 validation (article, summary) pairs and 11,490 test (article, summary) pairs. In order to compare our systems with most of the works on this corpus, we used the non-anonymized version. It should be noted that the ground truth summaries provided by this corpus are abstractive (although they have a strong extractive tendency [274, 275]), and they were constructed by concatenation of the highlights associated to the documents.

[10] `https://cs.nyu.edu/~kcho/DMQA/`

On the other hand, the NewsRoom[11] corpus, proposed in [275] for the document summarization task, was also used. It consists of 1.3 million articles and summaries that have been written by the authors and the editors of 38 different major news publications. The corpus was created through a web-scale crawling of over 100 million pages from a set of online publishers by gathering the news and using the summaries provided in the HTML metadata. The summaries contained in this corpus combine both extractive and abstractive strategies to describe the content of the articles. The NewsRoom corpus was partitioned into 995,041 training (article, summary) pairs, 108,837 validation (article, summary) pairs and 108,862 test (article, summary) pairs. In turn, each set of NewsRoom is divided in different subsets in terms of the degree of abstractiveness, measured be means of statistics based in novel n-grams like coverage and density [275].

Some characteristics of both corpora are presented in Table 5.1. It is important to note that the NewsRoom corpus is much bigger than the CNN/DailyMail corpus as stated before. Regarding the number of article sentences and words in all the sample sets, both corpora are very similar. However, reference summaries are twice as long in CNN/DailyMail than in NewsRoom.

Table 5.1 Average number of sentences and words, including words per sentence, for both corpora.

		Sentences		Words		Words/Sentence	
Corpus	Set	Articles	Summ	Articles	Summ	Articles	Summ
CNN/DailyMail	Train	31.87	3.79	750.10	51.58	23.53	13.61
	Dev	26.77	4.11	737.06	57.57	27.53	14.00
	Test	27.11	3.88	745.59	54.65	27.51	14.07
NewsRoom	Train	29.91	1.40	773.57	30.37	25.86	21.65
	Dev	29.69	1.41	767.34	30.72	25.84	21.73
	Test	29.62	1.41	765.56	30.63	25.84	21.68

5.2.3 Evaluation

In this section, we show and discuss the results obtained by the systems of the *Attentional Extractive Summarization* framework (SHA-NN and SHTE, described in §5.1.1 and §5.1.2 respectively) on the CNN/DailyMail and NewsRoom corpora. We also performed comparisons with other approaches, including the state-of-the-art systems that have appeared more recently after our work. The evaluation of the performance of the systems have been done by using three variants of the ROUGE measure [265]. Concretely, ROUGE-N with unigrams

[11] `https://summari.es/`

and bigrams (R-1 and R-2) and ROUGE-L (R-L). A formal definition of the Precision, Recall and F_1 ROUGE metrics is shown in Eqs. 5.27 to 5.32:

$$\text{ROUGE-N}(C,S)_P = \frac{|\text{n-grams}(C) \cap \text{n-grams}(S)|}{|\text{n-grams}(C)|} \tag{5.27}$$

$$\text{ROUGE-N}(C,S)_R = \frac{|\text{n-grams}(C) \cap \text{n-grams}(S)|}{|\text{n-grams}(S)|} \tag{5.28}$$

$$\text{ROUGE-N}(C,S)_{F_1} = \frac{2 \cdot \text{ROUGE-N}(C,S)_P \cdot \text{ROUGE-N}(C,S)_R}{\text{ROUGE-N}(C,S)_P + \text{ROUGE-N}(C,S)_R} \tag{5.29}$$

$$\text{ROUGE-L}(C,S)_P = \frac{\sum_{s \in S} \bigcup_{c \in C} \text{LCS}(s,c)}{\sum_{c \in C} |c|} \tag{5.30}$$

$$\text{ROUGE-L}(C,S)_R = \frac{\sum_{s \in S} \bigcup_{c \in C} \text{LCS}(s,c)}{\sum_{c \in S} |s|} \tag{5.31}$$

$$\text{ROUGE-L}(C,S)_{F_1} = \frac{2 \cdot \text{ROUGE-L}(C,S)_P \cdot \text{ROUGE-L}(C,S)_R}{\text{ROUGE-L}(C,S)_P + \text{ROUGE-L}(C,S)_R} \tag{5.32}$$

where C is a candidate summary, S is a reference summary, n-grams(X) is the word n-grams multiset[12] of X, $s \in S$ are the sentences in the reference summary, $c \in C$ are the sentences in the candidate summary and LCS stands for Longest Common Subsequence. In this and following subsections, if not specified otherwise, we use the F_1 ROUGE metrics.

The hyper-parameters used for SHA-NN and SHTE are as follows. On the one hand, for SHA-NN system, we used pretrained word embeddings, obtained by means of a $d_e = 300$-dimensional skip-gram architecture, trained from the articles of the corpora. These embeddings were frozen during the training of the models. We used $N = 1$ sentence encoders and $\hat{N} = 1$ document encoders with $d_w = d_s = 512$. Adam [55] was used as update rule with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize the cross-entropy. In order to train the model for both corpora, we used batches of 64 (article, summary) pairs (32 positive and 32 negative randomly sampled following an uniform distribution) and we considered that a training epoch finishes after 500 batches. To extract the summaries, the top-k most relevant sentences, by following directly the attention score of the document encoder, were selected. On the other hand, for SHTE system, we used randomly initialized word embeddings with $d_e = 128$ which were trained simultaneously with the model. Most of the hyper-parameters were also fixed, such as $N = 2$ word encoders and $\hat{N} = 2$ sentences encoders, $h = 6$ heads, $d_k = d_v = d_q = 64$,

[12] The equations shown for ROUGE-N (5.27, 5.28 and 5.29) are simplifications of those proposed in [265], as we consider only one reference summary, and they can be interpreted as an intersection of n-grams multisets e.g., for unigrams, $C = aab \rightarrow \{a1, a2, b1\} \wedge S = aaabb \rightarrow \{a1, a2, a3, b1, b2\}$

$d_w = d_s = d_{fw} = d_{fs} = d_e$, P_1 is the identity function (we do not add positional information to the words inside each sentence) and P_2 is the sine-cosine function defined in [11]. We only used positional information on the sentences due to the empirical results obtained in [26], where positional information in sentences seems working better than positional information in words. Adam [55] was used as update rule with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize the cross-entropy, and Noam was used as learning rate scheduler with $warmup\_steps$ = 4000. To train the model with CNN/DailyMail we used batches of 64 (article, summary) pairs (32 positive and 32 negative randomly sampled following an uniform distribution). For training with NewsRoom, we used batches of 128 (article, summary) pairs. In both experiments, we considered that a training epoch finishes after 5000 batches. In order to extract the summaries, the top-k most relevant sentences, by following the scoring mechanisms presented in §5.1.2 at document level, were selected. For both systems we used early stopping with 20 epochs of tolerance during the training phase. For the summarization phase, both models extracted the $k = 3$ most relevant sentences for the CNN/DailyMail corpus and $k \in \{2,3\}$ for the NewsRoom corpus. All the experiments were performed in a single GPU NVIDIA GeForce RTX 2080.

In Table 5.2, the results of our systems and other state-of-the-art systems for the CNN/DailyMail corpus are shown[13]. It can be seen how our systems obtain better results than other widely used systems PGen+Cov [291], CopyCat [317] and SummaRunner [290]. The obtained results are worse in comparison to other extractive systems that use oracles, especially in the case of BertSumEXT [294], in spite of our systems share the same backbone architecture (transformer encoders). This illustrates the big boost that the extractive systems have recently obtained by means of finetuning powerful pretrained language models [57, 83] for the summarization task. Interestingly, the recently proposed text matching approach (MatchSum [295]), that is very similar to the objective of our attentional extractive framework, has been usefully applied for filtering summary candidates extracted from competitive summarization systems like BertSumEXT. This system is currently the best performing one in the CNN/DailyMail dataset. Also, it is interesting to observe that the results obtained by our systems are better than those obtained from some RL based systems such as DQN [306] and similar to Refresh [292], although, in general, the RL strategies seem helpful for improving the results. Therefore, our extractive summarization framework could be used as an alternative to RL approaches and oracle-based systems. Regarding the abstractive systems, it can be seen how, training from scratch with a maximum likelihood objective is not enough to make them comparable to the extractive systems (PGen+Cov and

[13]Most of these works were published after our work, but we considered them in this thesis with the aim of showing the evolution of the results for this task.

ML (w/ Intra-Attention) [269]). The results are improved if RL techniques are considered (RL (w/ Intra-Attention) [269]). However, similarly to the extractive systems, the highest performance boost has been obtained by finetuning massive language models pretrained on language generation tasks [95, 96, 296, 297].

Table 5.2 Results on CNN/DailyMail corpus for full-length Rouge. The strategy followed by each system is also specified, where Ext, Abs and Hyb are the stands of extractive, abstractive and hybrid (OC stands for *oracle*).

System	Strategy	R-1	R-2	R-L
Lead-3 (our)	Ext	40.24	17.70	36.45
SHA-NN (our)	Ext	39.99	17.75	36.27
SHTE (our)	Ext	39.96	17.60	36.19
SummaRunner [290]	Ext/OC	39.60	16.20	35.30
ExConSumm-Ext [278]	Ext/OC	41.70	18.60	37.80
BertSumEXT [294]	Ext/OC	43.25	20.24	39.63
MatchSum (Bert-based) [295]	Ext/OC	44.22	20.62	40.38
MatchSum (Roberta-based) [295]	Ext/OC	**44.41**	20.86	40.55
Refresh [292]	Ext/RL	40.00	18.20	36.60
DQN [306]	Ext/RL	39.40	16.10	35.60
Latent [305]	Ext/RL	41.10	18.80	37.40
BanditSum [293]	Ext/RL	41.50	18.70	37.60
PGen+Cov [291]	Abs	39.53	17.28	36.38
CopyCat [317]	Abs	39.15	17.60	36.17
ML (w/ Intra-Attention) [269]	Abs	38.30	14.81	35.49
RL (w/ Intra-Attention) [269]	Abs	41.16	15.75	39.08
T5 [296]	Abs	43.52	**21.55**	40.69
Pegasus [96]	Abs	43.90	21.20	40.76
BigBird-Pegasus [297]	Abs	43.84	21.11	40.74
BART [95]	Abs	44.16	21.28	**40.90**
ExConSumm-Comp [278]	Hyb	40.90	18.00	37.40
Latent-Comp [305]	Hyb	36.70	15.40	34.30

Tables 5.3 and 5.4 show the results, in terms of ROUGE, on the NewsRoom corpus. Specifically, Table 5.3 shows the results on the full test set and Table 5.4 shows the results on each one of the three test subsets defined in [275]. Each subset makes reference to the extractiveness degree of their summaries, measured in terms of the density metric proposed also in [275]. There are 3 different subsets: NR-Ext (subset whose reference summaries have high density of words that appear in the articles), NR-Mix (subset with medium density) and NR-Abs (subset whose reference summaries have a low density and, then, it can be considered as abstractive).

Table 5.3 Results on the full test of NewsRoom.

System	Strategy	R-1	R-2	R-L
Lead-3 (our)	Ext	30.66	21.09	28.35
SHA-NN ($k = 3$) (our)	Ext	28.99	19.42	26.69
SHTE ($k = 3$) (our)	Ext	29.19	19.37	26.81
Lead-2 (our)	Ext	33.98	23.30	31.14
SHA-NN ($k = 2$) (our)	Ext	32.78	21.86	29.85
SHTE ($k = 2$) (our)	Ext	32.38	21.25	29.40
ExConSum-Ext	Ext	**39.50**	**27.90**	**36.26**
PGen+Cov	Abs	26.43	13.76	22.90
TLM [323]	Hyb	33.30	20.06	29.26
FastRL [306]	Hyb	21.93	9.37	19.61
ExConSum-Comp	Hyb	39.06	27.36	36.13

Table 5.4 Results on the three test subsets of NewsRoom (Extractive, Mixed and Abstractive).

		NR-Ext			NR-Mix			NR-Abs		
System	Strategy	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Lead-3	Ext	51.98	47.85	51.20	25.62	13.00	22.30	14.57	2.62	11.73
SHA-NN ($k = 3$)	Ext	48.29	43.54	47.42	24.62	12.32	21.37	14.22	2.57	11.43
SHTE ($k = 3$)	Ext	48.62	43.35	47.65	24.76	12.20	21.43	14.33	2.53	11.51
Lead-2	Ext	57.87	53.03	56.83	28.60	14.33	24.46	15.68	2.77	12.35
SHA-NN ($k = 2$)	Ext	54.83	49.25	53.72	28.03	13.79	23.85	15.67	2.74	12.29
SHTE ($k = 2$)	Ext	53.97	47.87	52.59	27.78	13.41	23.56	15.57	2.67	12.22
ExConSum-Ext	Ext	**69.40**	**64.30**	**68.30**	**31.90**	**16.30**	26.90	17.20	3.10	13.60
PGen+Cov	Abs	39.10	28.00	36.20	25.50	11.00	21.10	14.70	2.30	11.40
TLM	Hyb	53.30	44.20	50.10	28.10	12.10	23.00	**18.50**	**3.90**	**14.70**
FastRL	Hyb	-	-	-	-	-	-	-	-	-
ExConSum-Comp	Hyb	68.40	62.90	67.30	31.70	16.10	**27.00**	17.10	3.10	14.10

It is possible to observe how to extract a number of sentences similar to the reference summary length (1.4 as shown in Table 5.1) improves notably the performance of the systems ($k = 2$ instead of $k = 3$). This behavior is observed especially in the NR-Ext and NR-Mix subsets, in comparison to the NR-Abs subset. This suggests that, when the reference summaries are extractive, in addition to determine the relevance of each sentence, it is also important to adjust correctly the length of the summaries. However, when the reference summaries are abstractive, the results by using $k = 2$ and $k = 3$ are very similar and clearly lower for all the systems. These bad results are due to the abstractiveness nature of this set of reference summaries, taking into account that the systems are extractive and hybrid. Also, it is interesting to highlight that, although Lead is a robust baseline in the NR-Ext and NR-Mix subsets, it is not so good in the NR-Abs subset, where our systems obtain almost the same results.

