uc3m | Universidad Carlos III de Madrid

Master Degree in Big Data Analytics

2019-2020

Master Thesis

"Transformers applied to Stance Detection for Spanish and Catalan languages"

Pavel Razgovorov

Advisor: Isabel Segura Bedmar

Madrid, September 2020

SUMMARY

Stance detection is a task whose objective is to classify the inclination of a given text with regard to a particular subject or topic to be “in favor”, “against” or “neutral”. While in recent years, the task has been extensively researched, most of the work has been concentrated on the English language. The Catalonia Independence Corpus (CIC) is a recent effort to provide a new annotated dataset of tweets for stance classification in both Spanish and Catalan languages, offering a better alternative than the previous MultiStanceCat dataset presented at the IberEval 2018 workshop. This project uses a new transformer-based model, BETO (a fine-tuned BERT for the Spanish language), to the already proposed approaches for the CIC dataset. It is the first time that this novel technique has been applied to this dataset, as well as stance detection tasks for Spanish and Catalan in general. This approach achieves new state-of-the-art performance for both languages with a macro F1 score of 74.5% for the Spanish language and 74% for the Catalan one. In order to compare the influence of the languages, additional experiments were performed using a “combined” dataset by merging both languages’ corpora; however, the obtained results were slightly lower than for the individual languages. Moreover, a comparison of this and other previous approaches (TF-IDF+SVM, FastText and BiLSTM) for both score performance and training/evaluation time are presented. Finally, an error analysis has been conducted for the BETO model’s misclassifications in order to better understand its weaknesses.

Keywords: Stance Detection, Transformers, BERT, BETO, BiLSTM, FasText, SVM, Tweets, Twitter, Word Embeddings, Spanish Language, Catalan Language.

DEDICATION

Kudos to everyone who have helped me with and during this project, in every sense. Thank you all.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter puts in context the most relevant topics in which this thesis focuses on, presenting the initial objectives and the document's structure.

1.1. Motivation

Due to the increasingly rising volumes of textual content generated on the Internet by mass media and Social Media users, the importance of Natural Language Processing (NLP) approaches is increasing. NLP is a specific artificial intelligence field – in which linguistics are commonly related – that studies the interactions between the computer and the human language; in particular, how to program computers to be able process and analyse natural, human language texts. Although the textual content is unstructured, it provides important information and can be helpful for a variety of purposes. It provides public opinions quickly and overwhelmingly on virtually any topic in society: services, products, politics and much more, whereas the traditional methods of public opinion such as polling and surveys are costly and take far more time, money and human resources. For this cause, the need for automated categorisation and opinion mining is on the rise. This knowledge is of particular importance for advertisers, analysts and lawmakers, but also business intelligence, public intelligence, decision support sytems, policy and social science researchers.

Social media in Spain has nowadays a significant influence. Spain, for instance, is one of Twitter's top 15 most popular countries. In July 2020, there were 7.1 million users of this micro-blogging platform in this region [1]. Moreover, the dialogue in Spain is strongly politicised on this social network. Any political incident resonates more rapidly in Twitter than any other platforms through social media posts. This might be because most Spanish politicians and political leaders use Twitter and share their thoughts on many topics directly, arousing heated discussions and generating viral news. These discussions can be later analysed in order to have a better understanding of a concrete political topic [1].

One of Twitter's advantages is the instant access to what their users express which is provided in this micro-blogging platform at no cost. This makes it a great source of text data that can have several use cases: from sentiment analysis and text classification [2] to users' interests detection [3]. The most common task that uses Twitter data is the sentiment analysis one, whose aim is to determine whether a piece of writing is positive, negative or neither (neutral), although it might extend to more sentiments. Another application that uses Twitter data and is closely related to sentiment analysis is stance

[1] https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

detection.

Stance is defined as the lexical and grammatical expression of attitudes, feelings, judgements, or commitment concerning the propositional content of a message [4]. Stance detection is an emerging opinion mining paradigm where it is automatically predicted by machine learning algorithms, in which sentiment analysis is tightly related to it. Both tasks are, nonetheless, not exactly the same [5]: you could be in favour of having a Catalan republic by either exposing its theoretical benefits – which would be classified as a positive sentiment – or by expressing anger towards the Spanish government that does not allow the referendum to happen, which would be classified as negative sentiment. Despite that, both tasks use similar linguistic features, usually associated with attributes such as comparative adjectives, negation and adverbs of degree [6].

In empirical studies measuring public opinions on social media, especially on political and social questions, this paradigm has a significant role. The essence of these topics is often contentious, as opposed views are commonly voiced on various grounds [7].

Nonetheless, although Twitter data might be a great choice to use it in a stance detection problem, it has its disadvantages as well. Twitter messages are:

- Brief (no more than 240 characters).
- Many of them contain just a few words, informal language like slang and Internet words and abbreviations.
- They usually are emotionally charged.

All this makes that tweets are in many cases problematic to categorise even for humans, especially if only a few tweets from a concrete user, without context and without connections with other participants, are available [8].

As mentioned earlier, NLP – and more concretely sentiment analysis – can be a great substitute of more costly and time-consuming polls and surveys [9]. It is important to know, however, that both polls and sentiment analysis studies may not reflect the reality, especially when forecasting [10]). Having this in mind, the subject matter of this thesis will be the stance detection of political discourse in Twitter for both Spanish and Catalan languages. The stance detection topic will be the Catalan self-determination referendum that took place in Catalonia, Spain, on 1st October 2017. This referendum was authorised by the Catalan Parliament on 6th September, but was declared unconstitutional by the Spanish Government and banned by the Spanish Constitutional Court a few days later [11]. During the referendum celebration, many acts of police violence took place from police officers that were displaced some days before from other regions by order of the Spanish government. All these historical events caused intense discussions between the supporters (“independentists”) and the detractors (“unionists”) [12] [13].

As for the stance detection of political discourse in Twitter for Catalan self-determination referendum (commonly known by it numeronym “1Oct” or just “1O”), the task is to de-

Stance	Tweet	Translation
Against	@JonInarritu Aquest és un malalt que hauria d'ingressar al psiquiàtric	@JonInarritu This is a patient who should be admitted to a psychiatric hospital
Favour	Volem una REPÚBLICA CATALANA INDEPENDENT. No una república espanyola, #NoSurrenderCat	We want an INDEPENDENT CATALAN REPUBLIC. Not a Spanish republic, #NoSurrenderCat
Neutral	El tenista destacó que la solución al conflicto pasa por la convivencia	The tennis player stressed that the solution to the conflict passes through coexistence

Table 1.1. TWEETS ON CATALAN 1OCT REFERENDUM

tect whether the tweet represents the stance in favour, against or whether none of them are expressed (neutral stance) towards the Catalan referendum. In table 1.1 you can see some tweet examples, its English translation and their stance.

1.2. Research Objectives

The focus of this thesis is to study different machine learning methods for automatic stance detection on political discourse – more concretely the Catalan referendum – in Twitter. The thesis' research objectives are the following:

1. Review the main dataset for stance detection.
2. Study the main techniques applied to this task.
3. Analyse the selected "Catalan Independence Corpus (CIC)" dataset's structure and develop a pre-processing pipeline for it.
4. Study different machine learning techniques applied to stance detection, exploring both classical and modern approaches such as deep learning techniques.
5. Compare the classifiers' performance in terms of prediction scores and execution time.
6. Compare the best model's scores against the previous state of the art results.
7. Perform an error analysis over the best model's predicted results to evaluate the weaknesses it presents.
8. Draw the main conclusions of this study and identify future directions.

1.3. Document's structure

The rest of this document is organised in the following way: In Chapter 2, we review some previous corpora and presented the main systems developed for stance classification for both English and Spanish/Catalan languages. In Chapter 3, we present the methodology and methods applied to address the task, as well as the selected dataset' structure. In Chapter 4, we illustrate each approach's implementation details and decisions. In Chapter 5, we analyse the obtained results and examine the errors that the best classifier seems to have. Finally, Chapter 6 presents the main conclusions and future work for this study.

2. STATE OF THE ART

The first study [14] about stance detection was focused on domain-specific topics like congressional floor-debate transcripts. Later, in 2014, Rajadesingan and Liu's work[15] classified gun-control related tweets' stance at user level. These authors presumed that if many users retweeted to a specific pair of tweets shortly, this pair of tweets possibly had something in common and held the same opinion on the subject.

Currently, stance detection problems are being tackled from two different perspectives: traditional machine learning and deep learning. Both are presented in the following subsections.

2.1. Traditional Machine Learning Approaches

Stance detection for social media text began to attract popularity after that the SemEval 2016 challenge organised by the National Research Council Canada presented a specific task for it [16]. The challenge was to identify stance from single tweets without considering the conversational context of online forums and authors' information. More concretely, it had two sub-tasks: the first was to predict stance of a tweet in which five different targets ('Atheism', 'Climate Change is a Real Concern', 'Feminist Movement', 'Hillary Clinton', and 'Legalisation of Abortion') and 2,914 labelled training data instances (and for the test, another 1,249 data instances) were given; in the second one, the task was about a prediction of 707 tweets towards the Donald Trump target, in which 78,000 tweets were provided with no stance annotation.

Mohammad et al., 2016 [5] outperformed (with a 58.3% of macro F1 score) the rest of the SemEval 2016 stance classification participant systems by training a linear SVM and having several hand-engineered features (n-grams, sentiment, target, part-of-speech tagging and encodings for different tweet characteristics). Bøhler et al., 2016 [17] also provided a classical approach by using a Multinomial Naive Bayes classifier along with GloVe word vector-based representations. In this case, the approach did not perform as well as the one proposed by Mohammad et al., due to some mistakes made like having a threshold for low-confidence predictions which resulted in a decisive (4.13%) loss in overall macro F1 score, obtaining the 10^{th} place in the ranking with a 62.47% on the test data.

Since the SemEval 2016 stance detection task only had English tweets, this motivated similar initiatives in other workshops. The first task for stance detection in Spanish and Catalan was presented in the IberEval 2017 workshop[2]. The organisers of the workshop offered this task presenting a dataset of tweets in both Spanish and Catalan where the in-

[2] http://stel.ub.edu/Stance-IberEval2017

dependence of Catalonia is discussed [18]. The Italian group iTACOS [19] submitted the best system, which was based on three diverse groups of features: stylistic, structural and context-based. The solution introduced two features that exploit significant characteristics conveyed by the presence of Twitter marks and URLs. The authors explored different traditional machine learning algorithms: Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LG), Decision Tree (DT) and Multinomial Naive Bayes (MNB) combined using a Majority Voting (MV) system. They obtained an macro F1 score of 48.88% for the Spanish language and 49.01% for the Catalan one.

In the next year, a new stance detection task is presented during the IberEval 2018 workshop[3] with the name "MultiModal Stance Detection in tweets on Catalan #1Oct Referendum"[20]. Its goal was to detect the authors' stances – in favour, against or neutral – with respect to the already mentioned Catalan Referendum of 2017 from tweets written in Spanish and Catalan from a multimodal perspective. Multimodal means that, besides the tweet's text, extra information like context (tweet before and after) as well as multimedia (images) aggregated data were available for this task. For this case, the LABDA team obtained the best scores for the Spanish language (F1 score of 28.02%) by using a bag-of-words approach with TF-IDF text vectorization (using only the tweet's text) to then pass it as an input for many classification algorithms, being the Linear Support Vector Machine (SVM) and Multinomial Naive Bayes the most performant ones [21]. The best system for the Catalan language was provided by the CriCa team [22] ((F1 score of 30.68%). This system combined both Catalan and Spanish datasets in order to then apply a stemming to the tweets – so similar words from both languages could end up being the same, which helped to generalise. It also exploited the contextual tweets (the before and after tweets). The authors applied a Linear SVM classifier.

Two years after the presentation of the "MultiModal Stance Detection in tweets on Catalan #1Oct Referendum" dataset, Zotova et al., 2020 [23] proposed a new multilingual (Spanish and Catalan) dataset for stance detection as an effort to offer a better-quality dataset in terms of classes distribution, although it lacks a multimodal property since single tweets were the only type of data collected. The authors surpassed the IberEval 2018 top scores in both languages by using a SVM classifier with a TF-IDF text representation (the text was previously lemmatized and information gain features were added to the SVMs input) obtaining a 60.78% macro F1 score for the Spanish dataset and 58.44% for the Catalan one. A year later, Zotova et al., 2020 [23] achieved to improve the last score using the same system for the Catalan language (obtaining a 58.77% macro F1 score) and the FasText library along with pre-trained word embeddings for the Spanish one (obtaining a 67.48% macro F1 score). The dataset used in that work and the obtained scores with it is further analysed with more detail in Section 3.1.

[3] https://sites.google.com/view/ibereval-2018

2.2. Deep Learning Approaches

Contrary to the long-established machine learning methods, some other authors try to address this task using deep learning approaches. For the already mentioned SemEval 2016 task, these were some of the proposed approaches:

- Wei et al. 2016 [24] presented a Convolutional Neural Network (CNN) – named as "pkudblab" – to use it as a voting system using the softmax results to predict the label of test set instead of predicting them using the accuracy obtained in the validation set as we would normally do.
- Zarrella and Marsh, 2016 (MITRE team) [25] introduced a combination of two recurrent neural network (RNN) classifiers, using the first one to predict hashtags related to the task from two large unlabelled Twitter datasets and the second one – initialised with the first one's results – to train over the SemEval 2016 dataset and then make predictions.

However, neither of these deep learning approaches was able to outperform the linear SVM based system proposed by Mohammad et al., 2016 [5].

Nevertheless, posterior deep learning systems are able to overcome the top systems described above. Sun et al. 2018 [26] proposed a hierarchical attention network of linguistic knowledge such as polarity and the document's statements that recognise the reciprocal effect between text representation and the associated feature sets. Later, in Sun et al., 2019 [27], the network was improved. A new joint neural network was proposed, which could first determine both stance and sentiment simultaneously, and then use a neural stacking model to leverage the first obtained information for the actual stance detection task. The latter approach obtained a 60.16% macro F1 score.

Currently the best score for stance detection in SemEval 2016 is held by Li and Caragea, 2019 [28], using a multi-task model. This model takes into account the sentiment analysis as a separate task and an attention mechanism for each target to then apply a fully-connected layer and a SoftMax one to get the final predictions. The approach obtains a 65.33% as the macro F1 score.

Deep learning approaches have been also used in the IberEval 2017 stance detection task, such as LSTMs [29], Multilayer Perceptrons [30] [31], [32] and other deep learning techniques (CNN, FastText, BI-LSTM) [33].

As for the IberEval 2018 workshop, the Casacufans team used a CNN in order to identify either Spanish or Catalan flags in the provided images[4]. The ELiRF team [34] also introduced CNN with word embeddings and polarity/emotion lexicons. Later, Zotova et al. [23], [35] studied to apply Bi-LSTM networks using the Flair Toolkit, developed by

[4]Unfortunately, the Casacufans team did not publish any kind of notes or explanation for their proposed approach

Akbik et al., 2019 [36]. Deep learning methods did not overcome the traditional machine learning solutions for the IberEval workshops.

3. METHODOLOGY

3.1. The dataset: Tweets on Catalan 1Oct Referendum

Although stance detection has been fairly well researched in the recent years, most the work has been focused on English corpora, leaving this topic on a second level for the rest of the languages. The main reason for this might be a combination of the lack of annotated data as well as popular libraries not having as great support as for English, especially for the case of the Catalan language.

As described in 2, one of the most important initiatives to promote campaigns for Natural Language Processing Systems in Spanish and other Iberian Languages were the IberEval 2017 and 2018 workshops. Taulé [20] describes the dataset for the MultiModal Stance Detection in tweets on Catalan #1Oct Referendum (MultiStanceCat), which was presented at IberEval 2018 evaluation campaign. This dataset is imbalanced – especially for the Catalan language, in which the AGAINST class represents 87.23% of the tweets, while the FAVOR and NEUTRAL constitute a scarce 2.54% and 10.22%, respectively (as seen in Table 3.1).

Zotova et al., 2020 [23] tried to address the shortcomings of this dataset by creating a new Catalan Independence Corpus (CIC)[5] for stance detection in Catalan and Spanish. This new dataset does not extend the previous one, but is created from a collection of tweets prepared for commercial research, which the authors already had at their disposal. The main advantages are that the new dataset is more extensive (4195 more tweets for the Spanish language and 4532 for the Catalan one) and present a more balanced distribution of classes (Table 3.1 shows a comparison of them). One important disadvantage of the CIC dataset over the MultiStanceCat dataset is that it does not include images or the contextual tweets (the tweet before and after it for context) [20]). The contextual information was crucial for obtaining the highest F1-scores in the IberEval 2018 workshop[20].

For our research, we have selected the CIC dataset for two reasons: first, it offers a larger corpus and a more balanced distribution of classes as shown in Table 3.1, and secondly, since the IberEval's dataset is not available anymore on the Internet[6], there is no other choice but to use the CIC one, which is still available for download.

Regarding the source of texts, merely social media data, and the fact that it comes from Twitter, implies that the texts represent a very informal language. Moreover, tweets are usually short texts (tweets are limited to 140 characters), responses to other tweets, abundance of slang vocabulary, emojis, misspelled words, etc. These issues should be addressed to achieve better results in any NLP task. As Singh et al., 2016 states in his

[5] https://github.com/ixa-ehu/catalonia-independence-corpus
[6] https://www.autoritas.net/MultiStanceCat-IberEval2018

	MultiStanceCat				CIC			
Label	Catalan		Spanish		Catalan		Spanish	
	CLS	DIST	CLS	DIST	CLS	DIST	CLS	DIST
Against	5106	87.23%	2099	37.85%	3988	39.79%	4105	40.73%
Favour	149	2.54%	2231	40.23%	3902	38.83%	4104	40.72%
Neutral	598	10.22%	1215	21.91%	2158	21.48%	1868	18.54%
Total	5853		5545		10048		10077	

Table 3.1. CLASSES DISTRIBUTION COMPARISON BETWEEN THE MULTISTANCECAT AND CIC CORPORA

work [37],

> Users create their own words and spelling shortcuts and punctuation, misspellings, slang, new words, URLs, and genre specific terminology and abbreviations. Thus such kind of text demands to be corrected.

The CIC dataset comes as two compressed ZIP files (one per language). Each file inflates another three CSV files: one for training, one for validation and one for testing. The relation split of the dataset among these files is 60%, 20% and 20%, respectively. All the CSV files share the same column structure (with the tab character as the delimiter) which is displayed in Table 3.2.

In similar studies [35], a mixed dataset (preprocessed separately) is created in order to compare a classifier's performance against using it solely on a concrete language, having a slightly lower score than its counterpart as a result. In this thesis, we also evaluate all of our methods using this mixed dataset.

3.2. Development methodology and environment

With the purpose of facilitating access to the performed experiments, all the produced code is available on GitHub[7]. We have used Jupyter Notebook format (one notebook per method). All the development has been carried out using the Google Colab platform[8], which uses the Python language (the version that Colab provides is the 3.6.9 one). The reasoning behind these decisions are the following:

- Python is probably the most popular language among the scientific community, especially for the machine learning one.
- Similarly to the previous point, Jupyter Notebooks are the *de facto* standard to distribute code among the scientific community.

[7] https://github.com/putopavel/stance-detection-for-spanish-and-catalan
[8] https://colab.research.google.com

id_str	TWEET	LABEL
1099284472267182080	RT @EFEnoticias: Arrimadas se presenta a las generales sacar a Sánchez de Moncloa #EFEURGENTE	AGAINST
1102569937447673856	@gabrielrufian Derecho a la autodeterminación - Golpe de estado Presos políticos - Golpistas Merienda con los colegas - Rebelión 15 meses, no hay plan B - Hasta pudrirme de pasta De nada.	AGAINST
1103423009606561792	Veieu tota aquesta gent? És la que avui ha anat a escoltar Ortega Smith a l'acte de la ultradreta vox. Quan les dictadures feixistes tornin a arrasar Europa, ells en seran els primers culpables #shameEU https://t.co/EHjP9x5pdA	FAVOR
10975347106732237e+18	Mossos y Guardia Urbana inician el operativo permanente contra el top manta en Plaza Catalunya https://t.co/7RTf5nZLGN	NEUTRAL

Table 3.2. SOME EXAMPLE TWEETS OF THE CIC DATASET

- The notebooks are hosted at GitHub since it is the largest platform of open source code, which makes it easier to reach to more people, to get a better visibility.
- Google Colab offers an already-prepared environment with GPU support (mainly used for Deep Learning methods) at no cost. It has the ability to load a Notebook that is hosted from GitHub, which makes it a perfect combination for being able to run the code in just a couple of clicks.

In summary, this approach favours the reproducibility of our experiments by delivering the code in a familiar format and by facilitating its testing without the need of complicated setup.

Regarding the development process, all the notebooks follow the same structure: data loading and pre-processing, model training and model evaluation – all for each language dataset. The implementation is organised across little functions, and after that each section presents the same experiment but for each language (the same function calls, but each one having different arguments for the language files). All the implemented proposals are explained in detail in Section 4.2 as well as references to the libraries used for the implementation.

3.3. Evaluation Metrics

Since the thesis' subject dataset represents a multi-class – that is, non-binary – classification problem, the evaluation is present in the three possible classes: against, favour and neutral. The evaluation metric used will be the F1 score. The F1 score, typically used in information retrieval, evaluates accuracy using the statistics precision p and recall r. Precision is the ratio of true label predictions (tp) to all predicted labels ($tp + fp$). Recall is the ratio of concrete label positives to all actual concrete label positives ($tp + fn$). The F1 score is given by:

$$F1 = 2\frac{p \cdot r}{p + r} \quad \text{where } p = \frac{tp}{tp + fp}, r = \frac{tp}{tp + fn}$$

The reason behind choosing this metric is because it weights recall and precision equally, and a good retrieval algorithm will maximise both precision and recall simultaneously. Thus, moderately good performance on both will be favoured over extremely good performance on one and poor performance on the other.