In both cases, the results obtained by our systems, are better than those obtained by the widely adopted PGen+Cov or by RL based systems such as FastRL [307]. Also, they obtain better results than TLM [323] in terms of ROUGE-2 and ROUGE-L on the full dataset, in spite of this system stands out in the abstractive subset. The only systems that consistently outperforms the Lead heuristic are those based on ExConSumm (both in the extractive and hybrid variants), mainly due to they largely outperform the results on NR-Ext and NR-Mix subsets. Differently from our systems, these systems are able to generate variable-length summaries depending on the input text.

It is interesting to observe in Table 5.5 that, in spite of the significant differences in terms of loss and accuracy during the evaluation with the development set, the results in terms of ROUGE in the evaluation of the test summaries are very similar. This clearly illustrates the mismatch discussed in [292], derived from the disconnect between the task definition and the training objective. This is the main drawback of the summarization systems based on optimizing the cross-entropy instead of the ROUGE measure. Due to this reason, it is interesting to search alternatives to RL in order to optimize directly the evaluation measure.

5.2.4 Analysis

In this section, we present several analyses to study the behavior of the systems. Specifically, we analyze the convergence of the systems, the length of their generated summaries and their bias to early sentences in some specific examples from the test sets. We also provide specific analyses for the SHTE model focused on the impact of the positional information and the strategy of averaging attentions from all the heads to rank the sentences.

First, we analyzed the systems in terms of their convergence. In Table 5.5, several details about the convergence of our systems are shown. Specifically, it shows the number of samples that each system has seen until convergence, the value of the loss function, the accuracy on the development set (for each sample in the development set, two samples are built, one positive and one negative randomly sampled), and the time until the convergence. It is possible to see how the SHTE model visited a large number of samples during the training until convergence, at the same time that obtains significantly worse results in terms of accuracy. However, the time required to train these models is significantly lower, requiring up to four times lower than SHA-NN for the NewsRoom corpus. Furthermore, as Tables 5.2 and 5.3 show, the results in terms of ROUGE on both corpora are very similar for both systems. Thus, SHTE constitutes an efficient alternative to SHA-NN since, with a lower training time, obtains very similar results in terms of ROUGE. In comparison to other systems such as BanditSum [293] (76 hours in a single NVIDIA Geforce Titan Xp), DQN [306] (10 days on a single NVIDIA GeForce GTX 1080) or Refresh [292] (12 hours "on a single GPU"), both systems require a

significantly lower training time for the CNN/DailyMail corpus. Furthermore, they dispense with the computation of sentence oracles previously to the training step.

Table 5.5 Convergence statistics of our systems.

Corpora	System	Samples	Loss	Acc	Time (h)
CNN/DailyMail	SHA-NN	**2,624,000**	**0.007**	**99.62 ±0.10**	3.51
	SHTE	4,160,000	0.209	91.92 ±0.46	**2.38**
NewsRoom	SHA-NN	**5,088,000**	**0.083**	**96.16 ±0.11**	6.45
	SHTE	5,760,000	0.230	90.61 ±0.17	**1.65**

Following the experimentation carried out in [278], we analyzed the lengths of the summaries generated by our proposals. Figure 5.5 shows the word-length distributions of the summaries for Lead, SHA-NN and SHTE systems (with $k \in \{2,3\}$) applied on CNN/DailyMail corpus and NR-Ext subset of NewsRoom. We included also the word distribution of the human reference summaries for both corpora.

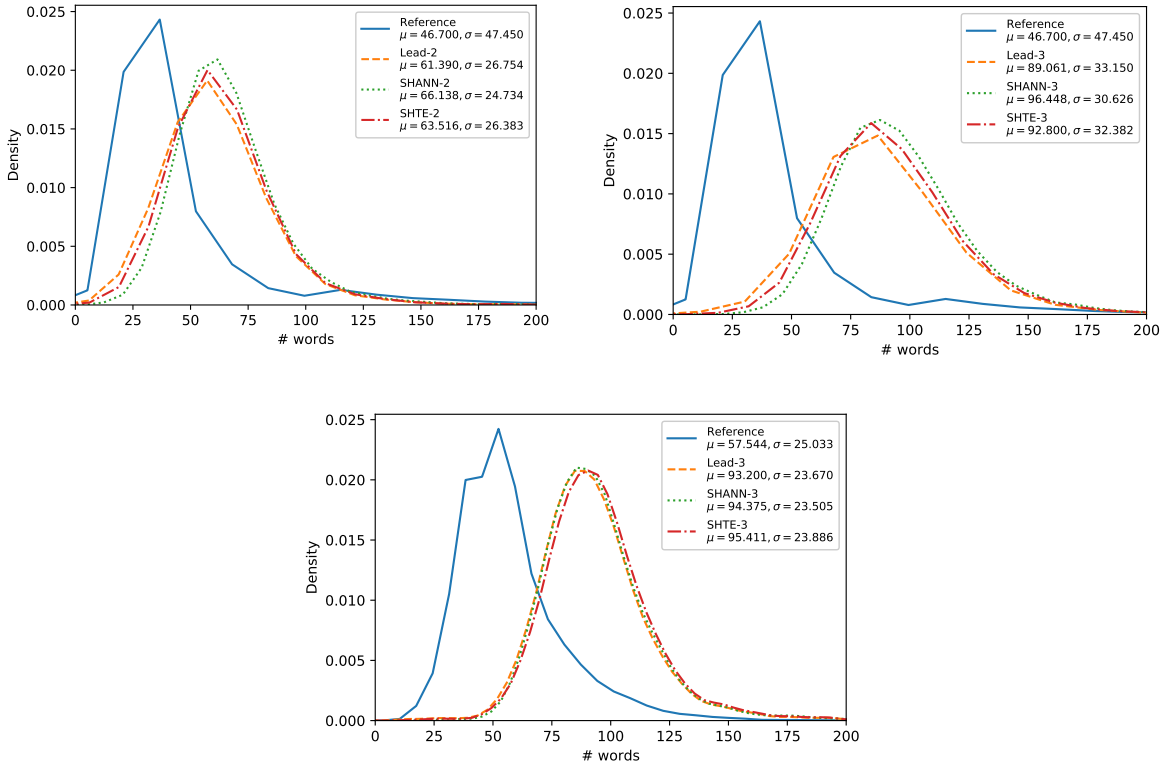

Fig. 5.5 Word-length distribution of system generated summaries in comparison to human reference summaries for NR-Ext ($k = 2$) and NR-Ext ($k = 3$) (top left and right respectively), and CNN/DailyMail (bottom).

It can be seen that the word-length distributions of the summaries extracted by our proposals are almost identical to the distribution of the Lead heuristic. This similarity can be

observed also in other systems, based on RL which dispenses with oracles, such as Latent [305] and Refresh [292], as shown in [278]. These results suggest that the extractive systems that do not use oracles, are biased to select the first sentences in a higher extent than oracle-based systems. For both corpora, all the system distributions are shifted considerably to the right in comparison to the distribution of the human reference summaries. Thus, our systems seem not to be able to generate summaries in lower length ranges (12-50 for CNN/DailyMail, 5-25 for NR-Ext with $k = 2$ and 20-50 for NR-Ext with $k = 3$). This is mainly due to they are not able to build variable-length summaries and they are limited to select all the words of a fixed number of sentences without making word-level operations e.g., compression [278] or selection [291].

Regarding the analyses of SHTE, we consider two interesting aspects to be analyzed. The first one consists in the impact of the positional information on the selection of the most relevant sentences. Concretely, we explore three ways for the incorporation of positional information: i) just at the sentence level, ii) both at word and sentence level; and iii) without positional information. We used in this analysis the CNN/DailyMail corpus, where the first sentences of the documents tend to be the most representative sentences to compose the summary. This is due to the journalistic style, that tries to grab the attention of the reader in the first paragraph of the articles. For this reason, we expect the sentence positional information to be especially relevant.

The second aspect to analyze is the strategy of averaging attentions from all the heads of our model in order to rank the sentences. We hypothesize that the combination of all the relationships captured by the different heads is more adequate than individual attentions captured by only one head for computing the relevance of each sentence. So, we try to show how the summarization problem requires to combine the different properties learned by the attention heads, which implies that there is not only a single attention head specialized on the task. It is important to highlight that we only used the attentions of the last encoder at sentence level because the relationships captured at this level are semantically richer than the relationships captured in the first encoder.

The results of this experimentation are shown in Table 5.6, where three blocks of experiments were done by varying the positional information (no positional, at sentence level, and both at word and sentence level). The column labelled as "Head" represents what head was used to assign the scores to the sentences (only one head or the average). On the one hand, it can be seen that the addition of positional information only at sentence level is more informative than its combination with positional information at the word level. The improvements obtained by adding positional information on the sentences seem to support the assumption of the importance of the sentence order in the generation of the summaries.

Moreover, both types of positional information provide better results than not using positional information. On the other hand, the strategy of averaging the attention heads is the best mechanism for sentence scoring in almost all the cases. Concretely, it obtains always the best results in terms of F_1 and it seems to have worse results in terms of Precision. Although the improvements are not statistically significant, it is possible to see that there are heads which capture less relevant relationships than others and the averaging of them with the remaining heads counters these low results. An interesting future work is the interpretation of these attention mechanisms and the search for combinations among them.

Table 5.6 Experimentation modifying the addition of positional information and the selected attention head to rank the sentences. Results were computed on the test set of the CNN/DailyMail corpus.

		Precision			Recall			F_1		
	Head	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
No Positional	1	24.28	7.92	21.79	45.06	15.15	40.38	29.75	9.80	26.68
	2	24.58	8.11	22.13	44.15	14.90	39.64	29.89	9.92	26.88
	3	**24.79**	7.97	**22.29**	43.48	14.42	38.98	29.64	9.62	26.62
	4	24.14	7.81	21.67	44.14	14.71	39.25	29.51	9.63	26.46
	5	24.49	7.94	22.02	43.40	14.39	38.90	29.61	9.66	26.58
	6	24.42	7.60	21.89	41.90	13.33	37.41	29.00	9.09	25.95
	Avg Heads	24.67	**8.23**	22.16	**45.45**	**15.53**	**40.73**	**30.20**	**10.15**	**27.10**
Sent Positional	1	27.79	11.07	25.21	51.31	20.78	47.34	34.76	13.82	31.51
	2	27.17	10.66	24.62	52.36	20.67	47.38	34.29	13.47	31.06
	3	29.19	11.71	26.53	51.74	20.86	46.98	35.83	14.39	32.55
	4	**29.84**	**12.09**	**27.15**	52.17	21.24	47.41	36.15	14.58	33.16
	5	29.12	11.87	26.48	**53.09**	**21.66**	**48.19**	36.03	14.68	32.74
	6	29.60	12.01	26.91	52.30	21.30	47.45	36.21	14.73	32.99
	Avg Heads	29.64	12.03	26.97	52.46	21.36	47.67	**36.36**	**14.76**	**33.37**
Sent-Word Positional	1	24.68	8.12	22.13	44.20	14.70	39.59	30.11	9.94	27.03
	2	23.91	7.84	21.51	44.34	14.87	39.79	29.45	9.74	26.47
	3	**25.83**	**9.69**	**23.32**	50.38	18.98	45.37	32.16	11.74	28.95
	4	23.59	7.66	21.18	43.99	14.61	39.39	28.98	9.48	25.98
	5	25.23	8.86	22.72	47.47	17.02	42.68	31.38	11.10	28.24
	6	23.94	7.49	21.56	39.29	12.76	35.82	28.35	8.94	25.49
	Avg Heads	25.33	9.42	22.84	**50.92**	**19.02**	**45.85**	**32.40**	**12.04**	**29.18**

In Figures 5.6 and 5.7, we show two examples of summaries generated by the SHTE and SHA-NN systems both for NewsRoom and CNN/DailyMail respectively.

In the NewsRoom example, it can be observed how both systems, in spite of the bias towards the first sentences, decide to dispense with the second sentence to generate the summary. Also, in addition to the first sentence, both select the third sentence that matches exactly with the reference summary. For the CNN/DailyMail example, both systems extract also the first article sentence. Along with it, SHTE extracts a sentence related with the reference summary and one irrelevant sentence. In the same way, this behavior is also

• **Article**: kylie jenner ' s twitter account was hacked on sunday , and the starlet took to snapchat to refute some of the messages the hacker had sent out . " well my sex tape with tyga was trash , " the hacker wrote on jenner ' s account . the 18-year-old responded to the mention of a sex tape with her now-ex-boyfriend , which has been rumored in the past . " everyone is like 'leak the sex tape , ' " jenner said in a video . " guys , you are never going to see a sex tape from me . it 's not going to happen . " the " keeping up with the kardashians " star also clarified that the messages that were sent out , which bashed stars like justin bieber , were not from her . " so my twitter was hacked , " she said in another quick video . " i do n't really care . " i 'm just letting them have fun . "

• **Reference**: the 18-year-old responded to the mention of a sex tape with her now-ex-boyfriend , which has been rumored in the past .

• **SHTE**: kylie jenner ' s twitter account was hacked on sunday , and the starlet took to snapchat to refute some of the messages the hacker had sent out . the 18-year-old responded to the mention of a sex tape with her now-ex-boyfriend , which has been rumored in the past .

• **SHA-NN**: kylie jenner ' s twitter account was hacked on sunday , and the starlet took to snapchat to refute some of the messages the hacker had sent out . the 18-year-old responded to the mention of a sex tape with her now-ex-boyfriend , which has been rumored in the past .

Fig. 5.6 Summarization of a NewsRoom test sample.

• **Article**: allan donald has confirmed he is to step down as south africa bowling coach . the 48-year-old former test paceman has served his country as part of the coaching team since 2011 . he said : ' i have had some time to reflect after the world cup and have come to the conclusion that the time is right to move on . allan donald has confirmed he is to step down as south africa bowling coach after four years in the role . ' it was always a big dream of mine to work in the south african cricket environment after my playing days , and i was incredibly honoured to be given the opportunity .' the last four years have been the best of my life , and being involved with the proteas in the 2015 cricket world cup was a goal i had set my sights on . 'cricket south africa chief executive haroon lorgat added : ' allan was and always will be a stalwart in south africa , having served his country with distinction both on and off the field . ' he brought great knowledge and international experience to the proteas set-up , and we wish him well in his future endeavours .

• **Reference**: allan donald served as south africa bowling coach since 2011 . donald said ' it was always a big dream ' to work in south african cricket . chief executive haroon lorgat said donald will ' always be a stalwart ' .

• **SHTE**: allan donald has confirmed he is to step down as south africa bowling coach . he said : ' i have had some time to reflect after the world cup and have come to the conclusion that the time is right to move on . ' it was always a big dream of mine to work in the south african cricket environment after my playing days , and i was incredibly honoured to be given the opportunity .

• **SHA-NN**: allan donald has confirmed he is to step down as south africa bowling coach . the 48-year-old former test paceman has served his country as part of the coaching team since 2011 . allan donald has confirmed he is to step down as south africa bowling coach after four years in the role .

Fig. 5.7 Summarization of a CNN/DailyMail test sample.

observed in SHA-NN where the related sentence extracted is different from the one extracted

by SHTE. If the 3rd sentence extracted by SHTE and the 2nd extracted by SHA-NN were selected, almost all the semantic of the reference summary would be covered. Also it is interesting to note that the generated summaries are much longer than the reference summaries, due to our systems are restricted to select full article sentences, however, the reference summaries could be composed by simplified sentences.

For the previous examples, in Figure 5.8 we show the attentions that each system assign to each sentence (the lighter the more relevant is a sentence). The left part of this figure refers to the SHTE and SHA-NN systems when they are applied on the NewsRoom example, whereas the right part refers to their application on the CNN/DailyMail example. SHTE$_H$ is the averaged matrix shown in Eq. 5.25 for the SHTE system, SHTE$_\alpha$ are the relevance scores assigned to each sentence by the SHTE system following the Eq. 5.26 and SHA-NN$_\alpha$ are the relevance scores assigned to each sentence by the SHA-NN system.

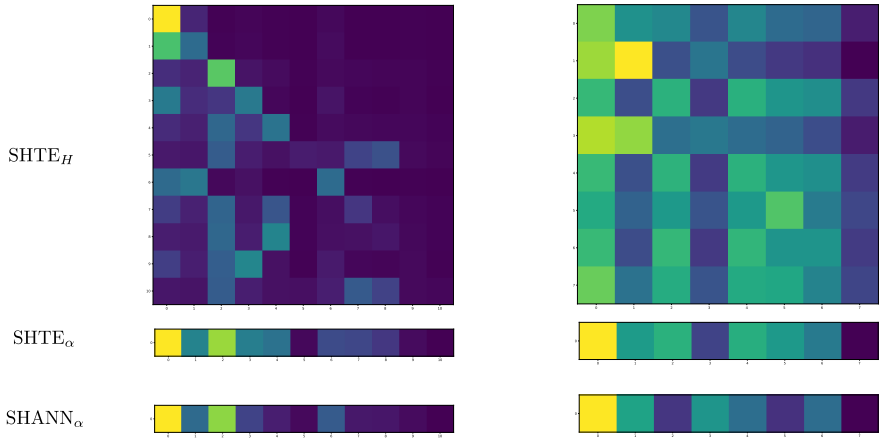

Fig. 5.8 Attentions for the NewsRoom and CNN/DailyMail test examples for both SHTE and SHA-NN. Clearer colors indicate a higher attention value.

It can be seen the bias towards the first sentences, where the attentions decrease from the first to the last sentences of the article. However, both systems are able to assign a lower score to some early sentences than to some late sentences. Also, in spite of the positional bias, as we will see in the following section, those systems are capable of generalizing on unseen documents where the sentences are more scattered [27]. The matrix H of the SHTE system, in the NewsRoom example, is almost a lower triangular matrix, suggesting that the dependencies among the sentences are given only backwards. This does not happen in the example of CNN/DailyMail where the attentions seem to compose patterns repeated at regular intervals within the same column.

5.3 Summarization of Spanish Talk Shows

Just as the volume of textual documents available on the web has grown dramatically in recent years, the same is true in the case of TV programs collections. The television channels make available to the public the programs of their own production, generating with it large collections of videos. For the audience who could not follow the broadcast of the live programs, it is interesting the possibility of accessing them. Also, the TV documentation services have current needs for quickly process large amounts of information, and this can be provided by means of adequate information retrieval searches and automatic summarization systems of contents. This is especially interesting for talk shows, which consist of several speakers giving opinions on various topics introduced by the program's presenter.

In this section, we focus on debate talk shows of "La Noche en 24 horas" (LN24), a program of the Spanish television (TVE), where the most important daily headlines from different newspapers are discussed. In these talk shows, several analysts interpret national and international news and discuss different topics like sports, economy, social events. Also, they hold interviews with political leaders or representatives of social, economic or cultural sectors, and they connect with reporters at the locations of relevant events. We address a text summarization problem, on the transcriptions of these talk shows, with the aim of summarizing the interventions of the speakers about a given topic. To do this, we studied the transferability of SHA-NN, trained for news article summarization, to the domain of talk shows in Spanish. Since we do not have a sufficiently large annotated corpus of TV programs to train our summarization system for the talk shows, we used the news articles domain as a proxy, due to, in our case, both domains consider very similar topics. However, it should be noted that they have also different characteristics. For example, due to the journalist style, the first paragraph condenses the main ideas and the relevant information underlying the article, while the debate talk shows are dialogs where the ideas and relevant information are more scattered.

The application of SHA-NN to automatic summarization of news articles requires the availability of adequate corpora consisting of a set of document-summary pairs. Although there are lots of works that developed appropriate corpora for English [274, 275, 289, 298, 312], this is not the same for other languages, such as Spanish. With the aim of building a corpus for Spanish, we followed a strategy similar to the proposed in [275], for the construction of the NewsRoom corpus. In [275], they take advantage of the summaries provided in the HTML metadata, written by authors and editors in the newsroom of a set of online publishers, in order to obtain reference summaries. The corpus was created through a web-scale crawling of over 100 million pages, from a set of online publishers, by gathering news about sports, entertainment, finances, and other kinds of publications along with their

reference summaries. To develop the system, we have built a corpus of Spanish news articles, the ES-NEWS corpus. It consists of a set of 277,675 (article, summary) pairs, extracted from 11 different Spanish newspapers. The ES-NEWS corpus contains articles and summaries of news, sports, politics, culture, and other topics. The use of this corpus in this work is two-fold. On the one hand, we evaluate our summarization system [28] with ES-NEWS in order to study the transferability of our system from English to Spanish. On the other hand, we use it to train the system that we apply to the summarization of Spanish talk shows.

In summary, we address the application of the SHA-NN system to summarize TV programs, in particular, Spanish TV talk shows of the LN24 program. First, in order to train our summarization system, the ES-NEWS corpus was built. Second, the SHA-NN system was evaluated on this text corpus. Third, a test corpus has been built with talk shows, the LN24-SUMM corpus. Finally, the summarization system trained with the ES-NEWS corpus has been applied to the LN24-SUMM test corpus. We did a preliminary evaluation of our summarization system on the transcribed speech of the LN24-SUMM test corpus. Despite the different characteristics between the two corpora, the results of transferability between domains are promising.

5.3.1 Corpora

The ES-NEWS corpus is composed by newspaper articles extracted from around 1 million URLs, which were collected during the last week of June 2018. To enforce the diversity of summarization styles, 11 websites of relevant newspapers of Spain have been used. These newspapers are: *Elconfidencial*, *EconomiaDigital*, *HuffingtonPost*, *ABC*, *FormulaTV*, *EldiarioCantabria*, *Publico*, *Vozpopuli*, *Rioja2*, *PeriodistaDigital* and *Eldiario*. We excluded newspapers that do not include highlights such as *ElMundo*, newspapers that only consider a single short highlight per article such as *ElPais*, and newspapers whose crawled content are not articles but web content (advertising, keywords, etc.) such as *EuropaPress*.

Once all the URLs were crawled, following [275], we have used the field *og:description* to extract the highlights that, concatenated, were considered as reference summaries. We made a preprocess in order to remove noise such as duplicated articles, empty summaries and articles, and non-journalistic articles. All the text was lowercased and tokenized by using the Spanish version of Stanford CoreNLP [324].

The corpus consists of 277,675 (article, summary) pairs, that are splitted in training, development and test partitions, following similar proportions to CNN/Dailymail corpus [290] (90%, 5.5% and 4.5% respectively). Thus, resulting in a training set of 249,919 pairs, a development set of 15,266 pairs and a test set of 12,490 pairs. Some ES-NEWS corpus statistics are shown in Table 5.7.

Table 5.7 ES-NEWS corpus statistics.

Dataset Size	277,675 pairs
Mean Article Length	813.8 words
Mean Summary Length	46.0 words
Mean Word Overlapping	28.6 words
Mean Extractive Fragment Coverage	0.67
Mean Extractive Fragment Density	7.27
Mean Compression Ratio	20:1
Articles Vocabulary Size	961,485 words
Summaries Vocabulary Size	167,822 words
Overlapping Vocabulary	157,863 words

To make a comparison between the ES-NEWS and the NewsRoom corpora, we have used the Extractive Fragment Coverage, Extractive Fragment Density and Compression Ratio measures. These metrics, proposed in [275], aim to measure the overlapping between summaries and articles to analyze the diversity of summarization styles. A formal definition of these metrics is given in Equations (5.33)–(5.35), where A is the sequence of words of the article, S is the sequence of words of the summary, $\mathscr{F}$ is the set of common fragments (common sequences of words) between A and S, computed using the greedy algorithm also proposed in [275], and $|\cdot|$ stands for the length, in terms of words, of the sequences.

$$\text{Coverage}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}} |f| \tag{5.33}$$

$$\text{Density}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}} |f|^2 \tag{5.34}$$

$$\text{Compression}(A,S) = \frac{|A|}{|S|} \tag{5.35}$$

The Extractive Fragment Coverage is computed as the sum of the lengths of all the common fragments between the article and its summary divided by the size of the summary. Thus, the greater its value, the more common fragments or larger common fragments have been found between the article and its summary. The Extractive Fragment Density is a measure similar to the Extractive Fragment Coverage but using the square of common fragment lengths. Therefore, with the same number of common words, summaries with longer common fragments obtain higher values than those with more but shorter common fragments.

All these measures were proposed with the aim of quantify the degree of abstractiveness, in terms of novel n-grams (lower coverage and density), and the degree of extractiveness, in terms of shared n-grams (higher coverage and density).

Figure 5.9 shows the density and coverage distributions along with the compression ratio for each newspaper in ES-NEWS corpus. Each box is a normalized bivariate density plot of Extractive Fragment Coverage (x-axis) and Extractive Fragment Density (y-axis) of a newspaper. Furthermore, the final distributions on full ES-NEWS is shown in the bottom-right box. As Figure 5.9 shows, the distribution of Extractive Fragment Density and Extractive Fragment Coverage of ES-NEWS is led by *ElDiario* (coverage between 0.6 and 0.8 with a high density), due to that newspaper brings the largest number of articles to the corpus. Despite this, generally for all the newspapers, the mean coverages and densities show that the introduction of novel n-grams in the summaries and the use of long extractive fragments is moderate, although both are higher than in NewsRoom corpus [275].

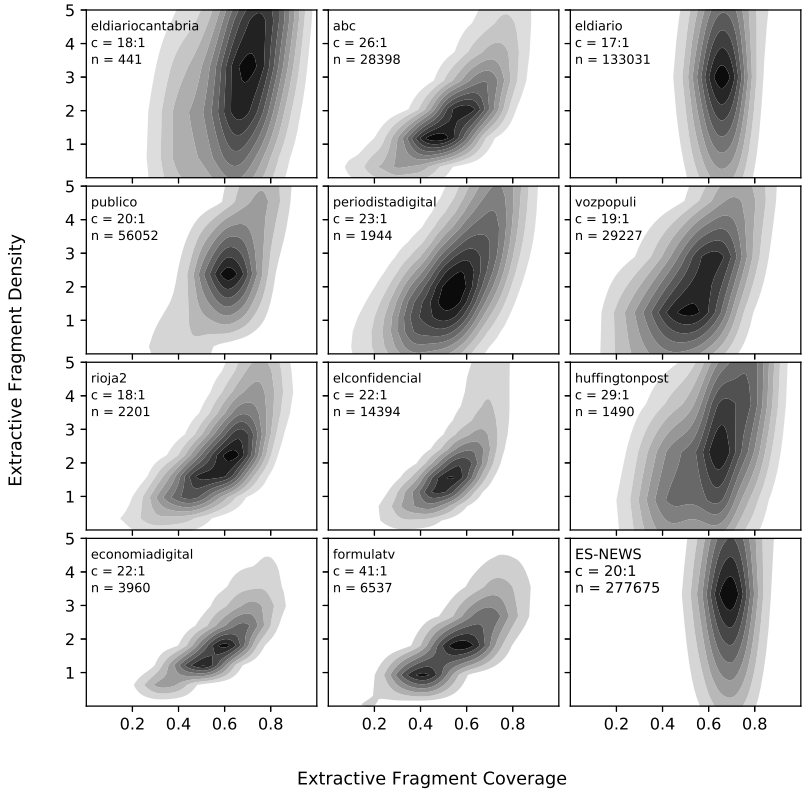

Fig. 5.9 Extractive Fragment Density and Extractive Fragment Coverage distributions on ES-NEWS corpus, where c is the Mean Compression Ratio and n is the number of article-summary pairs.

We have also built a corpus for summarization of the speakers' interventions about topics discussed in the Spanish talk show LN24, the LN24-SUMM corpus. It consists of a set of 30

(document, summary) pairs. Documents of this corpus are extracted from 5 talk shows of "La Noche en 24 horas" (approximately 10 hours of video), a program of the Spanish television (TVE). These talk shows were emitted in 2015/2016, and they contain some relevant topics such as the November 2015 Paris attacks and the elections to the Parliament of Catalonia in 2015. Documents have been obtained from the transcriptions of these TV programs, which have been manually segmented into pieces, first from the Twitter hashtags appearing in the program videos, and second, from the interventions of the different speakers. This segmentation was made with the aim of summarize the intervention of a speaker about a given topic (identified by the hashtag). Four experts generated the reference summaries. Figure 5.10 illustrates the process for extracting a document with all the interventions of a speaker given a given topic, and its corresponding manually written summary.