Note that, since this is a multi-class classification problem, we would need to calculate the F1-score for each existent label. In addition to this, we follow two kinds of F1 macro-averages that were used for the IberEval 2017 and 2018 submissions: one taking into account only the FAVOUR and AGAINST classes (used in most previous works) and the other one taking the NEUTRAL class as well:

$$F1_{2classes} = \frac{F1_{favour} + F1_{against}}{2} \quad \text{and} \quad F1_{3classes} = \frac{F1_{favour} + F1_{against} + F1_{neutral}}{3}$$

4. APPROACHES FOR MULTILINGUAL STANCE DETECTION

This chapter presents the pre-processing pipeline of the analysed dataset as well as the different followed approaches in order to tackle the stance detection task.

4.1. Dataset preprocessing

As observed in Table 3.2, we see that tweets may contain URLs, hashtags, user mentions, stopwords and some other issues (such as "emojis"). Some of these elements do not add valuable information to the tweet, but even may introduce noise that makes more difficult training a machine learning model.

With this in mind, we prepared a preprocessing pipeline in order to clean the data by making use of the Texthero[9] novel and efficient Python library, which provides a solid and simple pipeline to clean text data among other things. More concretely, we perform the following tasks on each tweet:

1. Replace not assigned values with empty spaces.
2. Lowercase all text.
3. Remove all URLs.
4. Remove all blocks of digits.
5. Remove all punctuation.
6. Remove extra whitespaces.

Note that the process does not have some other common cleaning operations like stopwords, emojis or diacritics removing as well as word stemming or lemmatization. The main reason for this is that the deep learning approaches presented in this document (see Sections 4.2.3 and 4.2.4) use either embedding files or pretrained models that contain stopwords and many inflected forms of a same word (these have been proved to give better results in learning word representations [38]), so keeping them will (presumably) be taken into account during the algorithms' training. Moreover, some previous studies reveal that removing stopwords from a text might lead to a decrease in the performance of Twitter sentiment classification approaches [39].

Evidence of this behaviour can be seen in Zotova's experiments [35] with a BiLSTM network (presented in Section 4.2.3) over the MultiStanceCat dataset, in which better scores are obtained when the tweets are not lemmatized. The reason for keeping emojis

[9] https://texthero.org

is similar, since the utilisation of emoji characters in sentiment analysis results in higher sentiment scores [40].

Some other preprocessing tasks such as duplicates removal, too short tweets among others were not needed since the CIC dataset's authors already performed those actions.

4.2. Proposed Machine Learning Methods

This section describes the machine learning methods that we have used to deal with the stance detection task.

4.2.1. TF-IDF+SVM

Some of the most common classical Machine Learning approaches to text classification are K-Nearest Neighbour, Winnow, Naive Bayes, Maximum Entropy and Support Vector Machine (SVM), being the latter the most successful among the others [41] [42] [43]. Along with the SVM approach, usually comes the TF-IDF (Term Frequency times Inverse Document Frequency) [44] representation of features.

TF-IDF uses are various, from document/text classification to topic modelling and information retrieval. Its goal is to balance the impact of words that appear the most in a given dataset and thus give less information. In contrary, those who appear the less will have a greater score, representing the most meaningful words in a dataset. This technique transforms text into a vector of numbers (vectorization) like some other popular methods such as Bag-of-Words. More concretely, it is the product of the term frequency and inverse document frequency, which is computed as:

$$tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

where: $tf_{i,j}$ = frequency of i in j,
df_i = number of documents containing i and
N = total number of documents.

The sci-kit learn library[10] offers support for this algorithm to transform our tweets into its corresponding TF-IDF matrix, which can be the input for a Support Vector Machine (SVM) classifier, available as well in the same library.

Some extra actions have been performed during our experiment. In order to avoid SVM's curse of dimensionality [45], the number of features that TF-IDF is storing has been limited to 10,000 features. Moreover, since the optimal hyper-parameters of the SVM classifier cannot be directly learnt from the data, they must be previously tuned. In

[10] https://scikit-learn.org/

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Fig. 4.1. Example of a TF-IDF representation for two sentences (taken from https://medium.com/analytics-vidhya/fundamentals-of-bag-of-words-and-tf-idf-9846d301ff22)

Parameter	Values
C	0.001, 0.1, 1, 10, 100
gamma	1, 0.1, 0.01
kernel	rbf, linear

Table 4.1. SVM'S CROSS-VALIDATION PARAMETER GRID

order to do so, a grid-search cross-validation is performed. The goal of the grid-search technique is to find the best hyper-parameters by doing an exhaustive search (examine all possible combinations of the parameters) through a concrete subset of parameters (a grid). For each combination, the classifier is evaluated by a validation subset, measured by a 5-fold split. Then, the combination that gave the best results (highest scores) is the selected one. Table 4.1 shows the proposed values for each tuned hyper-parameter and Table 4.2 the obtained ones for each language. Once the classifier with the best parameters was found, it was re-trained with the complete training data.

Parameter	Best Obtained Value		
	Spanish	Catalan	Combined
C	10	10	1
gamma	1	1	1
kernel	RBF	RBF	Linear

Table 4.2. BEST SVM HYPER PARAMETERS

	Yahoo		Amazon full		Amazon polarity	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
char-CNN	71.2	1 day	59.5	5 days	94.5	5 days
VDCNN	73.4	2h	63	7h	95.7	7h
fastText	72.3	5s	60.2	9s	94.6	10s

Fig. 4.2. Comparison between FastText and deep learning-based methods (taken from https://fasttext.cc/blog/2016/08/18/blog-post.html)

```
__label__NEUTRAL así es inés arrimadas su carrera en vídeos
```

Fig. 4.3. Example line of the file FastText uses as input data to train a model

4.2.2. FastText library

FastText[11] is a library for text representation and classification using word embeddings created by Facebooks's AI Research (FAIR) lab[12]. The library is the result of regrouping two papers [38] [46]. It is important to note that, although FastText uses neural networks for building the word representations, it is able to train in a fraction of time compared to deep neural networks, and only using the CPU, making it a very efficient approach on training and testing in contrast with some other deep learning approaches, which offer very good performance in exchange for slow training and testing times. Its efficiency comes from the fact that, among other things, it uses a hierarchical classifier instead of a flat structure, in which the different categories are organised in a tree. This decreases the time complexities of training and testing text classifiers from linear to logarithmic with respect to the number of classes.

Something important to highlight from this library is that it needs to have the dataset stored in files with a special format. More concretely, the file needs to have one document (tweet) per line with a prefix indicating its corresponding classification label (see 4.3). This implies that, apart from the performed preprocessing to the dataset, an additional step of saving the clean text alongside its label to a file (actually, one file for training, one for validation and one for testing) is required.

Similarly to described in Section 4.2.1, cross-validation has been applied to this approach as a way to obtain the hyper-parameters that maximise the classifier's performance. In this case, the library provides an automatic method[13] to perform this task, in which everything you need is a validation file with the same format as described in Figure 4.3. Unlike in sci-kit's cross-validation techniques, in this case having pre-defined combinations of hyper-parameters is not needed; but only the amount of time the training will

[11] https://fasttext.cc

[12] Although few information is available on the Internet about the FAIR lab, Facebook provides a website to check their AI projects: https://ai.facebook.com

[13] https://fasttext.cc/docs/en/autotune.html

spend trying to optimise these. By default, the search will take 5 minutes, a sensible measure for most of the cases.

Although the FastText library can load pre-trained word embeddings to then re-train the model over new data, such task has not been performed with this library, leaving it to the deep learning approaches, in which already-trained models have been selected in order to help obtaining greater scores (see Section 4.2.3).

4.2.3. Bidirectional LSTM

Bidirectional Long short-term memory (Bidirectional LSTM or just BiLSTM) are a special kind of LSTM deep learning networks. In the traditional LSTM algorithm by Hochreiter [47], each hidden layer cell's input is dependent on two things: the computation of the cell at the previous step and the input data in the current one. In a BiLSTM network [48], the information flow in both ways, so the computation of the cell at the next step will be taken into account as well. In other words, the sequence is trained by an LSTM network in forward and another one in reverse. Figure 4.5 illustrates this comparison more visually. BiLSTM is the preferred approach, since it has been proven that BiLSTM networks shows improved accuracy over the traditional LSTM for language modelling [49] and therefore for the proposed dataset in Section 3.1. Since the Bidirectional LSTM solely represents one layer of the full proposed network, in the following subsections the whole architecture is exposed. The neural network has been completely implemented with the help of the Keras API[14] (which is built on top of the TensorFlow library[15]).

Embedding Layer

The very first layer of the network is the embedding layer, whose role is to load the word embedding matrix. An embedding matrix contains a vector for each word from the trained vocabulary, and their values represent that word in the vector space. These are used to represent the words and their relationship in a numerical form [50]. Those relationships can be both semantic and syntactic, so words that have similar meanings would be close in the vector space. Some of the most popular word embeddings available have been created using neural network, such as Word2Vec [51], GloVe [52] and more recently using the FastText library, mentioned above.

There are many pre-trained word embeddings publicly available for NLP applications. In this case, the selected word embeddings are actually two (one for Spanish, and another for Catalan):

- The Spanish embeddings come from the Spanish Unannotated Corpora[16], a dataset of 3 billion words that has been trained using FastText, resulting in an embedding

[14] https://keras.io
[15] https://www.tensorflow.org
[16] https://github.com/josecannete/spanish-corpora

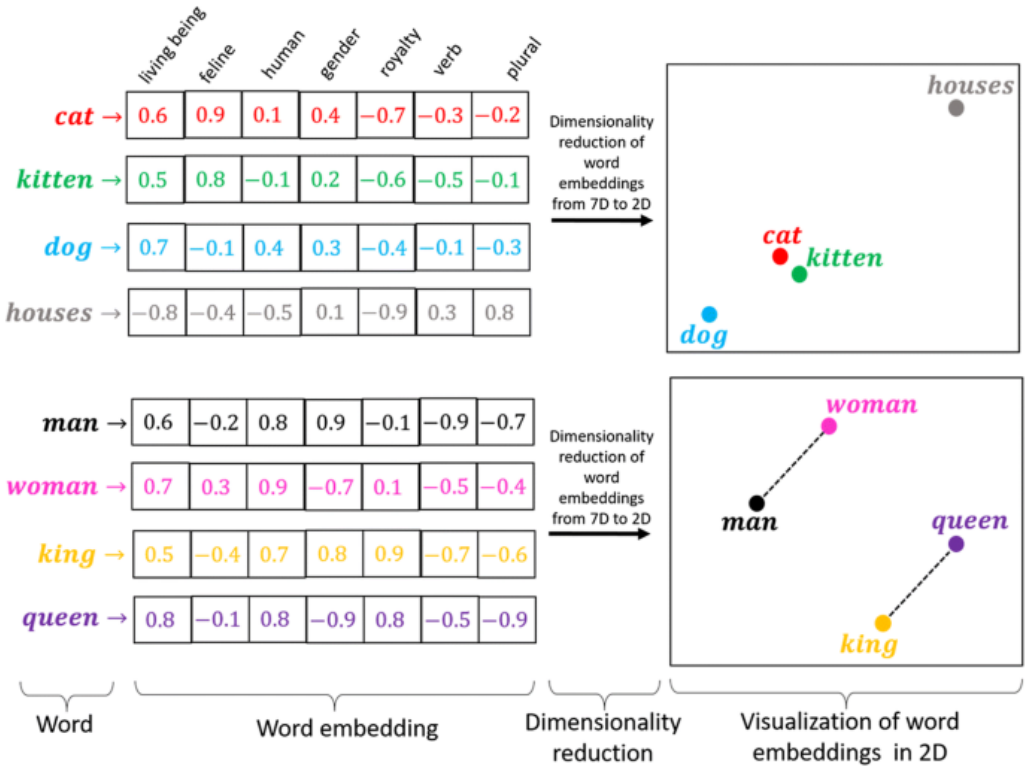

Fig. 4.4. Visual representation of a word embedding matrix in a vector space [53]

matrix of 1,313,423 vectors of dimension 300. This embedding, as some other Spanish word embeddings can be found in the same GitHub repository[17].

- For the Catalan case, there are only a few possibilities to choose from, since the amount of available corpora for this language is very limited. Nevertheless, the NLPL word embeddings repository[18] has two word embedding models available, both of them generated from the CoNLL17 corpus[19], a multi-lingual corpus including texts in Catalan. We decided to choose the ID 34 model, an embedding matrix of 799,020 vectors of dimension 100 generated with the Word2Vec Continuous Skipgram algorithm [51].

- For the combined dataset, since a pre-trained word embedding model for both languages does not exist, the Spanish one is used. The reason for this is that the amount of vectors is richer than the Catalan counterpart. Note that a combination of both word embeddings could not be possible since the dimensions of both do not coincide. Nonetheless, this might still being a reasonable approach since both languages share many words in their vocabulary.

[17] https://github.com/dccuchile/spanish-word-embeddings
[18] http://vectors.nlpl.eu/repository/
[19] http://universaldependencies.org/conll17/

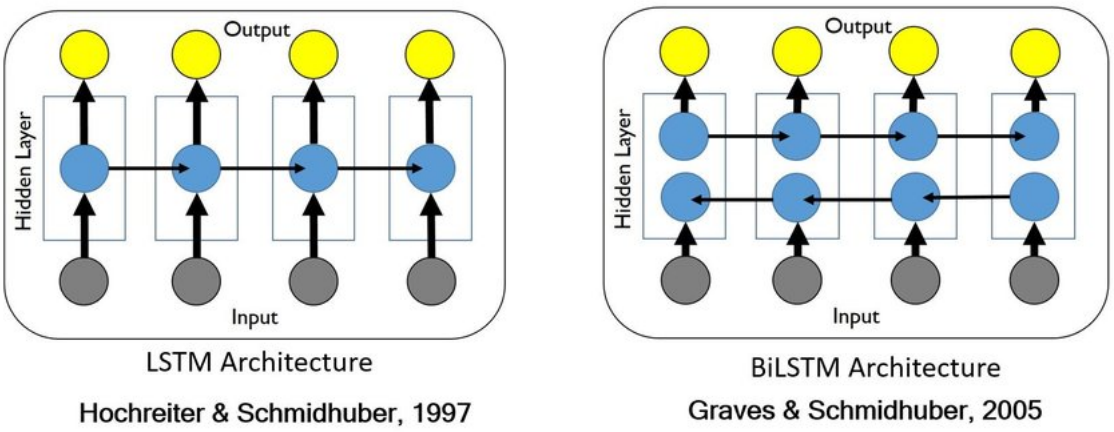

Fig. 4.5. LSTM and BiLSTM architectures, visually compared [54]

BiLSTM Layer

As stated before, LSTM is a one-way recurrent neural network, which transforms input from the beginning to the end. Therefore, it can preserve only relevant data during training from those past inputs (it does not know what the future information is). On the other hand, the Bi-LSTM connects the same output to two hidden layers in opposite directions. The output layer can simultaneously receive information from both forward and backward states. This is why Bi-LSTM can better capture information from the context.

Regarding the number of units, although some studies had successful results using BiLSTMs with 256 units for NLP tasks [55], using 128 units in this network performed slightly better.

1D Global Max Pooling Layer

Since it is not feasible to take each LSTM cells' output, after the BiLSTM layer we apply a pooling layer. A pooling layer is yet another building block of a neural network, whose function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. The most common approach used in pooling is max pooling. Particularly, the global max pooling takes the maximum value of each feature vector from each of the features. Figure 4.6 represents a global max pooling layer of 1 dimension, similar to the one used in the neural network.

This layer will help identifying the strongest trait of the sentence and point out the tweets which give the most information. For instance, some particular words that would possibly determine someone's stance could be identified among the rest.

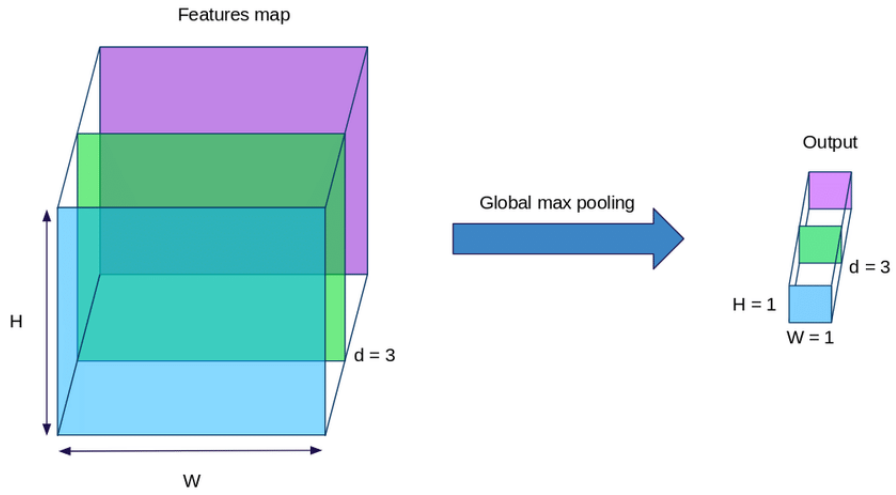

Fig. 4.6. 1D Global Max Pooling Layer visual example [56]

Dense layers

Some studies suggest that adding fully-connected layers (at least one) performs really well [57]. In this network, two dense layers (with dropout regularisation) have been added after the pooling layer:

1. One first layer with 40 units, followed by a dropout regularisation layer with probability of 30%
2. One second layer with 20 units, followed by a 30%-probability dropout layer as well.

Dropout [58] is a regularisation technique to avoid over-fitting by randomly ignoring units (neurons) during the training phase of a certain set of units which is chosen at random (defined by a probability, in our case 0.3) from the previous network's layer. Note that this technique might increase the number of iterations required for the network to converge. However, training time for each epoch gets decreased as there will be less units.

In both layers, a ReLU (Rectified Linear Unit) function has been used as activation function, whose behaviour consists in setting to 0 any negative input, making it a 'piecewise linear function' [59]. This is called a sparse representation and is a desirable property in representational learning as it can accelerate learning and simplify the model. Figure 4.7 displays a graphical interpretation of this function.

Softmax (output) Layer

Finally, in the output layer a softmax function is applied in order to give the prediction probability of each class. This function turn the last linear layer's numeric output vector

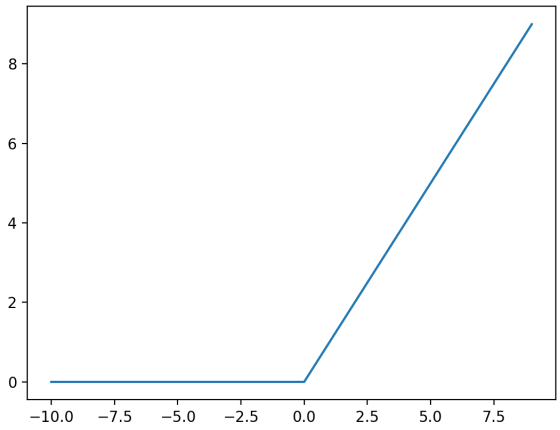

Fig. 4.7. Visualisation of a ReLU function

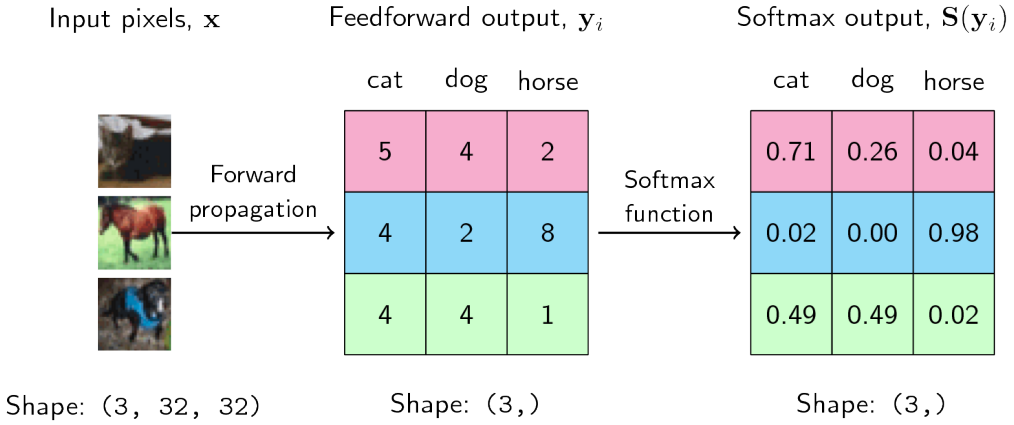

Fig. 4.8. Softmax function illustrated with an example (taken from https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood)

of a multi-class classification neural network into their corresponding probabilities. Each probability consists of an exponential evaluated at an element of a vector, which is normalised by the summation of the exponential of all the elements of that vector (so all of the output numbers add up to one). Figure 4.8 illustrates this in an intuitive way.

It is important to know that some other activation functions like sigmoid do not have place in this situation since those only work for binary classification problems.

Network's Training Details

Besides the network's architecture, there are some other important details that are related to the training phase that are worth mentioning.