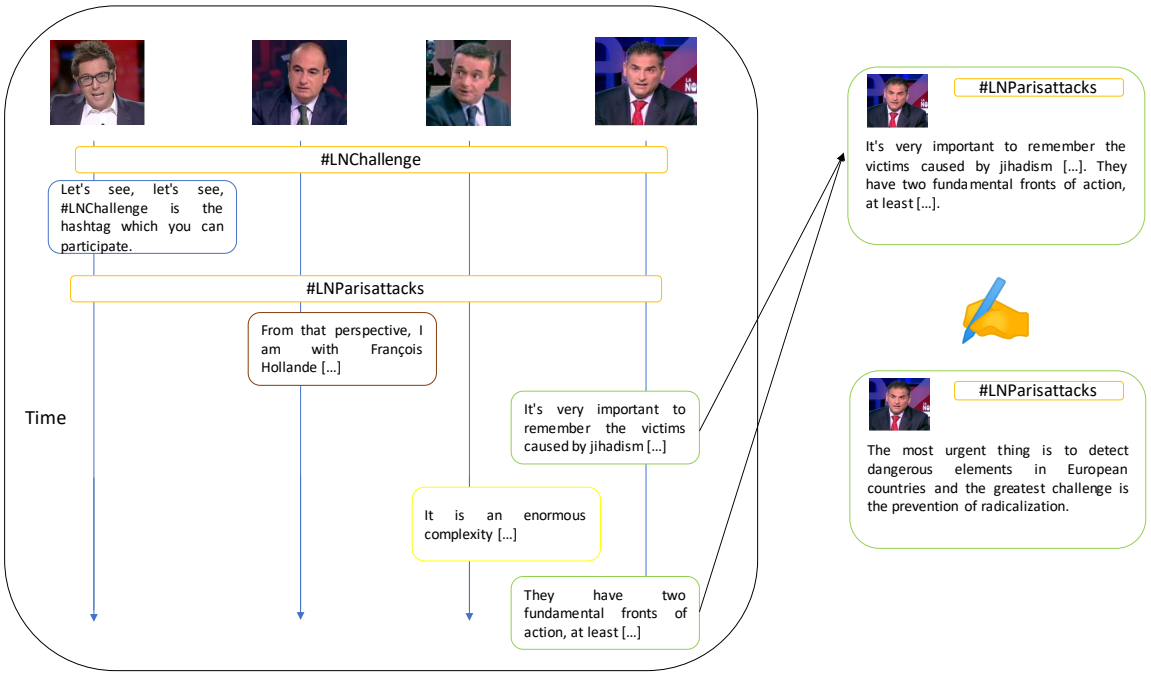

Fig. 5.10 Illustration of the process followed for building the LN24-SUMM corpus. The left box shows shortened fragments of a LN24 program emitted on 16/11/2015, where two topics are discussed. These two topics are #LNChallenge (intended to encourage the viewers' participation in Twitter, to solve challenges proposed by the presenter) and #LN24ParisAttacks. Four speakers participated in this fragment. In the example, all the interventions of the last speaker are gathered to compose the document we want to summarize. We also built reference summaries manually for these extracted documents.

A strategy based on paraphrasing the most representative sentences of each document was used for manually write the summaries. The generated summaries, although they are

abstractive (rewritten from the source document), have very high converage and density, as it can be seen in Table 5.8. In this table some additional LN24-SUMM statistics are also shown.

Table 5.8 LN24-SUMM corpus statistics.

Dataset Size	30 pairs
Mean Article Length	921.7 words
Mean Summary Length	108.6 words
Mean Word Overlapping	64.8 words
Mean Extractive Fragment Coverage	0.90
Mean Extractive Fragment Density	19.31
Mean Compression Ratio	8:1
Articles Vocabulary Size	3969 words
Summaries Vocabulary Size	1103 words
Overlapping Vocabulary	1069 words

It is interesting to see that the Mean Compression Ratio is lower in LN24-SUMM than in ES-NEWS corpus. Also, it can be seen that LN24-SUMM corpus has a very high mean Extractive Fragment Density and Coverage in comparison to ES-NEWS corpus. The differences could be because the newspaper summaries are written by many different journalists who are qualified in compressing information and in generating more diverse kind of summaries. In addition, extracting relevant information from newspaper articles is simpler than from speaker interventions in talk shows. In the case of newspaper articles there are some sentences, that mainly appear at the beginning of the article, that contain the main ideas or information of the article. However, the talk shows are dialogs where the relevant information is more scattered in the speaker turns and exhibits spontaneous speech phenomena. Therefore, they are more difficult to summarize. Additionally, the LN24-SUMM documents, that is, the transcribed and manually segmented talk shows, are very heterogeneous. Two examples to see the differences between both corpora are shown in Figure 5.11. In this figure it is possible to see that the LN24-SUMM summaries are composed by more scattered sentences than the summaries of ES-NEWS.

5.3.2 Evaluation

We carried out two different experiments. First, we trained and evaluated the SHA-NN system with ES-NEWS corpus, and second, we evaluated the trained system with the LN24-SUMM corpus.

Reference Summary (ES-NEWS): One of the best scientific minds in the world suffered from amyotrophic lateral sclerosis and lived 53 years longer than the doctors diagnosed him **(1/13)**. "Their courage and persistence with their brilliance and their humor inspired people all over the world," their children say in a statement **(4/13)**.

Reference Summary (LN24-SUMM): More than 72 h have passed since the attack on Friday **(3/28)**. The city suffered a heinous attack on several fronts, leisure centers, in football and in a concert hall **(11/28)**. Terrorists have attacked our way of life, even children are accompanied by security forces **(16/28)**. Francois Hollande is going to meet in Paris with John Kerry **(18/28)**. The five terrorists have been identified, four were French and one would have a Syrian passport **(23/28)**.

Fig. 5.11 Examples of summaries from ES-NEWS and LN24-SUMM corpora translated from Spanish. The position of the most related sentence in the document, for each sentence of the summary, is highlighted in bold at the end of the sentences.

In order to evaluate our proposal, we performed a experimental comparison with 5 extractive unsupervised summarization systems. Concretely, they are Lead [290], LexRank [282], TextRank [283], Latent Semantic Analysis (LSA) [325] and SumBasic [326]. A short description of each system is shown below.

- *Lead*: a very popular and robust strategy to generate snippets and summaries of article newspapers that consists in extracting the first k sentences of the documents. This strategy is typically used as a baseline in the automatic summarization of newspaper articles, since in the writing style of this type of documents the most relevant information is usually condensed in the first paragraphs to grab the attention of the reader.

- *LexRank*: an unsupervised, graph-based summary generation system inspired by both PageRank and HITS. It is based on the idea that the relevance of a sentence depends on its similarity with the rest of the sentences in the text. The nodes of the graph are the document sentences and the edges measure the similarity between two sentences using a idf based cosine distance. Two sentences are connected if the cosine similarity between them is greater than a certain threshold. The summary is made with the most salient sentences. If a sentence is similar to many others, then it must be salient in the document.

- *TextRank*: like LexRank, is an unsupervised graph-based system inspired by PageRank. It uses a variation of PageRank to extract the most salient sentences of the document. Its most significant difference from LexRank is the way in which the weights of the edges are calculated. In this case, the edges measure the similarity between the different nodes based on the number of common words in the sentences.

- *LSA*: a method based on Singular Value Decomposition, where a word-sentence matrix is decomposed in three new matrices. One of these matrices represents the association of underlying topics to sentences. This matrix is used to select the more salient sentences.

- *SumBasic*: it exploits frequency related properties of the words to compose summaries, arguing that high frequency words in the documents are very likely to appear in the human generated summaries. It is a greedy search approximation where, first the probability distributions of the words are computed, second by using these probabilities a weight is assigned to each document sentence and later, the best scoring sentence is selected to fill the summary until the desired summary length has been reached.

In all the experimentation, we used the implementation of these systems provided by the Python *sumy* library [14] using the default configuration. All these systems extract 3 sentences in order to compose the summary.

The performance of the systems was evaluated by using variants of the ROUGE. Concretely, ROUGE-N with unigrams and bigrams (ROUGE-1 and ROUGE-2) and ROUGE-L. A formal definition of the ROUGE metric is shown in Eqs. 5.27 to 5.32. Furthermore, the compression ratio of the generated summaries (Compression) was also analyzed. In order to compute the confidence intervals, we used the Bootstrap Confidence Intervals [193] approach. First, from the set of hypotheses provided by the system that we want to evaluate, we generated up to 1000 resamples by sampling with replacement from this original set of hypotheses. Each resample had the same size of the original set. Next, the value of the evaluation measure was computed for each of the resamples. Finally, we computed the 95% confidence interval using the bootstrap distribution.

Table 5.9 shows the results of our system compared to other summarization systems using the test set of ES-NEWS corpus. All the results in this experimentation are statistically significant.

[14] `https://github.com/miso-belica/sumy`

Table 5.9 Results on ES-NEWS corpus with respect to the ground truth (full length ROUGE F_1).

	ROUGE-1	ROUGE-2	ROUGE-L	Compression
SHA-NN	30.1 ± 0.19	14.6 ± 0.20	25.6 ± 0.19	6.5
Lead	**32.8** ± 0.21	**16.2** ± 0.24	**27.5** ± 0.21	8.7
LexRank	28.4 ± 0.18	12.4 ± 0.18	23.6 ± 0.16	6.3
TextRank	24.0 ± 0.18	10.7 ± 0.17	20.2 ± 0.16	4.7
LSA	27.1 ± 0.16	8.4 ± 0.17	21.7 ± 0.15	8.5
SumBasic	29.9 ± 0.19	10.1 ± 0.19	24.5 ± 0.18	10.7

As it can be seen in Table 5.9, the results of all the systems on the ES-NEWS corpus are lower, in terms of all the ROUGE variants, than those obtained on the English CNN/DailyMail corpus. This can be seen for the Lead and SHA-NN systems, that were also applied on CNN/DailyMail, and it seems to suggest a higher difficulty of ES-NEWS compared to CNN/DailyMail. We hypothesized that this could be due to two aspects: a higher compression ratio (20:1 in ES-NEWS and 14:1 in CNN/DailyMail) and the variety of strategies used by the journalists to write the reference summaries (see Figure 5.9 and [275]). Regarding SHA-NN system, the results show a good transferability between languages, as we hypothesized, due to it maintains a similar behavior than the Lead heuristic also in this case.

Using the summarization system trained in the above experimentation, we evaluated it with the LN24-SUMM test corpus, which contains 30 document-summary pairs (a small test set compared to the training set). Table 5.10 shows the results of the SHA-NN system compared to other summarization systems.

Table 5.10 Results on LN24-SUMM corpus with respect to the ground truth (full length ROUGE F_1).

	ROUGE-1	ROUGE-2	ROUGE-L	Compression
SHA-NN	**46.0** ± 4.38	**29.0** ± 5.94	**42.2** ± 4.46	7.0
Lead	33.2 ± 4.86	17.4 ± 5.86	29.9 ± 5.17	13.4
LexRank	39.7 ± 3.64	20.6 ± 4.73	35.2 ± 4.03	5.9
TextRank	43.3 ± 5.76	27.0 ± 7.65	39.7 ± 6.10	3.9
LSA	36.1 ± 4.60	15.8 ± 5.86	31.2 ± 5.10	9.4
SumBasic	31.8 ± 5.21	14.4 ± 5.45	28.7 ± 4.94	24.7

This table shows that the results of our summarization system are better than those of the other systems at all levels of ROUGE, although it should be considered that the small size of the test set does not allow to obtain statistically significant results. It should be noted that when working with the ES-NEWS corpus, the Lead heuristic, which consists of extracting the first 3 sentences of the article as a summary, outperforms the rest of the systems, including ours, as Table 5.9 shows. However, this system performed worse on LN24-SUMM corpus. This is due to the fact that in the LN24-SUMM corpus, unlike in the ES-NEWS corpus, the most relevant sentences are scattered across the document, and do not correspond to the first sentences. Also, it is interesting that, although SHA-NN was trained under the positional bias to the first sentences (ES-NEWS), it is capable of generalizing when the relevant sentences are more scattered in the document (LN24-SUMM). Also, regarding to the transferability between domains, it is possible to see that all the results, in terms of ROUGE, obtained on the LN24-SUMM corpus are higher than those obtained on the ES-NEWS corpus. Possibly, that is because the reference summaries of the LN24-SUMM corpus have a very high density (i.e., they are composed by long extractive fragments of the transcribed talk shows) and a very low compression ratio in comparison to the ES-NEWS corpus, as it can be seen in Tables 5.7 and 5.8.

Furthermore, it is interesting to see in Table 5.10 that in general, when the compression ratio of the generated summaries increases, the results in terms of ROUGE decrease. Our system provides the best trade-off ROUGE/Compression among all systems. Moreover, although TextRank obtains the most similar results with respect to SHA-NN, it suffers from a very low compression ratio due to it tends to extract the longest sentences.

Part of the research shown in this chapter was published in three papers by the author:

- *José-Ángel González, Segarra Encarna, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Siamese hierarchical attention networks for extractive summarization.* Journal of Intelligent and Fuzzy Systems*, 36(5):4599–4607, 2019*
- *José Ángel González, Encarna Segarra, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Extractive summarization using siamese hierarchical transformer encoders.* Journal of Intelligent & Fuzzy Systems*, 39:2409–2419, 2020. 2*
- *José-Ángel González, Lluís-Felip Hurtado, Encarna Segarra, Fernando Garcia-Granada, and Emilio Sanchis. Summarization of Spanish Talk Shows with Siamese Hierarchical Attention Networks.* Applied Sciences*, 9(18), 2019*

Chapter 6

Conclusions and Future Work

In this chapter, we summarize the work performed in this thesis. We present both the conclusions derived from each specific work, and holistic conclusions to discuss all the work as a whole. Finally, future lines of works and a discussion of the extensions, in which we are working currently, are presented. First of all, we highlight the following conclusions:

Dominance of Transformers models: in most of our experimentations, the Transformer encoders have dominated recurrent, convolutional, and collapsing approaches, even when they are trained from-scratch, applied on top of non-contextual word representations pretrained for the target domain and language, on downstream tasks. This strategy, allowed us to take profit from the powerful backbone neural architecture of modern language models, dispensing with an expensive pretraining, in order to contextualize pretrained word embeddings. We extensively evaluated and analyzed this approach on document-level sentiment analysis and irony detection on tweets written in English and several Spanish variants. The evaluation shows the adequacy of the proposal, which obtained very promising results in the TASS, IroSVA, and SemEval competitions, being always the first or second-ranked approach in these competitions. However, the difference between Transformer encoders and recurrent models for the automatic summarization corpora and the approaches we worked is not such as in the text classification case. In this case, the performance of both models is practically identical, showing that Transformer does not bring improved ROUGE compared with recurrent approaches [327, 328]. However, attending to the current state of the art in automatic summarization, where the Transformers dominate the leaderboards, this seems to indicate that the key for their better performance is the knowledge encoding after large-scale pretraining.

Interpretability of the attention mechanisms: in order to understand the behavior of deep models, the attention weights have been extensively studied for this purpose. Attention conveniently gives us one weight per element in the sequence, ideally denoting the relevance of that element in a specific task. Although there is a lot of controversy about the adequacy of the attention mechanisms for explainability purposes with mixed evidence on whether it can be used to this aim [73–77]. Despite the discussions about the best-ever technique for interpretability, in this thesis, we extensively explored attention mechanisms in order to observe expected behavior patterns of the models to address several tasks. For sentiment analysis, we studied how the attention heads of the Transformer encoder are specialized on detecting the polarity of the words and the presence of polarity modifiers. Our analysis found that this specialization actually occurs where the greatest burden of the word polarity detection falls on two heads both for positive and negative words. Furthermore, we found some attention heads that react more to polarity modifiers than to other words. For irony detection, we hypothesized the highest influence of some attention heads in the classification of ironic content and we proposed several analyses and algorithms in order to determine that attention heads. From those “ironic" heads, we studied and found several features captured by the models to detect the ironic class such as word polarities, word relationships, and individual words indicating irony. Finally, our attentional extractive framework for automatic summarization is highly dependent on the interpretability of the attention mechanisms, since it is based on selecting sentences following the weights assigned by these mechanisms, that ideally denote the relevance of each sentence in the proxy task we proposed for distinguishing correct summaries for documents, similarly to multiple-instance learning.

Evaluation metrics as loss functions: since the loss function is used by the back-propagation algorithm to guide the parameter estimation process in neural models, a straightforward approach to reduce the mismatch between the evaluation and the training objectives is to integrate differentiable approximations of the evaluation criteria as loss functions. For text classification, evaluation metrics that penalize the bias of the models towards the most frequent classes are typically used, which poses a problem when the loss functions used to train neural networks on highly imbalanced corpora do not take into account this imbalance. This is very common in practical situations where negative log-likelihood is used for training the models without additional considerations. In this thesis, we proposed the use of loss functions based on evaluation metrics to consider the imbalance among the classes and to address the mismatch between negative log-likelihood and evaluation metrics like the macro-averaged F_1. We evaluated

this proposal on a multi-label emotion classification task, where the combinations of emotions can potentially generate a large space of imbalanced classes, that show high effectiveness. We found that, in almost all cases, the models trained with the proposed loss functions obtained the best results in terms of the evaluation metric they approximate, reducing the bias of the model towards the most frequent classes if the differentiable approximation of the evaluation metric considers the class imbalance.

Pretrained language models: as stated before, the key for the better performance of the Transformers seems to be the knowledge encoding after large-scale pretraining. Especially, for the English language, the competitive behavior of pretrained BERT-based models has encouraged the scientific community to use BERT ubiquitously in a broad range of tasks. This is not the case of the Spanish language, where the lack of large pretrained Transformer language models has conducted the community to use multilingual pretrained versions such as Multilingual BERT. However, the performance of these multilingual approaches is even worse than other approaches based on non-contextual representations pretrained on the target language and domain. Although it is expensive to train Transformer models from-scratch, on large corpora for each domain and language, there are empirical clues that indicate a better behavior of these models. For this reason, we proposed TWilBERT, a specialization of BERT both for the Spanish language and the Twitter domain, that outperformed Multilingual BERT on 14 different datasets of text classification tasks with Spanish tweets, such as irony detection, sentiment analysis, emotion detection, hate speech detection, stance detection, and topic detection. The proposed models seem to capture better the topic and inter-sentence coherence between tweets, they are better language models on the Twitter domain, and their attention heads shown lower redundancy, compared to Multilingual BERT.

Twitter conversations to learn coherence: the benefits of including inter-sentence coherence for pretraining Transformer language models have been a controversial topic in the literature. BERT was trained with Next Sentence Prediction in order to incorporate inter-sentence coherence, but, progressively the use of this signal has declined due to negative empirical evidence towards it. However, we considered, like the authors of ALBERT who proposed the Sentence Order Prediction signal [82], that inter-sentence modeling is an important aspect for language understanding. For this reason, under the scope of our work with Twitter, we argue the importance of this coherence modeling and we proposed the Reply Order Prediction signal to learn coherence between pair of tweets by focusing on the sequentiality among a given tweet and the replies to this

tweet in Twitter conversations. This signal has shown to be one of the key factors of the success of TWilBERT, improving the average results on 14 text classification corpora, and the capability for topic and inter-sentence coherence between pairs of tweets, in comparison to Multilingual BERT.

Boosting the research on the Spanish language: the NLP research for the Spanish language is, by far, not as extensive as for the English language, and it is typically limited to following in the wake of advances in the English language. For this reason, we considered the Spanish language as the central language in our work, in order to contribute and motivate the study of computational approaches for addressing NLP problems in this widely spread and understudied language. Although there are many text classification corpora for Spanish, mainly for social media text analytics such as sentiment analysis or hate speech detection, there is a need to explore and generate resources for more complex tasks that require a greater understanding of the language by the models such as automatic summarization, question answering or commonsense reasoning. Nowadays, there are not alternatives to training models for working on these tasks with the Spanish language, which have driven us, under the scope of this thesis, to build Spanish corpora for automatic summarization, following previous works on the English language. With this conclusion, we try to appeal to the research community in order to promote the NLP research on the Spanish language. To contribute to this aim, most of our future works will be intended for the Spanish language.

Attentional extractive summarization framework: we proposed an attentional framework for extractive summarization, based on siamese hierarchical networks with attention mechanisms. It allows to developing models that dispense with extractive oracles and Reinforcement Learning techniques based on ROUGE to fit the task into a sequential binary classification problem. Under this framework, we proposed two different models, based on different attentional encoders: Siamese Hierarchical Attention Networks and Siamese Hierarchical Transformer Encoders. We have performed an extensive evaluation and several analyses of our systems both for the CNN/DailyMail and for the NewsRoom corpora. The obtained results are very promising, in comparison to systems that were proposed in the literature at the same time, previously to the explosion of large pretrained language models, and they suggest that there is still room for the improvement of our attentional framework. In fact, a novel state-of-the-art paradigm for extractive summarization is based on text matching (MatchSum [295]), that is highly related to our attentional extractive summarization framework, in the sense that both compute document, reference, and distractor representations,

and they leverage a siamese approach to represent the references closest to the source documents.

Domain transferability: we evaluated the domain transferability of Siamese Hierarchical Attention Neural Networks (SHA-NN) for extractive summarization, when they are trained on a Spanish news articles dataset (ES-NEWS), but evaluated on the summarization of the speakers' interventions about topics discussed in the Spanish talk show LN24, (LN24-SUMM). Despite the main difference between ES-NEWS and LN24-SUMM (the news articles are written language and the speaker's interventions are transcribed spontaneous speech) both of them discuss events of public interest about similar topics. Furthermore, most of the content of this kind of TV talk shows is completely based on articles published in the media press, so, there is a close relationship between both domains, in spite of differences in their forms. The results obtained in the transferability experimentation are very promising, thus posing the alternative of pretraining summarization models on source domains closely related to the target domain. In addition, although the news articles contain a positional bias towards the first sentences, and SHA-NN was exposed to that bias during training, it seems capable of generalizing when the relevant sentences are more scattered in the documents.

Replicability: we considered that the replicability of the published systems is a key factor to be taken into account. To this aim, we release the source code of all the systems of this thesis. The models for sentiment analysis and irony detection are released as a transferable result through the Office for the Promotion of Research, Innovation and Technology Transfer (UPV), under the software SENTAT, ES-IRONIC and EN-IRONIC. The works with our attentional extractive summarization framework are available on three Github repositories: `AES`, `SHA-NN` and `SHTE`. The source code of the differentiable evaluation metrics for text classification can be accessed from `DEVM-TC`. Finally, we provided a framework for training, evaluating, and fine-tuning BERT models, that also implements several improvements on Transformer models recently published in the literature. With this framework, we pretrained the TWilBERT models, whose weights are publicly available together with the source code of the framework in `TWilBERT`.

From the previous conclusions, we consider several future research lines that fall under the umbrella of the research projects currently developed in our research group:

Sentiment Analysis: we only addressed an oversimplification of the sentiment analysis problem, where the overall polarity has to be determined. However, there remain

several open research directions to be extensively studied, such as understanding the motive and cause of sentiment, sentiment reasoning, or sentiment-aware natural language generation. So, we believe that we should strive to move past simple classification as the benchmark of progress, and instead direct our efforts towards learning tangible sentiment understanding. Taking a step in this direction would include: developing large-scale high-quality resources, analyzing, customizing, and training modern architectures in the context of fine-grained sentiment analysis, along with the exploration of parallel new directions, such as multimodal learning, sentiment reasoning, sentiment-aware natural language generation, and its relationships with unexplored figurative language like the hyperbole or the metaphor. Also, from the point of view of our work, Aspect Based Sentiment Analysis (ABSA) is very interesting. In future works, we also plan to develop ABSA approaches and resources, in order to determine positive and negative aspects of influential actors, events, and TV programs discussed in social media platforms and TV talk shows. Regarding the methodology, we will work with Spanish and Catalan as main languages, we will continue modeling our approaches by following the wake of the modern pretrained language models, and we aim to continue exploiting the structure of Twitter, both in terms of historical tweets of the users (including threads of daisy-chained tweets) and in terms of Twitter conversations that expose interactions among the users. We will also explore the chaining of tweets among multiple users, resembling conversations, in order to acquire additional context information, for example, for performing distant supervision.

Automatic Summarization: we proposed an attentional framework for extractive summarization, however, the dominance of pretrained large Transformers on the summarization problem leads us to use work with these approaches for our purposes. Thus, we plan to develop abstractive summarization systems mainly based on the ideas of BART [95] and Pegasus [96]. In this regard, we are currently working on self-supervised pretraining Transformers for a set of denoising objectives such as Gap Sentence Generation, Token Infilling, and Sentence permutation, on large datasets of raw text in Spanish and in Catalan, in order to finetune them on the summarization of news articles from Spanish and Catalan newspapers. To this aim, we have already built Spanish and Catalan corpora both for pretraining and for summarization, that we plan, as one of the main objectives, to make it publicly available in next months (DACSA). Also, there are some interesting research lines directly derived from our attentional framework for extractive summarization. The first one is to fully integrate the developed systems into the TV talk shows analytics module, by modifying them in order to work with the output of the speech recognition models, instead of human-written

transcriptions. Furthermore, as the proposed models were trained from-scratch on top of non-contextual representations of words, it is interesting to explore the use of pretrained encoders, and also to use them in a two-stage process for first, extracting from the attention mechanisms several potential summary candidates, and second, score them following text matching approaches (we are currently working on this AES). We are also interested in the conjunction of the two broad fields discussed in this thesis i.e. opinion summarization. In this regard, it is very interesting to summarize the opinions of TV viewers about the topics discussed in TV talk shows, or about the TV programs themselves. Concerning the evaluation metrics, since ROUGE is based only on form overlapping and it could rank high some abstractive summaries that cannot be used in practice e.g., unfaithful summaries, we are also interested in proposing alternative automatic evaluation metrics intended to address this issue. Finally, we are currently working on a project of entity semantic aggregation, intended to improve the *abstraction* capabilities of abstractive summarization systems, that are typically focused on paraphrasing the source documents, instead of performing true *abstraction* actions such as semantic generalization, in the sense of deriving general concepts from specific instances.

Appendices

A.1 Evaluation Metrics

The supervised classification problem can be defined as the problem of learning a function $f : \mathscr{X} \rightarrow \mathscr{Y}$ from a set of labeled samples D, where

- $\mathbb{C}$ is a finite set of classes $\mathbb{C} = \{c_1, \cdots, c_{|\mathbb{C}|}\}$, and $|\mathbb{C}| > 1$
- $\mathscr{X}$ is an input space
- $\mathscr{P}(\mathbb{C})$ is the label power set of $\mathbb{C}$
- $\mathscr{Y}$ is the set of considered labels, $\mathscr{Y} \subseteq \mathscr{P}(\mathbb{C})$, and
- $D = \{(x_1, \gamma_1), \cdots, (x_n, \gamma_n)\}$ is a data set of samples, where $x_i \in \mathscr{X}$ and $\gamma_i \in \mathscr{Y}$

When $|\mathbb{C}| > 2$, the problem is called multi-class classification and when a sample can be assigned to more than one class (i.e. $\mathbb{C} \subset \mathscr{Y}$) the problem is called multi-label classification.

In order to automatically evaluate the performance of the classifier we assume that a labeled test set is available. Let (X, Y) be a test set consisting on m samples (x_i, γ_i), $1 \leq i \leq m$, $(x_i \in \mathscr{X}, \gamma_i \in \mathscr{Y})$, with a set of classes $\mathbb{C}$, $|\mathbb{C}| = n$. Let $O = \{\theta_i : 1 \leq i \leq m\}$ be the set of predictions of the classifier on the test set, where $\theta_i \in \mathscr{Y}$ is the set of classes assigned to the sample x_i by the classifier.

One of the most used metrics to evaluate classifiers is Accuracy. In mono-label classification tasks, i.e. only one class per sample, Accuracy is defined as the percentage of correctly classified samples. For multi-label classification tasks, it is necessary to introduce an extension of the Accuracy metric. Eq. (EA.1) shows the formulation of multi-label **or Jaccard Accuracy** (Acc).

$$Acc = \frac{1}{m} \sum_{i=1}^{m} \frac{|\gamma_i \cap \theta_i|}{|\gamma_i \cup \theta_i|} \tag{EA.1}$$

where the numerator represents the number of correctly predicted classes for sample i, normalized by the size of the union of the predicted and correct class sets for sample i. Note that this metric is equivalent to Jaccard index per sample, averaged across all the samples.

The other metrics we study in this work are Precision, Recall, and F_1 measure. Moreover, there are two points of view to calculate a value for these metrics considering the complete test set, micro-averaging and macro-averaging.