- Along with the softmax activation layer described in 4.2.3, the selected loss function

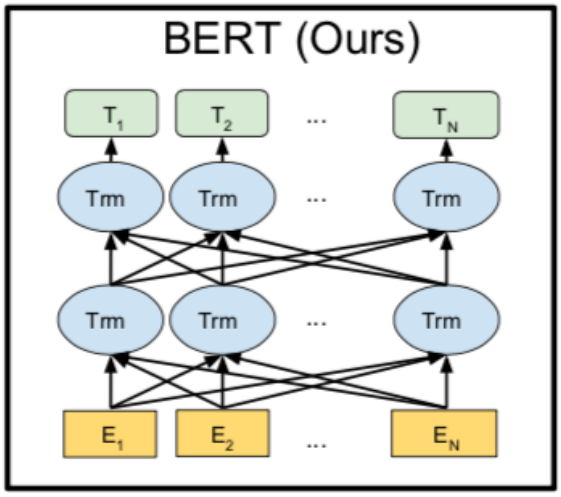

Fig. 4.9. BERT's architecture diagram [66]. Note the bidirectional transformers' flow

is the categorical cross-entropy, one the most suitable for multi-class classification tasks [60].

- The selected training optimiser is Adam, a method that, according to Kingma et al. [61], is 'computationally efficient, has little memory requirement, invariant to diagonal re-scaling of gradients, and is well suited for problems that are large in terms of data/parameters'. The selected parameters are the default ones present in the TensorFlow's Adam class[20].

- The number of training epochs has been set to 100, although the early-stopping technique is being used in order to stop training when the generalisation error checked from the validation set is notably higher than in the training set [62]. The batch size (number of samples processed before the model is updated) has been set to 100.

4.2.4. BETO: The Spanish BERT

Neural language representation models based on LSTM networks as seen in Section 4.2.3 such as ELMo [63], ULMFiT [64], Flair [65] and more have become very popular in the recent years due to their significant performance improvement. However, following closely, Devlin et al. [66] introduced BERT (Bidirectional Encoder Representations from Transformers), a different type of language model based on the Transformer [67] architecture that learns the contextual relationships between the words in a text [68]. This new model, instead of predicting the next word in a text as LSTM networks do, is trained to predict the original sentence from another one in which some words (tokens) have been replaced by a special *mask* token. This can be achieved because, contrary to the sequence learning that LSTM and BiLSTM does, Transformers' input is the whole sentence at once.

[20] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

The text representations generated by BERT have been proved to be have great performance for many NLP problems [67], [69]–[74], even those which were formerly considered difficult, such as question-answering or sentence similarity). As a result, in the recent years quite a few new methods based on this architecture, with some modifications in the architecture or the training objectives, have been developed and presented.

Some of the most important models based on BERT are RoBERTa [75], Transformer-XL [76], XLNet [77], Albert [78], and Reformer [79]. Some other important work for non-English languages has been made as well [80], and one of these is BETO [81], a BERT-based language model pre-trained exclusively on Spanish data[21]. Unfortunately, a Catalan language model does not exist at these moments, so BETO will be employed for all the languages of the dataset.

To the best of our knowledge, this is the first work that exploits BETO to deal with the task of stance detection. The model has 12 self-attention layers with 16 attention-heads each [67], using 1,024 as the hidden size. The model has 110M parameters in total. The authors offer two versions, one with cased data and one with uncased data, which can be used through the HuggingFace Transformers library[22]. This library's philosophy is to provide a very easy-to-use experience while providing state-of-the-art models with performances as close as possible to the original models. Such is the case that it offers cross-compatibility between PyTorch and TensorFlow 2.0 deep learning libraries allowing to easily switch between one and other with no effort. In this case, BETO authors have trained this model using PyTorch[23] as the underlying library, and the same approach is followed for this task.

As explained above, the HuggingFace Transformers library is very simple to use, allowing us to use pre-trained language models in order to re-train them for a specific use case. The main requirement for loading a model is that it needs to exist in the library's models repository[24]. With this, you are getting the model[25] and its associated tokenizer[26] out-of-the-box. You can specify additional configuration[27] in case it is required to change the model's default parameters. For this task, specifying that the classification had three possible labels was mandatory in order to make the model work. This library offers their re-training functionality through a very simple yet powerful API called Trainer[28], in which you specify all the training arguments and datasets.

One important thing to note is that this library expects the dataset to be in a concrete format, which might be confusing in the beginning. Nevertheless, the authors developed

[21] https://github.com/josecannete/spanish-corpora
[22] https://huggingface.co/transformers/
[23] https://pytorch.org
[24] https://huggingface.co/models
[25] https://huggingface.co/transformers/main_classes/model.html
[26] https://huggingface.co/transformers/main_classes/tokenizer.html
[27] https://huggingface.co/transformers/main_classes/configuration.html
[28] https://huggingface.co/transformers/training.html#trainer

a library[29], which is able to load data from many different sources, apply some changes in the data if necessary, among other tasks. This allows to have a concise working pipeline for this kind of models. In this case, besides loading the dataset from their CSV files, the tokenization of the tweets and a label encoding have been performed since the model expected tokens as input and integer labels instead of string ones. A reverse encoding is performed as well once the predictions have been obtained in order to restore their original labels.

[29] https://huggingface.co/nlp

5. EVALUATION AND DISCUSSION

As was explained above, the best setting for each proposed method was selected according to results obtained on the validation dataset. Then, these models are evaluated over the test datasets of each language comparing the predicted stance against the actual stance of each tweet.

5.1. Experiments' results

Table 5.1 displays a summarised view of the methods' scores (presented and explained in Chapter 3) for each language as well as the time taken in order to both train and predict the test dataset. So not only the performance of the method can be evaluated, but the computational cost to train them takes part in this discussion as well.

Before comparing each method, if a global analogy across languages is performed, we can see some interesting facts:

- For the Spanish language, the NEUTRAL case obtains the best F1 scores, which means that it is the easiest class to predict, followed by the AGAINST and the FAVOUR ones. This is also reflected in a higher F1 score for 3 classes than for 2 classes.
- Contrary to the previous case, the Catalan dataset obtains the best scores for the FAVOUR, AGAINST and NEUTRAL stances (in that order). Analogously, since the NEUTRAL label obtains the worst score, the 3-classes F1 scores are slightly below the 2-classes ones.
- The above observations cannot be explained relying on each dataset's class distribution, since the NEUTRAL one represents the minority of tweets and all the classifiers obtain the best scores for Spanish and the worst for Catalan.
- For the combined case, however, the scores among the classes remain similar. This might be due to the balance caused when combining both corpora: the lowest results on one class for one language turn out to be the highest ones for the other, so joining both results in a stabilisation of them. Note, however, that the scores obtained in this dataset do not surpass the corresponding ones for any of the other two languages.

We now discuss the results for each of the methods. The best system is BETO, which is able to outperform the rest of the proposed approaches, even some of them with a notable difference. For the Spanish language, BETO obtains the best scores in the 2 and 3-classes macro F1 score: 74.5% and 77%, respectively. This makes an improvement of

Method	F1 score			Macro F1 score		Training Time
	AG	FA	NE	AG+FA	AG+FA+NE	
	Spanish					
TF-IDF + SVM	**75.0**	73.0	80.0	74.0	76.0	**51s**
FastText	71.0	70.0	81.0	70.5	74.0	5min 2s
BiLSTM	67.0	64.0	74.0	65.5	68.3	9min 18s
BETO	**75.0**	**74.0**	**82.0**	**74.5**	**77.0**	34min 13s
	Catalan					
TF-IDF + SVM	71.0	75.0	64.0	73.0	70.0	**47s**
FastText	68.0	73.0	64.0	70.5	68.3	5min 16s
BiLSTM	62.0	64.0	61.0	63.0	62.3	2min 57s
BETO	**72.0**	**76.0**	**69.0**	**74.0**	**72.3**	33min 40s
	Catalan+Spanish					
TF-IDF + SVM	71.0	72.0	72.0	71.5	71.7	**2min 30s**
FastText	61.0	58.0	49.0	59.5	56.0	5min 48s
BiLSTM	65.0	65.0	66.0	65.0	65.3	10min 48s
BETO	**72.0**	**72.0**	**74.0**	**72.0**	**72.7**	1h 6min 36s

Table 5.1. F1 SCORES (BEST RESULTS IN BOLD) AND TRAINING AND PREDICTION TIMES. AG, FA AND NE ARE ACRONYMS FOR AGAINST, FAVOR AND NEUTRAL, RESPECTIVELY. AG+FA REPRESENTS THE MACRO F1 SCORE FOR THE CLASSES AGAINST AND FAVOR AND AG+FA+NE FOR ALL THEM: AGAINST, FAVOR AND NEUTRAL.

0.5 and and 1 points when compared to the SVM classifier, which obtains similar results. After them, the FastText classifier is having some lower scores (4 and 3 points below BETO) and finally BiLSTM obtains the worst results with a difference of 9 and 8.7 points compared to the best scores. For the Catalan dataset, something similar happens. BETO obtains the best scores with a 74% and 72.3% for the 2 and 3-classes macro F1 score, followed by SVM with 1 and 2.3 points of difference, then FastText is 3.5 and 5 points below and finally the BiLSTM approach obtains the lowest scores with a difference of 11 and 10 points over BETO. In the combined dataset, however, something different is observed: while BETO obtains the best scores, 72% and 72.7% for the 2 and 3-classes macro F1 score with an improvement of 0.5 and 1 over SVM, in this case BiLSTM obtains better scores (7 and 7.5 points below BETO) than FastText, which merely obtains 59.5% and 56% macro F1 scores (a difference of 12.5 and 16.7 points compared to the top scores).

We see that the classic SVM approach obtains very good scores, being the second best method right after the BETO model – even having draws in some scenarios like in the AGAINST score for the Spanish language. These results are similar to the ones obtained in the MultiStanceCat dataset by Zotova, 2019 [35] (in the sense that they outperformed the rest of the proposed systems). Moreover, they actually improve the ones obtained by Zotova et al., 2020 [23] in the same CIC dataset by 4.09% for the Spanish dataset and by 2.96% for the Catalan one[30].

In terms of time and computational effort, SVM reports the fastest time of all the proposals. However, it is important to note that the elapsed time for the combined dataset (Spanish and Catalan) dramatically increases compared to the rest of the approaches (a 200%, approximately). This is probably due to SVM's sensitiveness to the curse of dimensionality [45].

Regarding the FastText library's scores, these results place the method in the third place, closely following the SVM approach for the Spanish and Catalan languages. It is even able to achieve better results classifying the neutral stance for the Spanish language. However, it obtains the worst scores for the combined dataset (Catalan and Spanish). Moreover, our scores are lower (a decrease of 2.66% for Spanish and a 2.06% for the Catalan) than those obtained by Zotova et al., 2020 [23].