Following the micro-averaging approach, we can define the micro-Precision, or just **Precision** (P), as the fraction of all classes generated by the classifier that have been correctly predicted; and the micro-Recall, or just **Recall** (R), as the fraction of all correct classes that have been correctly predicted by the classifier. Both metrics are formally defined in Eqs. (EA.2) and (EA.3).

$$P = \frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\theta_i|} \tag{EA.2}$$

$$R = \frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\gamma_i|} \tag{EA.3}$$

Micro-F_1 (m-F_1) is a particular case of micro-F_β measure where $\beta = 1$, that is, the harmonic mean of Precision and Recall. Eq. (EA.4) shows the formulation of micro-F_1.

$$\text{m-F}_1 = \frac{2 \cdot P \cdot R}{P + R} = 2 \cdot \frac{\frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\theta_i|} \cdot \frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\gamma_i|}}{\frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\theta_i|} + \frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\gamma_i|}} = 2 \cdot \frac{\sum_{i=1}^{m} |\gamma_i \cap \theta_i|}{\sum_{i=1}^{m} |\theta_i| + \sum_{i=1}^{m} |\gamma_i|} \tag{EA.4}$$

Additionally, we can compute the Precision, Recall and F_1 per class. Eqs. (EA.5), (EA.6) and (EA.7) show the definition of these metrics for a specific class c. Note that, we use the Iverson bracket notation $[c \in \gamma_i \cap \theta_i]$ which has the value 1 if c bellows to $\gamma_i \cap \theta_i$ and 0 otherwise.

$$P_c = \frac{\sum_{i=1}^{m} [c \in \gamma_i \cap \theta_i]}{\sum_{i=1}^{m} [c \in \theta_i]} \tag{EA.5}$$

$$R_c = \frac{\sum_{i=1}^{m} [c \in \gamma_i \cap \theta_i]}{\sum_{i=1}^{m} [c \in \gamma_i]} \tag{EA.6}$$

$$F_{1,c} = 2 \cdot \frac{\sum_{i=1}^{m} [c \in \gamma_i \cap \theta_i]}{\sum_{i=1}^{m} [c \in \theta_i] + \sum_{i=1}^{m} [c \in \gamma_i]} \tag{EA.7}$$

Following the macro-averaging approach, we can compute the **macro-$F1$** (M-F_1) as the arithmetic mean of F_1 per class. Equation EA.8 shows the definition of macro-F_1. In this case, all classes equally contribute to the global measure regardless of the number of samples. It is convenient to highlight that also, macro-averaging versions of Precision and Recall can be computed, by means of averaging P_c and R_c respectively, for all the classes $c \in \mathbb{C}$.

$$\text{M-F}_1 = \frac{2}{|\mathbb{C}|} \cdot \sum_{c \in \mathbb{C}} \frac{\sum_{i=1}^{m} [c \in \gamma_i \cap \theta_i]}{\sum_{i=1}^{m} [c \in \theta_i] + \sum_{i=1}^{m} [c \in \gamma_i]} \tag{EA.8}$$

A.2 Corpora Statistics

Table A.1 Statistics of all the corpora used in this thesis. For each partition, |ℂ| refers the number of classes if applicable, |S| to the number of samples, |T| refers to the number of tokens and |V| is the vocabulary size. * indicates a multilabel class set, on |ℂ| independent classes, that can potentially generate $2^{|\mathbb{C}|}$ combinations.

					Training			Development			Test		
Corpus	Task	Language	Domain	|ℂ|	|S|	|T|	|V|	|S|	|T|	|V|	|S|	|T|	|V|
TASS 2019	Sentiment Analysis	Spanish (Spain)	Twitter	4	1.1k	17.5k	6.4k	581	9.3k	3.7k	1.7k	25.3k	9.0k
		Costa Rican	Twitter	4	777	11.8k	4.2k	390	6.5k	2.7k	1.2k	14.5k	5.2k
		Peruvian	Twitter	4	966	15.5k	5.5k	498	8.4k	3.4k	1.5k	22.4k	7.5k
		Uruguayan	Twitter	4	943	14.4k	5.3k	486	8.1k	3.2k	1.4k	18.7k	6.5k
		Mexican	Twitter	4	989	16.8k	5.7k	510	8.6k	3.4k	1.5k	25.4k	7.6k
SemEval 2018 Task 1	Emotion Detection	Spanish	Twitter	11*	3.6k	48.1k	14.3k	679	8.9k	3.7k	2.9k	38.2k	11.9k
		English	Twitter	11*	6.8k	110.0k	24.4k	886	14.1k	5.9k	3.3k	51.8k	15.9k
IroSVA	Irony Detection	Spanish (Spain)	Twitter	2	2.1k	49.5k	13.1k	300	7.2k	3.0k	600	14.7k	5.3k
		Mexican	Twitter	2	2.1k	42.1k	11.8k	300	6.1k	2.6k	600	12.5k	4.7k
		Cuban	Twitter	2	2.1k	59.1k	13.7k	300	8.2k	3.1k	600	17.0k	5.6k
COSET	Topic Classification	Spanish	Twitter	5	2.2k	43.2k	13.8k	250	4.7k	2.3k	624	11.7k	4.8k
StanceCat	Stance Detection	Spanish	Twitter	3	3.5k	55.0k	13.2k	862	13.4k	4.7k	1.1k	17.8k	5.3k
MultiStanceCat	Stance Detection	Spanish	Twitter	3	3.6k	59.1k	14.8k	888	15.0k	5.2k	1.1k	19.2k	5.8k
HateEval	Hate Speech Detection	Spanish	Twitter	2	4.5k	95.0k	26.2k	500	10.8k	4.5k	1.6k	35.0k	10.9k
CNN/DailyMail	Summarization	English	Newspapers	N/A	287.2k	257.0M	564.9k	13.4k	11.7M	104.9k	11.5k	10.13M	100.1k
NewsRoom	Summarization	English	Newspapers	N/A	995.0k	780.0M	2.8M	108.8k	86.9M	674.9k	108.9k	86.7M	667.0k
ES-NEWS	Summarization	Spanish	Newspapers	N/A	250.0k	215.0M	917.4k	15.3k	13.2M	201.5k	12.5k	10.7M	181.4k
LN24-Summ	Summarization	Spanish	TV talk shows	N/A	N/A	N/A	N/A	N/A	N/A	N/A	30	30.9k	4.0k

References

[1] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.

[2] Manuel Carlos Díaz-Galiano, Manuel García Vega, Edgar Casasola, Luis Chiruzzo, Miguel Ángel García Cumbreras, Eugenio Martínez Cámara, Daniela Moctezuma, Arturo Montejo-Ráez, Marco Antonio Sobrevilla Cabezudo, Eric Sadit Tellez, Mario Graff, and Sabino Miranda-Jiménez. Overview of TASS 2019: One more further for the global spanish sentiment analysis corpus. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 550–560. CEUR-WS.org, 2019.

[3] Noam Chomsky. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, 1965.

[4] B. F. Skinner. *About behaviorism.* About behaviorism. Alfred A. Knopf, Oxford, England, 1974.

[5] Jean Piaget. *The theory of stages in cognitive development.*, pages ix, 283–ix, 283. Measurement and Piaget. McGraw-Hill, New York, NY, US, 1971.

[6] L.S. Vygotskii, E. Hanfmann, G. Vakar, and A. Kozulin. *Thought and Language*. The MIT Press. MIT Press, 2012.

[7] J.S. Bruner and R. Watson. *Child's Talk: Learning to Use Language*. Oxford University Press, 1983.

[8] D. Goleman. *Emotional Intelligence*. A Bantam book. Bantam Books, 2006.

[9] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA, 2017. Curran Associates Inc.

[12] José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Self-attention for twitter sentiment analysis in spanish. *Journal of Intelligent & Fuzzy Systems*, 39:2165–2175, 2020.

[13] Rosario Sanchis-Font, Maria Jose Castro-Bleda, José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. Cross-domain polarity models to evaluate user experience in e-learning. *Neural Processing Letters*, May 2020.

[14] Rosario Sanchis-Font, Maria Jose Castro-Bleda, and José-Ángel González. Applying sentiment analysis with cross-domain models to evaluate user experience in virtual learning environments. In Ignacio Rojas, Gonzalo Joya, and Andreu Catala, editors, *Advances in Computational Intelligence*, pages 609–620, Cham, 2019. Springer International Publishing.

[15] José-Ángel González, José Arias Moncho, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at TASS 2020: TWilBERT for Sentiment Analysis and Emotion Detection in Spanish Tweets. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 179–186. CEUR-WS.org, 2020.

[16] José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at TASS 2019: Transformer Encoders for Twitter Sentiment Analysis in Spanish. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, pages 571–578, 2019.

[17] José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at TASS 2018: Sentiment Analysis in Twitter based on Deep Learning. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 37–44, 2018.

[18] José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at TASS 2017: Sentiment Analysis in Twitter based on Deep Learning. In *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2017, co-located with 33th SEPLN Conference (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 37–44, 2017.

[19] José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV at SemEval-2017 task 4: Sentiment analysis using deep learning. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 723–727, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[20] Lluís-F Hurtado, José-Ángel González, and Ferran Pla. Choosing the right loss function for multi-label emotion classification. *Journal of Intelligent & Fuzzy Systems*, 36(5):4697–4708, 2019.

[21] José-Ángel González, Lluís-F. Hurtado, and Ferran Pla. ELiRF-UPV at SemEval-2019 task 3: Snapshot ensemble of hierarchical convolutional neural networks for

contextual emotion detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 195–199, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[22] José-Ángel González, Ferran Pla, and Lluís-F. Hurtado. ELiRF-UPV en TASS 2018: Categorización emocional de noticias (ELiRF-UPV at TASS 2018: Emotional Categorization of News Articles). In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 103–109, 2018.

[23] José Ángel González, Lluís-F. Hurtado, and Ferran Pla. Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. *Information Processing & Management*, 57(4):102262, 2020.

[24] José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at IroSvA: Transformer Encoders for Spanish Irony Detection. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, pages 278–284, 2019.

[25] José-Ángel González, Lluís-F. Hurtado, and Ferran Pla. ELiRF-UPV at SemEval-2018 tasks 1 and 3: Affect and irony detection in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 565–569, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[26] José Ángel González, Encarna Segarra, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Extractive summarization using siamese hierarchical transformer encoders. *Journal of Intelligent & Fuzzy Systems*, 39:2409–2419, 2020. 2.

[27] José-Ángel González, Lluís-Felip Hurtado, Encarna Segarra, Fernando Garcia-Granada, and Emilio Sanchis. Summarization of Spanish Talk Shows with Siamese Hierarchical Attention Networks. *Applied Sciences*, 9(18), 2019.

[28] José-Ángel González, Segarra Encarna, Fernando García-Granada, Emilio Sanchis, and Lluís-F. Hurtado. Siamese hierarchical attention networks for extractive summarization. *Journal of Intelligent and Fuzzy Systems*, 36(5):4599–4607, 2019.

[29] Emilio Sanchis Fernando García-Granada José Ángel González, Julien Delonca and Encarna Segarra. Applying Siamese Hierarchical Attention Neural Networks for multi-document summarization. *Procesamiento del Lenguaje Natural*, 63(0):111–118, 2019.

[30] José Ángel González, Lluís-F. Hurtado, and Ferran Pla. TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter. *Neurocomputing*, 426:58 – 69, 2021.

[31] José-Ángel González, Lluís-F. Hurtado, Encarna Segarra, and Ferran Pla. ELiRF-UPV at SemEval-2018 task 10: Capturing discriminative attributes with knowledge graphs and Wikipedia. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 968–971, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[32] José-Ángel González, Lluís-F. Hurtado, Encarna Segarra, and Ferran Pla. ELiRF-UPV at SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 1034–1037, 2018.

[33] José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. ELiRF-UPV at MultiStanceCat 2018. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 173–179, 2018.

[34] José-Ángel González, Ferran Pla, and Lluís-Felip Hurtado. ELiRF-UPV at IberEval 2017: Stance and Gender Detection in Tweets. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 193–198, 2017.

[35] José-Ángel González, Ferran Pla, and Lluís-Felip Hurtado. ELiRF-UPV at IberEval 2017: Classification Of Spanish Election Tweets (COSET). In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 55–60, 2017.

[36] Lluís-F. Hurtado, Encarna Segarra, Ferran Pla, Pascual Carrasco, and José-Ángel González. ELiRF-UPV at SemEval-2017 task 7: Pun detection and interpretation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 440–443, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[37] Victor Nina-Alcocer, José-Ángel González, Lluís-Felip Hurtado, and Ferran Pla. Aggressiveness detection through deep learning approaches. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 544–549. CEUR-WS.org, 2019.

[38] Fernando García-Granada, Emilio Sanchis, María José Castro Bleda, José-Ángel González, and Lluís-F. Hurtado. Word discovering in low-resources languages through cross-lingual phonemes. In Albert Ali Salah, Alexey Karpov, and Rodmonga Potapova, editors, *Speech and Computer - 21st International Conference, SPECOM 2019, Istanbul, Turkey, August 20-25, 2019, Proceedings*, volume 11658 of *Lecture Notes in Computer Science*, pages 133–141. Springer, 2019.

[39] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[40] Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems. *CoRR*, abs/1512.08756, 2015.

[41] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[42] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[43] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[44] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989.

[45] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theor.*, 39(3):930–945, May 1993.

[46] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[47] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.

[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[50] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[51] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[52] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[53] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

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

[55] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[56] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020.

[57] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[58] Terrence J. Sejnowski and Charles R. Rosenberg. *NETtalk: A Parallel Network That Learns to Read Aloud*, page 661–672. MIT Press, Cambridge, MA, USA, 1988.

[59] Alexander Waibel, Toshiyuki Hanazawa, Geofrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. *Phoneme Recognition Using Time-Delay Neural Networks*, page 393–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[60] Yann Lecun. *Generalization and network design strategies*. Elsevier, 1989.

[61] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery.

[62] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[63] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[64] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 999888:2493–2537, November 2011.

[65] Michael I. Jordan. Chapter 25 - serial order: A parallel distributed processing approach. In John W. Donahoe and Vivian Packard Dorsel, editors, *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471 – 495. North-Holland, 1997.

[66] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179 – 211, 1990.

[67] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997.

[68] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[69] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[70] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā Ser.*, 26:359–372, 1964.

[71] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. Curran Associates, Inc., 2014.

[72] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[73] Jasmijn Bastings and Katja Filippova. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online, November 2020. Association for Computational Linguistics.

[74] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[75] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics.

[76] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On Identifiability in Transformers. In *8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia*, April 2020.

[77] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online, November 2020. Association for Computational Linguistics.

[78] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[79] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[80] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

[81] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[82] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020.

[83] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[84] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[85] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 3294–3302, 2015.

[86] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

[87] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[88] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Lyn Untalan Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. In *In submission to: EMNLP demonstration*, Brussels, Belgium, 2018. In submission.

[89] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[90] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, March 2003.

[91] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010.

[92] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529, 2019.

[93] Alexis CONNEAU and Guillaume Lample. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc., 2019.

[94] Patrick Xia, Shijie Wu, and Benjamin Van Durme. Which *BERT? A survey organizing contextualized encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7516–7533, Online, November 2020. Association for Computational Linguistics.

[95] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[96] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2020.

[97] J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

[98] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NIPS*, pages 6297–6308, 2017.

[99] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019. cite arxiv:1906.08237Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet.

[100] Jan H. Kietzmann, Kristopher Hermkens, Ian P. McCarthy, and Bruno S. Silvestre. Social media? get serious! understanding the functional building blocks of social media. *Business Horizons*, 54(3):241 – 251, 2011. SPECIAL ISSUE: SOCIAL MEDIA.

[101] Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[102] Ferran Pla and Lluís-F. Hurtado. Political tendency identification in Twitter using sentiment analysis techniques. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 183–192, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[103] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In William W. Cohen and Samuel Gosling, editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*. The AAAI Press, 2010.

[104] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welpe. Election forecasts with twitter: How 140 characters reflect the political landscape. *Social Science Computer Review*, 29(4):402–418, 2011.

[105] Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A. Smith. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 293–296, USA, 2010. Association for Computational Linguistics.

[106] Johan Bollen and Huina Mao. Twitter mood as a stock market predictor. *Computer*, 44(10):91–94, October 2011.

[107] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research, 2020.

[108] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA, 2004. Association for Computing Machinery.

[109] Julio Villena-Román, Janine García-Morera, Miguel Ángel García Cumbreras, Eugenio Martínez-Cámara, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López. Overview of TASS 2015. In *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015*, volume 1397 of *CEUR Workshop Proceedings*, pages 13–21. CEUR-WS.org, 2015.

[110] Miguel Ángel García Cumbreras, Julio Villena-Román, Eugenio Martínez Cámara, Manuel Carlos Díaz-Galiano, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López. Overview of TASS 2016. In *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 13–21. CEUR-WS.org, 2016.

[111] Manuel Carlos Díaz-Galiano, Eugenio Martínez-Cámara, Miguel Ángel García Cumbreras, Manuel García Vega, and Julio Villena-Román. The democratization of deep learning in TASS 2017. *Proces. del Leng. Natural*, 60:37–44, 2018.

[112] Eugenio Martínez Cámara, Yudivián Almeida-Cruz, Manuel Carlos Díaz-Galiano, Suilan Estévez-Velarde, Miguel Ángel García Cumbreras, Manuel García Vega, Yoan Gutiérrez, Arturo Montejo-Ráez, Andrés Montoyo, Rafael Muñoz, Alejandro Piad-Morffis, and Julio Villena-Román. Overview of TASS 2018: Opinions, health and emotions. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, volume 2172 of *CEUR Workshop Proceedings*, pages 13–27. CEUR-WS.org, 2018.

[113] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[114] Julio Villena Román, Janine García Morera, Eugenio Martínez Cámara, and Salud M. Jiménez Zafra. Tass 2014 - the challenge of aspect-based sentiment analysis, 2015-03.

[115] Manuel García Vega, Manuel Carlos Díaz-Galiano, Miguel Ángel García Cumbreras, Flor Miriam Plaza del Arco, Arturo Montejo-Ráez, Salud María Jiménez Zafra, Eugenio Martínez Cámara, César Antonio Aguilar, Marco Antonio Sobrevilla Cabezudo, Luis Chiruzzo, and Daniela Moctezuma. Overview of TASS 2020: Introducing emotion detection. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 163–170. CEUR-WS.org, 2020.

[116] Ferran Pla and Lluís-F. Hurtado. Spanish sentiment analysis in twitter at the tass workshop. *Lang. Resour. Eval.*, 52(2):645–672, June 2018.

[117] David Vilares, Yerai Doval, Miguel A. Alonso, and Carlos Gómez-Rodríguez. LyS at TASS 2015: Deep learning experiments for sentiment analysis on Spanish tweets. In *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN. Alicante,*

Spain, September 15th, 2015, volume 1397 of *CEUR Workshop Proceedings*, pages 47–52, 2015.

[118] Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. Learning sentiment lexicons in spanish. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[119] Xabier Saralegi and Inaki San Vicente. Elhuyar at tass 2013. In *XXIX Congreso de la Sociedad Espaola de Procesamiento de lenguaje natural, Workshop on Sentiment Analysis at SEPLN (TASS2013)*, pages 143–150, 2013.

[120] M. Dolores Molina-González, Eugenio Martínez-Cámara, María-Teresa Martín-Valdivia, and José M. Perea-Ortega. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications*, 40(18):7250 – 7257, 2013.

[121] Fermín L. Cruz, José A. Troyano, Beatriz Pontes, and F. Javier Ortega. Building layered, multilingual sentiment lexicons at synset and lemma levels. *Expert Systems with Applications*, 41(13):5984 – 5994, 2014.

[122] Antonio Quirós, Isabel Segura-Bedmar, and Paloma Martínez. LABDA at the 2016 TASS challenge task: Using word embeddings for the sentiment analysis task. In Julio Villena-Román, Miguel Ángel García Cumbreras, Eugenio Martínez Cámara, Manuel Carlos Díaz-Galiano, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López, editors, *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN co-located with 32nd SEPLN Conference (SEPLN 2016), Salamanca, Spain, September 13th, 2016*, volume 1702 of *CEUR Workshop Proceedings*, pages 29–33. CEUR-WS.org, 2016.

[123] Cristian Cardellino. Spanish Billion Words Corpus and Embeddings, August 2019.

[124] Aiala Rosá, Luis Chiruzzo, Mathias Etcheverry, and Santiago Castro. Retuyt in tass 2017: Sentiment analysis for spanish tweets using svm and cnn, 2017.

[125] Ignacio González Godino and Luis Fernando D'Haro. GTH-UPM at TASS 2019: Sentiment analysis of tweets for spanish variants. In Miguel Ángel García Cumbreras, Julio Gonzalo, Eugenio Martínez Cámara, Raquel Martínez-Unanue, Paolo Rosso, Jorge Carrillo-de-Albornoz, Soto Montalvo, Luis Chiruzzo, Sandra Collovini, Yoan Gutiérrez, Salud M. Jiménez Zafra, Martin Krallinger, Manuel Montes-y-Gómez, Reynier Ortega-Bueno, and Aiala Rosá, editors, *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 579–588. CEUR-WS.org, 2019.

[126] Gaël Letarte, Frédérik Paradis, Philippe Giguère, and François Laviolette. Importance of self-attention for sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 267–275, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[127] Artaches Ambartsoumian and Fred Popowich. Self-attention: A better building block for sentiment analysis neural network classifiers. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 130–139, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[128] Eugenio Martínez Cámara. Análisis de opiniones en español. *Procesamiento del Lenguaje Natural*, 56(0):103–106, 2016.

[129] Ángela Almela, Rafael Valencia-García, and Pascual Cantos. Seeing through deception: A computational approach to deceit detection in written communication. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 15–22, Avignon, France, April 2012. Association for Computational Linguistics.

[130] J. A. Cerón-Guzmán and E. León-Guzmán. A sentiment analysis system of spanish tweets and its application in colombia 2014 presidential election. In *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, pages 250–257, 2016.

[131] Víctor D. Cortés, Juan D. Velásquez, and Carlos F. Ibáñez. Twitter for marijuana infodemiology. In *Proceedings of the International Conference on Web Intelligence*, WI '17, page 730–736, New York, NY, USA, 2017. Association for Computing Machinery.

[132] José Antonio García-Díaz, María Pilar Salas-Zárate, María Luisa Hernández-Alcaraz, Rafael Valencia-García, and Juan Miguel Gómez-Berbís. Machine learning based sentiment analysis on spanish financial tweets. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo, editors, *Trends and Advances in Information Systems and Technologies*, pages 305–311, Cham, 2018. Springer International Publishing.

[133] José Antonio García-Díaz, Óscar Apolinario-Arzube, José Medina-Moreira, Harry Luna-Aveiga, Katty Lagos-Ortiz, and Rafael Valencia-García. Sentiment analysis on tweets related to infectious diseases in south america. In *Proceedings of the Euro American Conference on Telematics and Information Systems*, EATIS '18, New York, NY, USA, 2018. Association for Computing Machinery.

[134] Salud María Jiménez-Zafra, M. Teresa Martín-Valdivia, M. Dolores Molina-González, and L. Alfonso Ureña-López. How do we talk about doctors and drugs? sentiment analysis in forums expressing opinions for medical domain. *Artificial Intelligence in Medicine*, 93:50 – 57, 2019. Extracting and Processing of Rich Semantics from Medical Texts.

[135] Aitor García, Seán Gaines, and María Teresa Linaza. A lexicon based sentiment analysis retrieval system for tourism domain. *e-Review of Tourism Research (eRTR)*, pages 35–38, 2012.

[136] Estanislao López-López, María del Pilar Salas-Zárate, Ángela Almela, Miguel Ángel Rodríguez-García, Rafael Valencia-García, and Giner Alor-Hernández. Liwc-based sentiment analysis in spanish product reviews. In Sigeru Omatu, Hugues Bersini,

Juan M. Corchado, Sara Rodríguez, Paweł Pawlewski, and Edgardo Bucciarelli, editors, *Distributed Computing and Artificial Intelligence, 11th International Conference*, pages 379–386, Cham, 2014. Springer International Publishing.

[137] Veronica Rosas, Rada Mihalcea, and Louis-Philippe Morency. Multimodal sentiment analysis of spanish online videos. *IEEE Intelligent Systems*, 28(3):38–45, May 2013.

[138] Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 73–80. The Association for Computer Linguistics, 2014.

[139] Cynthia Van Hee, Els Lefever, and Véronique Hoste. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[140] Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. Irony detection in twitter: The role of affective content. *ACM Trans. Internet Technol.*, 16(3), July 2016.

[141] Shiwei Zhang, Xiuzhen Zhang, Jeffrey Chan, and Paolo Rosso. Irony detection via sentiment-based transfer learning. *Information Processing & Management*, 56(5):1633 – 1644, 2019.

[142] Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. SemEval-2015 task 11: Sentiment analysis of figurative language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478, Denver, Colorado, June 2015. Association for Computational Linguistics.

[143] Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not. *Knowledge-Based Systems*, 108:132 – 143, 2016. New Avenues in Knowledge Bases for Natural Language Processing.

[144] *NeSp-NLP '10: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, USA, 2010. Association for Computational Linguistics.

[145] Lifeng Jia, Clement Yu, and Weiyi Meng. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1827–1830, New York, NY, USA, 2009. Association for Computing Machinery.

[146] Reynier Ortega-Bueno, Francisco Rangel, DI Hernández Farıas, Paolo Rosso, Manuel Montes-y Gómez, and José E Medina Pagola. Overview of the task on irony detection in Spanish variants. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019). CEUR-WS. org*, pages 229–256, 2019.

[147] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[148] S.J.C. Gaulin and D. McBurney. *Evolutionary Psychology*. Pearson/Prentice Hall, 2004.

[149] W. McDougall. *An Introduction to Social Psychology*. Psychology Series. Dover Publications, 2003.

[150] P. Ekman, W.V. Friesen, P. Ellsworth, A.P. Goldstein, and L. Krasner. *Emotion in the Human Face: Guidelines for Research and an Integration of Findings*. Pergamon general psychology series. Elsevier Science, 2013.

[151] ROBERT PLUTCHIK. Chapter 1 - a general psychoevolutionary theory of emotion. In Robert Plutchik and Henry Kellerman, editors, *Theories of Emotion*, pages 3 – 33. Academic Press, 1980.

[152] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[153] *The nature of emotion: Fundamental questions.* Series in affective science. Oxford University Press, New York, NY, US, 1994.

[154] J. Panksepp. *Affective Neuroscience: The Foundations of Human and Animal Emotions*. Series in Affective Science. Oxford University Press, 2004.

[155] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98 – 125, 2017.

[156] Jianhua Tao and Tieniu Tan. Affective computing: A review. In Jianhua Tao, Tieniu Tan, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, pages 981–995, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[157] K. R. Scherer and H. G. Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310–328, Feb 1994. 8195988[pmid].

[158] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, page 70–74, USA, 2007. Association for Computational Linguistics.

[159] Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[160] Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[161] Saif Mohammad and Felipe Bravo-Marquez. WASSA-2017 shared task on emotion intensity. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 34–49, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[162] François-Régis Chaumartin. UPAR7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 422–425, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[163] Zornitsa Kozareva, Borja Navarro, Sonia Vázquez, and Andrés Montoyo. UA-ZBSA: A headline emotion classification through web information. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 334–337, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[164] Ameeta Agrawal and Aijun An. Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, page 346–353, USA, 2012. IEEE Computer Society.

[165] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, page 1556–1560, New York, NY, USA, 2008. Association for Computing Machinery.

[166] Phil Katz, Matt Singleton, and Richard Wicentowski. SWAT-MP:the SemEval-2007 systems for task 5 and task 14. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 308–313, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[167] DK Kirange and RR Deshmukh. Emotion classification of news headlines using svm. *Asian Journal of Computer Science and Information Technology*, pages 104–106, 2012.

[168] Y. Jia, Z. Chen, and S. Yu. Reader emotion classification of news headlines. In *2009 International Conference on Natural Language Processing and Knowledge Engineering*, pages 1–6, 2009.

[169] Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. NTUA-SLP at SemEval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 245–255, 2018.

[170] Man Liu. EmoNLP at SemEval-2018 task 2: English emoji prediction with gradient boosting regression tree method and bidirectional LSTM. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 390–394, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[171] Yanghoon Kim, Hwanhee Lee, and Kyomin Jung. AttnConvnet at SemEval-2018 task 1: Attention-based convolutional neural networks for multi-label emotion classification. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 141–145, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[172] Ji Ho Park, Peng Xu, and Pascale Fung. PlusEmo2Vec at SemEval-2018 task 1: Exploiting emotion knowledge from emoji and #hashtags. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 264–272, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[173] Hardik Meisheri and Lipika Dey. TCS research at SemEval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[174] Angelo Basile, Marc Franco-Salvador, Neha Pawar, Sanja Štajner, Mara Chinea Rios, and Yassine Benajiba. SymantoResearch at SemEval-2019 task 3: Combined neural models for emotion classification in human-chatbot conversations. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 330–334, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[175] Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Jamin Shin, Yan Xu, Peng Xu, and Pascale Fung. CAiRE_HKUST at SemEval-2019 task 3: Hierarchical attention for dialogue emotion classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 142–147, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[176] Sanghwan Bae, Jihun Choi, and Sang-goo Lee. SNU IDS at SemEval-2019 task 3: Addressing training-test class distribution mismatch in conversational classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 312–317, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[177] Parag Agrawal and Anshuman Suri. NELEC at SemEval-2019 task 3: Think twice before going deep. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 266–271, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[178] Yoonhyung Lee, Yanghoon Kim, and Kyomin Jung. MILAB at SemEval-2019 task 3: Multi-view turn-by-turn model for context-aware sentiment analysis. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 256–260, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[179] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification, 2017.