In time and computational cost terms, not much can be said about FasText's efficiency, since we are setting a fixed time in order to perform the parameter tuning during the model training. While it is true that before applying the parameter auto-tuning, the execution times were extremely fast (less than a second in most cases), this is not a representative measure since the obtained scores were very poor. However, setting five minutes for training (and some extra seconds for the predictions and other things) seems to provide reasonable F1 scores. Moreover, the elapsed time for the combined dataset almost has no difference between the other, smaller datasets, making this alternative a perfect fit in case

[30] the measures compared are the two-classes F1 score for against Zotova's $F1_{macro}$ score (the same measure, actually, since the neutral stance is never considered in the second case)

a scalable solution is needed.

The BiLSTM model obtains the lowest scores for each stance and language (with the exception of FastText in the combined dataset), making it the worst approach among the proposed ones. These results are similar to the ones obtained by Zotova et al., 2020 [23], with an increase of 12.37% for the Spanish dataset and 6.85% for the Catalan one.

With reference to the training time of the BiLSTM model, a slight increase can be observed compared to the previous approaches. By examining the elapsed time for Catalan language, a notable decrease is perceived. This may be due to the size of the Catalan word embeddings, which is much smaller than its Spanish counterpart (which is used in both Spanish and combined dataset).

It is important to note, however, that both BiLSTM and BETO are being trained using GPU acceleration, a possibility that SVM nor FastText provide. This implies that, since the algorithms are not evaluated under the same conditions, a rigorous comparison of their training times cannot be performed; on the other hand, the fact that neural networks can benefit from GPU acceleration means that these approaches can be evaluated in a reasonable time (otherwise they would be completely impractical), being in the same magnitude as the classical approaches.

As said before, BETO obtains the best scores in these experimentation for each language (with the exception of the AGAINST F1 score, in which performed as good as the TF-IDF + SVM approach). This is the first time a bidirectional transformer has been implemented to deal with the stance detection task on this dataset. BETO is able to overcome all approaches proposed by Zotova et al., 2020.

By comparing these results with the best ones obtained by Zotova et al., 2020 [23], the BETO language model outperforms the FastText library (with the FastText pre-trained word embeddings) for both languages in every F1 score. For the Spanish case, the AGAINST and FAVOUR F1 scores increased in a 2.46% and 4.03% while for the 2-classes macro F1 score the gain was 2.86%. For the Catalan one, the improvement was of a 1.8%, 5.25% and 3.54%, respectively. This establishes these marks as the state-of-art ones. Since the previous results were only taking into account the AGAINST and FAVOUR stances, neither the NEUTRAL F1 and the 3-classes macro F1 scores can be compared in any way.

A possible drawback of the BETO model is that it is by far the approach that takes the most time for both training and evaluation, even if the GPU acceleration is being in use. This implies that experimenting using this model is very laborious and almost unpractical in very large datasets.

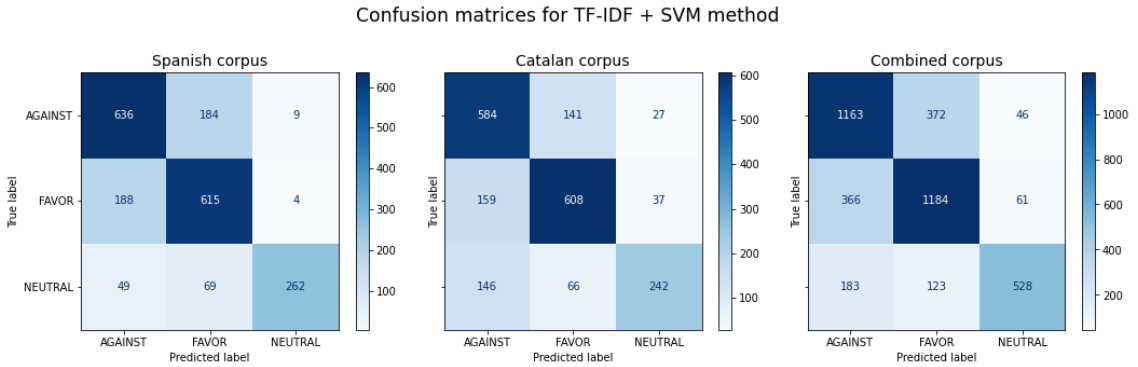

Fig. 5.1. Confusion matrices for TF-IDF + SVM method

5.2. Error Analysis

In this section, the models' predictions are intuitively analysed both in general terms by examining their correspondent confusion matrices in order to explain the obtained F1 scores as well as in detail by checking some concrete failed predictions that BETO – the best model in terms of F1 scores as discussed in Section 5.1 – performed.

5.2.1. Confusion matrices

For the SVM approach (Figure 5.1), the most confused stances are AGAINST and FAVOUR. In those cases, the classifier usually seems to believe that the target stance is definitely not NEUTRAL, especially for the Spanish dataset. The opposite case seems slightly different, since we observe more NEUTRAL stances classified as either AGAINST or FAVOUR. Nevertheless, despite the amount of NEUTRAL true positives being lower than their AGAINST and FAVOUR counterparts, it obtains the best F1 score since the latter are misclassifying a lot of AGAINST stances as FAVOUR and vice versa, resulting in a lower precision and recall (contrary to what happens in the NEUTRAL case).

With respect to the FastText method (Figure 5.2), the results are somehow similar to the ones obtained using the SVM, but with some key differences. The most noticeable is that, for the combined dataset, most of the NEUTRAL stances are actually classified as AGAINST ones (they are indeed more than the correctly predicted ones; almost the double). The same happens to the FAVOUR class, although the difference is not that towering. This is basically the reason of such low scores observed in Table 5.1.

As for the BiLSTM technique (Figure 5.3), the confusion matrices resemble a lot to the SVM's ones. The main distinction is that BiLSTM misclassifies many more cases, especially the FAVOUR stances (the NEUTRAL one, however, remains similar and even gets less false AGAINST and FAVOUR classifications). It is important to note as well that the matrix for the Catalan dataset is the most sparse overall due to the large amount of AGAINST and FAVOUR stances misclassified as NEUTRAL ones, something that almost does not happen in the rest of the classifiers (not even for this one in the Spanish

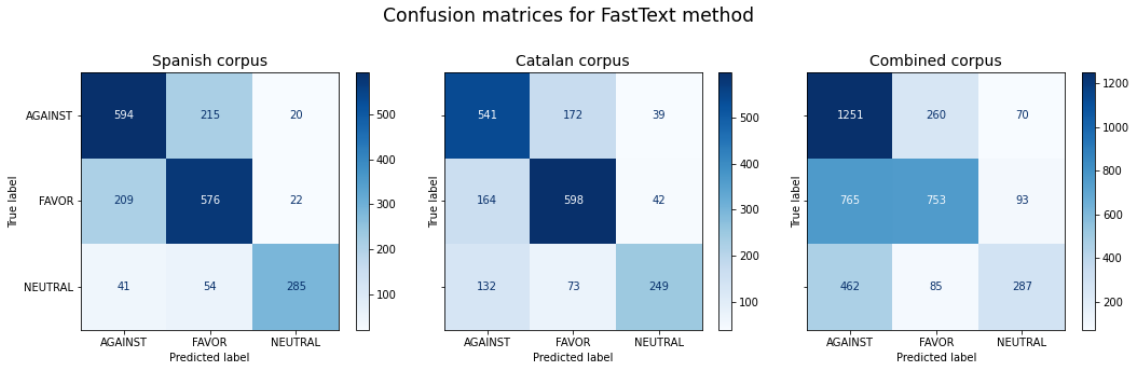

Fig. 5.2. Confusion matrices for FastText method

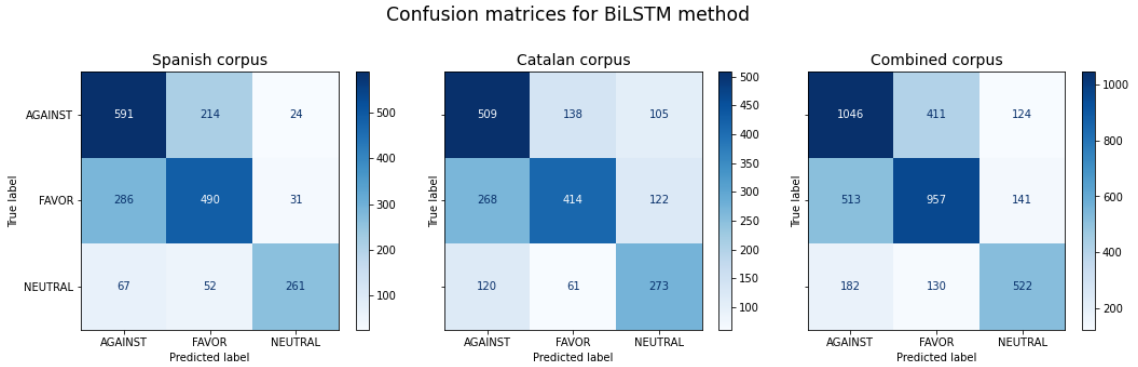

Fig. 5.3. Confusion matrices for BiLSTM method

and Combined corpora).

Finally, for the BETO language model (Figure 5.4), a greater number of correctly classified instances can be observed compared to the rest of the classifiers for each language in general. The remarkable exceptions are the FAVOUR correct classifications for the Spanish language – in which SVM and FastText seem to perform better – and the AGAINST ones for the Catalan dataset – only surpassed by the SVMs ones in this case. For the rest of the classifications for each language, it follows the same structure that the previously presented methods, but having slightly better results for the majority of the cases.

It is indeed observed that the problem of being able to distinguish between the two possible non-NEUTRAL stances is plausible among all the proposed techniques, which might indicate that extra work must be performed in order to be able to distinguish these two opposite stances. This is further discussed in Section 6.

5.2.2. Some examples of misclassifications

In Table 5.2, some misclassified tweets are exposed in order to illustrate some of the encountered flaws of both the BETO classifications and the analysed dataset itself. Although the translation of each tweet has been added for clarification, the original tweets are where

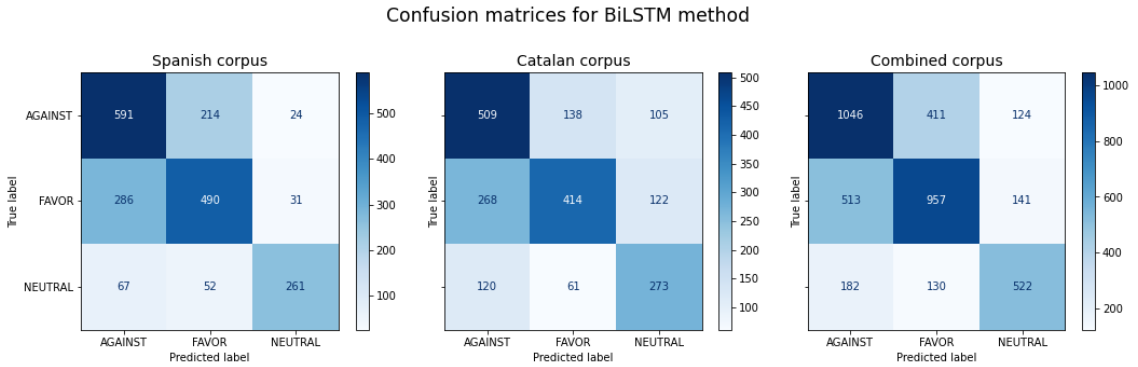

Fig. 5.4. Confusion matrices for BETO method

the focus must be put on.