[180] Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotłowski, Willem Waegeman, and Eyke Hüllermeier. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *Proceedings of the 30th*

International Conference on International Conference on Machine Learning - Volume 28, ICML'13, page III–1130–III–1138. JMLR.org, 2013.

[181] Joan Pastor-Pellicer, Francisco Zamora-Martínez, Salvador España-Boquera, and María José Castro-Bleda. F-measure as the error function to train neural networks. In Ignacio Rojas, Gonzalo Joya, and Joan Gabestany, editors, *Advances in Computational Intelligence*, pages 376–384, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[182] J. Bilmes, K. Asanovic, Chee-Whye Chin, and J. Demmel. Using phipac to speed error back-propagation learning. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 4153–4156 vol.5, 1997.

[183] Brendan O'Connor, Michel Krieger, and David Ahn. Tweetmotif: Exploratory search and topic summarization for twitter. In William W. Cohen and Samuel Gosling, editors, *ICWSM*. The AAAI Press, 2010.

[184] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs, 2011.

[185] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210, May 2005.

[186] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[187] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[188] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[189] Saif Mohammad. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[190] James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*, 2007.

[191] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. Multimedia lab @ ACL WNUT NER shared task: Named entity recognition for Twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China, July 2015. Association for Computational Linguistics.

[192] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[193] D.S. Moore, G.P. McCabe, W.M. Duckworth, and S.L. Sclove. *The Practice of Business Statistics Companion Chapter 18: Bootstrap Methods and Permutation Tests*. W. H. Freeman, 2003.

[194] Deirdre Wilson and Dan Sperber. On verbal irony. *Lingua*, 87(1):53 – 76, 1992.

[195] Penelope Brown, Stephen C Levinson, and Stephen C Levinson. *Politeness: Some universals in language usage*, volume 4. Cambridge university press, 1987.

[196] Yanfen Hao and Tony Veale. Support structures for linguistic creativity: A computational analysis of creative irony in similes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31, pages 1376–1381, 2009.

[197] R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev. *The Princeton Encyclopedia of Poetry and Poetics*. Princeton reference. Princeton University Press, 2012.

[198] Herbert Colston and Raymond Gibbs. A brief history of irony. *Irony in language and thought: A cognitive science reader*, pages 3–21, 2007.

[199] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

[200] Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5), September 2017.

[201] Cynthia Van Hee. *Can machines sense irony? : exploring automatic irony detection on social media*. PhD thesis, Ghent University, 2017.

[202] Cynthia Van Hee, Els Lefever, and Véronique Hoste. We usually don't like going to the dentist: Using common sense to detect irony on Twitter. *Computational Linguistics*, 44(4):793–832, December 2018.

[203] Pierre Schoentjes. *La poetica de la ironia.* Cathedra,, 2003.

[204] Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. Irony detection in Twitter: The role of affective content. *ACM Trans. Internet Technol.*, 16(3):1–24, July 2016.

[205] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1601–1612, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[206] Jihen Karoui, Farah Benamara Zitoune, Véronique Moriceau, Nathalie Aussenac-Gilles, and Lamia Hadrich Belguith. Towards a contextual pragmatic model to detect irony in tweets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 644–650, Beijing, China, July 2015. Association for Computational Linguistics.

[207] Byron C. Wallace, Do Kook Choe, and Eugene Charniak. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1035–1044, Beijing, China, July 2015. Association for Computational Linguistics.

[208] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China, July 2015. Association for Computational Linguistics.

[209] Zoltán Gendler Szabó. Compositionality. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2017 edition, 2017.

[210] D.C. Muecke. Irony markers. *Poetics*, 7(4):363 – 375, 1978.

[211] Fernando Poyatos. *La comunicación no verbal*, volume 13. Ediciones AKAL, 1994.

[212] Paolo Rosso, Francisco Rangel, Irazu Hernández Farías, Leticia Cagnina, Wajdi Zaghouani, and Anis Charfi. A survey on author profiling, deception, and irony detection for the Arabic language. *Language and Linguistics Compass*, 12(4), 4 2018.

[213] Alessandra Teresa Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti, Paolo Rosso, et al. Overview of the Evalita 2018 task on irony detection in Italian tweets (Ironita). In *Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)*, volume 2263, pages 1–6. CEUR-WS, 2018.

[214] Eshrag Refaee and Verena Rieser. An Arabic Twitter corpus for subjectivity and sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2268–2273, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[215] Hairo Ulises Miranda-Belmonte and Adrián Pastor López-Monroy. Early fusion of traditional and deep features for irony detection in Twitter. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019.*, pages 272–277, 2019.

[216] Alessandra Teresa Cignarella and Cristina Bosco. ATC at IroSvA 2019: Shallow Syntactic Dependency-based Features for Irony Detection in Spanish Variants. In

Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019., pages 257–263, 2019.

[217] Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. THU_NGN at SemEval-2018 task 3: Tweet irony detection with densely connected LSTM and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[218] Christos Baziotis, Athanasiou Nikolaos, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. NTUA-SLP at SemEval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive RNNs. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 613–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[219] Omid Rohanian, Shiva Taslimipoor, Richard Evans, and Ruslan Mitkov. WLV at SemEval-2018 task 3: Dissecting tweets in search of irony. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 553–559, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[220] Harsh Rangwani, Devang Kulshreshtha, and Anil Kumar Singh. NLPRL-IITBHU at SemEval-2018 task 3: Combining linguistic features and emoji pre-trained CNN for irony detection in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 638–642, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[221] Thanh Vu, Dat Quoc Nguyen, Xuan-Son Vu, Dai Quoc Nguyen, Michael Catt, and Michael Trenell. NIHRIO at SemEval-2018 task 3: A simple and accurate neural network model for irony detection in Twitter. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 525–530, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[222] Francisco Yus. Propositional attitude, affective attitude and irony comprehension. *Pragmatics & Cognition*, 23(1):92–116, 2016.

[223] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[224] Rolandos Alexandros Potamias, Georgios Siolas, and A. Stafylopatis. A transformer-based approach to irony and sarcasm detection. *ArXiv*, abs/1911.10401, 2019.

[225] Javier Iranzo-Sánchez and Ramon Ruiz-Dolz. VRAIN at IroSva 2019: Exploring Classical and Transfer Learning Approaches to Short Message Irony Detection. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 322–328, 2019.

[226] Chiyu Zhang and Muhammad Abdul-Mageed. Multi-task bidirectional transformer representations for irony detection. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, pages 391–400, 2019.

[227] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4619–4629, 2019.

[228] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[229] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics.

[230] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2662–2670, 2017.

[231] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017.

[232] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, pages 1–11, 2015.

[233] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[234] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.

[235] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. Bertje: A dutch bert model, 2019.

[236] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert: Unsupervised language model pre-training for french, 2019.

[237] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics.

[238] Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: Bert for finnish, 2019.

[239] Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, volume 2481. CEUR, 2019.

[240] Jihang Mao and Wanli Liu. Factuality classification using the pre-trained language representation model BERT. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 126–131, 2019.

[241] Jihang Mao and Wanli Liu. A bert-based approach for automatic humor detection and scoring. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 197–202, 2019.

[242] Marcos Pastorini, Mauricio Pereira, Nicolás Zeballos, Luis Chiruzzo, Aiala Rosá, and Mathías Etcheverry. Retuyt-inco at TASS 2019: Sentiment analysis in spanish tweets. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 605–610, 2019.

[243] François Chollet et al. Keras. `https://keras.io`, 2015.

[244] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[245] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8546–8557, 2019.

[246] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: Pretrained language model for scientific text. In *EMNLP*, 2019.

[247] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 09 2019.

[248] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[249] Maite Giménez Fayos, Tomás Baviera, Germán Llorca, José Gámir, Dafne Calvo, Paolo Rosso, and Francisco M. Rangel Pardo. Overview of the 1st classification of spanish election tweets task at ibereval 2017. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 1–14, 2017.

[250] Paolo Rosso and Francisco M. Rangel Pardo. Author profiling in social media: The impact of emotions on discourse analysis. In *Statistical Language and Speech Processing - 5th International Conference, SLSP 2017, Le Mans, France, October 23-25, 2017, Proceedings*, pages 3–18, 2017.

[251] Mariona Taulé, Francisco M. Rangel Pardo, M. Antònia Martí, and Paolo Rosso. Overview of the task on multimodal stance detection in tweets on catalan #1oct referendum. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, pages 149–166, 2018.

[252] Mirko Lai, Alessandra Teresa Cignarella, and Delia Irazú Hernández Farías. itacos at ibereval2017: Detecting stance in catalan and spanish tweets. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN 2017), Murcia, Spain, September 19, 2017*, pages 185–192, 2017.

[253] Isabel Segura-Bedmar. Labda's early steps toward multimodal stance detection. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, pages 180–186, 2018.

[254] Hairo Ulises Miranda-Belmonte and Adrián Pastor López-Monroy. Early fusion of traditional and deep features for irony detection in twitter. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 272–277, 2019.

[255] Yanghoon Kim, Hwanhee Lee, and Kyomin Jung. AttnConvnet at SemEval-2018 task 1: Attention-based convolutional neural networks for multi-label emotion classification. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 141–145, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[256] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[257] Juan Manuel Pérez and Franco M. Luque. Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[258] Franco Martín Luque. Atalaya at TASS 2019: Data augmentation and robust embeddings for sentiment analysis. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, pages 561–570, 2019.

[259] Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[260] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2897–2903, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[261] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019.

[262] Alfonso Ortega, Eduardo Lleida, Rubén San Segundo, Javier Ferreiros, Lluís F. Hurtado, Emilio Sanchis Arnal, María Inés Torres, and Raquel Justo. AMIC: affective multimedia analytics with inclusive and natural communication. *Proces. del Leng. Natural*, 61:147–150, 2018.

[263] Oana Inel, Nava Tintarev, and Lora Aroyo. Eliciting user preferences for personalized explanations for video summaries. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 98–106, New York, NY, USA, 2020. Association for Computing Machinery.

[264] Nattapong Sanchan, Kalina Bontcheva, and Ahmet Aker. Understanding Human Preferences for Summary Designs in Online Debates Domain. *Polibits*, pages 79 – 85, 12 2016.

[265] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[266] Natalie Schluter. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain, April 2017. Association for Computational Linguistics.

[267] F. Liu and Y. Liu. Exploring correlation between rouge and human evaluation on meeting summaries. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):187–196, 2010.

[268] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics.

[269] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[270] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2020.

[271] Annie Louis and Ani Nenkova. Automatically evaluating content selection in summarization without human models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, page 306–314, USA, 2009. Association for Computational Linguistics.

[272] Annie Louis and Ani Nenkova. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300, 2013.

[273] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. Summarization evaluation in the absence of human model summaries using the compositionality of word embeddings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 905–914, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[274] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. Association for Computational Linguistics, 2018.

[275] Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[276] Clément Jumel, Annie Louis, and Jackie Chi Kit Cheung. TESA: A Task in Entity Semantic Aggregation for abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8031–8050, Online, November 2020. Association for Computational Linguistics.

[277] Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[278] Afonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André F. T. Martins, and Shay B. Cohen. Jointly extracting and compressing documents with summary state representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3955–3966. Association for Computational Linguistics, 2019.

[279] Peter J. Liu*, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

[280] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[281] Makbule Gulcin Ozsoy, Ilyas Cicekli, and Ferda Nur Alpaslan. Text summarization of turkish texts using latent semantic analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 869–876, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[282] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004.

[283] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[284] Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[285] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

[286] Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. Document summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2862–2867, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[287] Nadira Begum, Mohamed Fattah, and Fuji Ren. Automatic text summarization using support vector machine. 5:1987–1996, 07 2009.

[288] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[289] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[290] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3075–3081, 2017.

[291] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[292] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[293] Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[294] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3728–3738, Hong Kong, China, November 2019. Association for Computational Linguistics.

[295] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.

[296] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[297] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences, 2020.

[298] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc., 2015.

[299] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[300] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[301] Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics.

[302] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, volume 11856 of *Lecture Notes in Computer Science*, pages 194–206. Springer, 2019.

[303] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[304] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[305] Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[306] Kaichun Yao, Libo Zhang, Tiejian Luo, and Yanjun Wu. Deep reinforcement learning for extractive document summarization. *Neurocomputing*, 284, 02 2018.

[307] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[308] Wen Xiao and Giuseppe Carenini. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3009–3019, Hong Kong, China, November 2019. Association for Computational Linguistics.

[309] Ziqiang Cao, Chengyao Chen, Wenjie Li, Sujian Li, Furu Wei, and Ming Zhou. Tgsum: Build tweet guided multi-document summarization dataset. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2906–2912, 2016.

[310] Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[311] Dario Stojanovski and Alexander Fraser. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60, Belgium, Brussels, October 2018. Association for Computational Linguistics.

[312] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[313] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, AKBC-WEKEX '12, page 95–100, USA, 2012. Association for Computational Linguistics.

[314] Ani Nenkova. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, page 1436–1441. AAAI Press, 2005.

[315] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[316] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992.

[317] Julia Ive, Pranava Madhyastha, and Lucia Specia. Deep copycat networks for text-to-text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3225–3234, Hong Kong, China, November 2019. Association for Computational Linguistics.

[318] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. Gsum: A general framework for guided neural abstractive summarization, 2020.

[319] Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. Boosting factual correctness of abstractive summarization with knowledge graph, 2020.

[320] Hanqi Jin, Tianming Wang, and Xiaojun Wan. Semsum: Semantic dependency guided neural abstractive summarization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8026–8033. AAAI Press, 2020.

[321] Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models, 2020.

[322] Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[323] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models, 2020.

[324] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Prismatic Inc, Steven J. Bethard, and David Mcclosky. The stanford corenlp natural language processing toolkit. In *In ACL, System Demonstrations*, 2014.

[325] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[326] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005.

[327] Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online, November 2020. Association for Computational Linguistics.

[328] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy, July 2019. Association for Computational Linguistics.