In the first example, we observe a tweet in which a clear AGAINST stance is being represented. The predicted stance is, however, FAVOUR, contrary to the expected one. This is a perfect example of the confusion that the proposed classifiers seem to have as seen in Section 5.2.1. There is no clear evidence of the possible reasoning for the model to classify this tweet as a FAVOUR one, although we could infer that possibly the "Catalan republic" term is referred as something that the model relates to a FAVOUR stance, without taking into consideration that the author qualifies it as "pure fascism" (a term usually used to express extreme disagreement). Contrary to this, we have the fourth example, a tweet that presumably has nothing to do with the Catalan independence. In that case, the stance is not easy to classify even manually. There are many examples like this in which either the model makes a mistake, because the stance is not very clear, or something in-between.

Another thing that is observed in the pre-processed tweets is that some mistakes are made during the pre-processing task. If we check the second's example cleaned tweet,

> well086 o sea que si yo tengo un amigo votante de en comú al que hace un año le estoy comiendo la oreja para que se haga indepe cuando ya está a punto de decidirse ¿va el tardà y le llama estúpido ¿ésto no es denunciable

The Spanish opening questions marks ("¿") were not removed, which might lead to an incorrect text representation for some words that come either after or before it. The dataset has many more cases like this one in which strange symbols were being kept. Some other things that were not properly cleaned were the "RT" ("Retweet") keyword, user mentions, hashtags and other strange, misspelled words (the latter ones might be laborious to either fix or remove) like in the fifth example (this might be something debatable, since the model might relate some hashtags and users to concrete stances).

Finally, the dataset itself might have some incorrectly tagged stances (as stated by Zotova et al., 2020 [23]) which can be observed in the third, fourth and fifth examples. The third one differs in the other two in the fact that it is actually referencing the Catalan

Tweet	Translation	Dataset Stance	Predicted Stance
En la Republica Catalana los presidentes de organizacciones particulares deciden sobre el bien y el mal. Puro fascismo	In the Catalan Republic, the presidents of private organizations decide on good and evil. Pure Fascism	AGAINST	FAVOR
@Well086 O sea, que si yo tengo un amigo votante de En Comú al que hace un año le estoy comiendo la oreja para que se haga indepe, cuando ya está a punto de decidirse, ¿va el Tardà y le llama estúpido? ¿Ésto no es denunciable?	@Well086 In other words, if I have a friend who is a voter of En Comú who for a year I have been sweet-talking to so that he becomes independent, when he is about to make up his mind, does Tardà go and call him stupid? Is this not reportable?	FAVOR	AGAINST
La #Hacienda catalana no era la única 'estructura de Estado' que planeaba el #Govern de Puigdemont	The Catalan #Hacienda was not the only 'State structure' planned by the #Govern of Puigdemont	AGAINST	NEUTRAL
Guaidó denuncia que funcionaris del Govern veneçolà es senten segrestats per la dictadura	Guaidó denounces that Venezuelan government officials feel kidnapped by the dictatorship	NEUTRAL	AGAINST
@ManuelValls anuncia la incorporación de @MLuzGuilarte, @ParEraEva y @NoemiMartinCs a la candidatura #VallsBCN2019 #vallsporlaigualdad #DiaInternacionalDeLaMujer	@ManuelValls announces the incorporation of @MLuzGuilarte, @ParEraEva and @NoemiMartinCs to the candidacy # VallsBCN2019 #vallsporlaigualdad #DiaInternacionalDeLaMujer	AGAINST	NEUTRAL

Table 5.2. SAMPLE TWEETS OF MISCLASSIFIED STANCES.
EMOJIS AND URLS WERE REMOVED)

republic. In this case, no clear statement is being made about the topic; it seems more like an explanatory or illustrative text, probably the title of a newspaper's article – which presumably is something objective –, so in this case the model might actually correctly predicted this tweet as neutral, while the semi-automatic annotation process made a mistake annotating as favour. The other two examples, however, are loosely tied to the topic, which makes them difficult to be classified. For instance, in the fourth example, classified as NEUTRAL, can be predicted as AGAINST since many "unionists" believed that the Catalan government acted like a dictatorship. In addition to this, it is very hard to deduct in the fifth example that Manuel Valls were directly related to the "Ciudadanos" political party, which was against the Catalan independence referendum since by just looking at the text everybody would say that the tweets stance seems to be NEUTRAL.

6. CONCLUSIONS AND FUTURE WORK

This project explores different stance detection techniques, such as SVM, FastText, BiLSTM and the novel proposed BETO, is a BERT model trained on a large Spanish dataset, applied to a non-English (Spanish and Catalan) dataset. Most of previous Spanish and Catalan stance detection works used an imbalanced dataset, leading to moderate results. Recently, the Catalan Independence dataset (CIC)[23] was published, providing a dataset with a more reasonable class distribution. This new dataset offers a better alternative to the previous MultiStanceCat dataset, leading to a systematic increase of performance by just applying the same techniques as before. In this case, the innovative BETO transformer is the very first attempt to tackle down this task for both languages, being this the project's main contribution. Such unprecedented approach is able to outperform state-of-the-art results[23] with little to no configuration, besides the one required to adapt the pre-trained model for this concrete challenge.

In addition to that, an error analysis is performed in order to compare the proposed approaches' performance as well as an in-depth examination of the BETO model's misclassifications in order to enlighten some of the possible causes of the model's errors and weaknesses. As observed, the main obstacle for the BETO model to properly detect stance is the lack of some context (like who is the tweet's author referring to) as well as some messages being not very clear about whether they are against or in favour. Moreover, although the pre-processing seems to do an effective work, there still might be some improvements to be made.

Having said this, here are some possible improvements and further experimentation that can be performed following the research lines proposed in this work:

- Improve and adapt the pre-processing pipeline for each approach: in this work, the text pre-processing step has been completely homogeneous for all the approaches in order to make the results more comparable. However, better approaches can be proposed. In one hand, some fine-tuning in the non-alphabetic symbols removal can be proposed in order to clean specific characters like the opening question mark ("?") among others. In the other hand, some other alternative processes can be tried as well, such as removing user handles, hashtags, repeated characters in words and other non-textual or non-relevant information. For the approaches in which no pre-trained models are involved, lemmatization of words may also be useful in order to reduce the vocabulary size. It is important to note that all these possibilities would need to be measured in order to compare the effectiveness of each alternative.

- Extract additional information from the tweet. Usually, tweets not only contain text, but links to either referenced tweets, articles, images, among other things. We could use these information to obtain more context from the tweet. For instance,

referenced tweets and articles could be used to enlarge the tweet's text. Moreover, performing a Named Entities Extraction can be used to detect the tweet's target, which might help reducing confusions as observed in Section 5.2.2.

- Train the BiLSTM and BETO models from scratch: currently, we have applied pre-trained models to initialise them. However, it might be interesting to train them from scratch with no previous initialisation. This allows us to know how much pre-trained models affect the approaches' performance.

- Debug the BETO model: the HuggingFace Transformers library offer tools like TensorBoard[31] in which you can analyse many kind of metrics during the training phase of the model. Some possible future work would be to inspect whether the model is having some weak point in its configuration and analyse a possible solution of improvement.

- Tune the BETO model's parameters: this could be done along with the previous proposal. Although the HuggingFace Transformers library might seem limited in terms of network customisation, it currently offers some fine-grain configurations[32] to adapt the model to our needs. Maybe putting some focus on this functionality would allow us to improve BETO's performance.

- Try other Transformer models: besides BETO, there are more multi-lingual models – like XML-R [82] – that could perform better than BETO.

- Combine both Spanish and Catalan pre-trained word embedding models: the proposal is to find two models that match in vector dimension in order to merge them in a single model. With this done, we could evaluate the BiLSTM approach with these new word vectors in order to compare it with the used ones for this project. An alternative for this would be to train a new word embedding model by combining both training corpora and then use it with the BiLSTM network.

Finally, it is important to highlight as well that all the developed code for each approach is available on GitHub[33] in order to reproduce the experiments that were carried out for this project and freely analyse, extend and improve them.

Acknowledgments

This work was supported by the Research Program of the Ministry of Economy and Competitiveness - Government of Spain, (DeepEMR project TIN2017-87548-C2-1-R) and the Interdisciplinary Projects Program for Young Researchers at Universidad Carlos III of Madrid founded by the Community of Madrid (NLP4Rare-CM-UC3M).

[31] https://www.tensorflow.org/tensorboard
[32] https://huggingface.co/transformers/main_classes/configuration.html
[33] https://github.com/putopavel/stance-detection-for-spanish-and-catalan

BIBLIOGRAPHY

[1] F. Guerrero-Solé, "Interactive behavior in political discussions on twitter: Politicians, media, and citizens' patterns of interaction in the 2015 and 2016 electoral campaigns in spain", *Social Media+ Society*, vol. 4, no. 4, p. 2 056 305 118 808 776, 2018.

[2] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!", in *Fifth International AAAI conference on weblogs and social media*, Citeseer, 2011.

[3] S. Piao and J. Whittle, "A feasibility study on extracting twitter users' interests using nlp tools for serendipitous connections", in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, IEEE, 2011, pp. 910–915.

[4] D. BIBER and E. FINEGAN, "Styles of stance in english: Lexical and grammatical marking of evidentiality and affect", *Text - Interdisciplinary Journal for the Study of Discourse*, vol. 9, no. 1, 1989. DOI: 10.1515/text.1.1989.9.1.93. [Online]. Available: https://doi.org/10.1515/text.1.1989.9.1.93.

[5] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, *Stance and sentiment in tweets*, 2016. arXiv: 1605.01655 [cs.CL].

[6] A. Jaffe *et al.*, *Stance: sociolinguistic perspectives*. OUP USA, 2009.

[7] A. AlDayel and W. Magdy, *Stance detection on social media: State of the art and trends*, 2020. arXiv: 2006.03644 [cs.SI].

[8] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data", in *Coling 2010: Posters*, 2010, pp. 36–44.

[9] N. Beauchamp, "Predicting and interpolating state-level polls using twitter textual data", *American Journal of Political Science*, vol. 61, no. 2, pp. 490–503, 2017.

[10] P. Burnap, R. Gibson, L. Sloan, R. Southern, and M. Williams, "140 characters to victory?: Using twitter to predict the uk 2015 general election", *Electoral Studies*, vol. 41, pp. 230–233, 2016.

[11] D. Cetrà, E. Casanas Adam, and M. Tàrrega, "The 2017 catalan independence referendum: A symposium", *Scottish Affairs*, vol. 27, pp. 126–143, Feb. 2018. DOI: 10.3366/scot.2018.0231.

[12] V. Hernández-Santaolalla and S. Sola-Morales, "Postverdad y discurso intimidatorio en Twitter durante el referéndum catalán del 1-O", es, *Observatorio (OBS*)*, vol. 13, pp. 102–121, Mar. 2019. [Online]. Available: http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S1646-59542019000100006&nrm=iso.

[13] R. C. Polaino, E. V. Cirujano, and L. T. Fuentes, "Twitter como herramienta de comunicación polıtica en el contexto del referéndum independentista catalán: Asociaciones ciudadanas frente a instituciones públicas", *Revista ICONO14 Revista cientıfica de Comunicación y Tecnologıas emergentes*, vol. 16, no. 1, pp. 64–85, Jan. 2018. DOI: 10.7195/ri14.v16i1.1134. [Online]. Available: https://doi.org/10.7195/ri14.v16i1.1134.

[14] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts", in *Proceedings of EMNLP*, 2006, pp. 327–335.

[15] A. Rajadesingan and H. Liu, *Identifying users with opposing opinions in twitter debates*, 2014. arXiv: 1402.7143 [cs.SI].

[16] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "SemEval-2016 task 6: Detecting stance in tweets", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 31–41. DOI: 10.18653/v1/S16-1003. [Online]. Available: https://www.aclweb.org/anthology/S16-1003.

[17] H. Bøhler, P. Asla, E. Marsi, and R. Sætre, "IDI@NTNU at SemEval-2016 task 6: Detecting stance in tweets using shallow features and GloVe vectors for word representation", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 445–450. DOI: 10.18653/v1/S16-1072. [Online]. Available: https://www.aclweb.org/anthology/S16-1072.

[18] M. Taulé *et al.*, "Overview of the task on stance and gender detection in tweets on catalan independence at ibereval 2017", in *2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017*, CEUR-WS, vol. 1881, 2017, pp. 157–177.

[19] M. Lai, A. T. Cignarella, D. I. HERNANDEZ FARIAS, *et al.*, "Itacos at ibereval2017: Detecting stance in catalan and spanish tweets", in *IberEval 2017*, CEUR-WS. org, vol. 1881, 2017, pp. 185–192.

[20] M. Taulé, F. M. R. Pardo, M. A. Martí, and P. Rosso, "Overview of the task on multimodal stance detection in tweets on catalan# 1oct referendum.", in *IberEval@SEPLN*, 2018, pp. 149–166.

[21] I. Segura-Bedmar, "Labda's early steps toward multimodal stance detection", in *IberEval@SEPLN*, 2018.

[22] C. A. Cuquerella and C. C. Rodríguez, "Crica team: Multimodal stance detection in tweets on catalan 1oct referendum (multistancecat)", in *IberEval@SEPLN*, 2018.

[23] E. Zotova, R. Agerri, M. Nuñez, and G. Rigau, *Multilingual stance detection: The catalonia independence corpus*, 2020. arXiv: 2004.00050 [cs.CL].

[24] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang, "Pkudblab at SemEval-2016 task 6 : A specific convolutional neural network system for effective stance detection", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 384–388. DOI: 10.18653/v1/S16-1062. [Online]. Available: https://www.aclweb.org/anthology/S16-1062.

[25] G. Zarrella and A. Marsh, *Mitre at semeval-2016 task 6: Transfer learning for stance detection*, 2016. arXiv: 1606.03784 [cs.AI].

[26] Q. Sun, Z. Wang, Q. Zhu, and G. Zhou, "Stance detection with hierarchical attention network", in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 2399–2409.

[27] Q. Sun, Z. Wang, S. Li, Q. Zhu, and G. Zhou, "Stance detection via sentiment information and neural network model", *Frontiers of Computer Science*, vol. 13, no. 1, pp. 127–138, 2019.

[28] Y. Li and C. Caragea, "Multi-task stance detection with sentiment and stance lexicons", 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)*, 2019, pp. 6300–6306.

[29] M. Wojatzki and T. Zesch, "Neural, non-neural and hybrid stance detection in tweets on catalan independence", in *IberEval@SEPLN*, 2017.

[30] D. A. García and A. M. L. Flor, "Stance detection at ibereval 2017: A biased representation for a biased problem", in *IberEval@SEPLN*, 2017.

[31] J.-Á. González, F. Plà, and L.-F. Hurtado, "Elirf-upv at ibereval 2017: Stance and gender detection in tweets", in *IberEval@SEPLN*, 2017.

[32] L. C. Chuliá and S. F. Sánchez, "Classification of spanish election tweets (coset) with neural networks.", in *IberEval@ SEPLN*, 2017, pp. 43–48.

[33] U. Politecnica De Valencia, "Comparative study of neural models for the coset shared task at ibereval 2017",

[34] J.-Á. González, L.-F. Hurtado, and F. Plà, "Elirf-upv at multistancecat 2018", in *IberEval@SEPLN*, 2018.

[35] E. Zotova, "Automatic stance detection on political discourse in twitter", 2019.

[36] A. Akbik *et al.*, "FLAIR: An easy-to-use framework for state-of-the-art NLP", in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 54–59. DOI: 10.18653/v1/N19-4010. [Online]. Available: https://www.aclweb.org/anthology/N19-4010.

[37] T. Singh and M. Kumari, "Role of text pre-processing in twitter sentiment analysis", *Procedia Computer Science*, vol. 89, no. Supplement C, pp. 549–554, 2016.

[38] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information", *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[39] H. Saif, M. Fernández, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of twitter", 2014.

[40] M. Shiha and S. Ayvaz, "The effects of emoji in sentiment analysis", *Int. J. Comput. Electr. Eng.(IJCEE.)*, vol. 9, no. 1, pp. 360–369, 2017.

[41] A. Abbasi, H.-c. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums", *ACM Trans. Inf. Syst.*, vol. 26, Jan. 2008. DOI: 10.1145/1361684.1361685.

[42] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews", *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 760–10 773, 2009. DOI: https://doi.org/10.1016/j.eswa.2009.02.063. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417409001626.

[43] M. Tsytsarau and T. Palpanas, "Survey on mining subjective data on the web", *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 478–514, 2012. DOI: 10.1007/s10618-011-0238-6. [Online]. Available: https://doi.org/10.1007/s10618-011-0238-6.

[44] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval", *Journal of documentation*, 1972.

[45] Y. Bengio, O. Delalleau, and N. Le Roux, "The curse of dimensionality for local kernel machines", *Techn. Rep*, vol. 1258, p. 12, 2005.

[46] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification", *ArXiv*, vol. abs/1607.01759, 2017.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures", *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[49] Z. Huang, W. Xu, and K. Yu, *Bidirectional lstm-crf models for sequence tagging*, 2015. arXiv: 1508.01991 [cs.CL].

[50] J. Mitchell and M. Lapata, "Vector-based models of semantic composition", in *proceedings of ACL-08: HLT*, 2008, pp. 236–244.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[52] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation", vol. 14, Jan. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[53] D. Rozado, "Using word embeddings to analyze how universities conceptualize "diversity" in their online institutional presence", *Society*, vol. 56, no. 3, pp. 256–266, Jun. 2019. DOI: 10.1007/s12115-019-00362-9. [Online]. Available: https://doi.org/10.1007/s12115-019-00362-9.

[54] A. T. Mohan and D. V. Gaitonde, *A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks*, 2018. arXiv: 1804.09269 [physics.comp-ph].

[55] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-entity recognition for indonesian language using bidirectional lstm-cnns", *Procedia Computer Science*, vol. 135, pp. 425–432, 2018.

[56] A. Brunel *et al.*, *A cnn adapted to time series for the classification of supernovae*, 2019. arXiv: 1901.00461 [cs.LG].

[57] Y. Kim, "Convolutional neural networks for sentence classification", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. [Online]. Available: https://www.aclweb.org/anthology/D14-1181.

[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[59] J. He, L. Li, J. Xu, and C. Zheng, *Relu deep neural networks and linear finite elements*, 2018. arXiv: 1807.03973 [math.NA].

[60] A. Jain, A. Fandango, and A. Kapoor, *TensorFlow Machine Learning Projects: Build 13 Real-World Projects with Advanced Numerical Computations Using the Python Ecosystem*. Packt Publishing, 2018.

[61] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. arXiv: 1412.6980 [cs.LG].

[62] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[63] M. E. Peters *et al.*, *Deep contextualized word representations*, 2018. arXiv: 1802.05365 [cs.CL].

[64] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification*, 2018. arXiv: 1801.06146 [cs.CL].

[65] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling", in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649. [Online]. Available: https://www.aclweb.org/anthology/C18-1139.

[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018. arXiv: 1810.04805 [cs.CL].

[67] A. Vaswani *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[68] J. Uszkoreit, "Transformer: A novel neural network architecture for language understanding", *Google AI Blog*, vol. 31, 2017.

[69] K. Bi, R. Jha, W. B. Croft, and A. Celikyilmaz, *Aredsum: Adaptive redundancy-aware iterative sentence ranking for extractive document summarization*, 2020. arXiv: 2004.06176 [cs.CL].

[70] S. Zhang, H. Huang, J. Liu, and H. Li, *Spelling error correction with soft-masked bert*, 2020. arXiv: 2005.07421 [cs.CL].

[71] S. Murty, P. W. Koh, and P. Liang, *Expbert: Representation engineering with natural language explanations*, 2020. arXiv: 2005.01932 [cs.CL].

[72] Z. Bouraoui, J. Camacho-Collados, and S. Schockaert, *Inducing relational knowledge from bert*, 2019. arXiv: 1911.12753 [cs.CL].

[73] C. Sun, X. Qiu, Y. Xu, and X. Huang, *How to fine-tune bert for text classification?*, 2019. arXiv: 1905.05583 [cs.CL].

[74] J. Yu, J. Jiang, L. Yang, and R. Xia, "Improving multimodal named entity recognition via entity span detection with unified multimodal transformer", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 3342–3352. DOI: 10.18653/v1/2020.acl-main.306. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.306.

[75] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach", *arXiv preprint arXiv:1907.11692*, 2019.

[76] Z. Dai *et al.*, "Transformer-xl: Attentive language models beyond a fixed-length context", *arXiv preprint arXiv:1901.02860*, 2019.

[77] Z. Yang *et al.*, "Xlnet: Generalized autoregressive pretraining for language understanding", in *Advances in neural information processing systems*, 2019, pp. 5753–5763.

[78] Z. Lan *et al.*, "Albert: A lite bert for self-supervised learning of language representations", *arXiv preprint arXiv:1909.11942*, 2019.

[79] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer", *arXiv preprint arXiv:2001.04451*, 2020.

[80] S. Dadas, M. Perełkiewicz, and R. Poświata, *Pre-training polish transformer-based language models at scale*, 2020. arXiv: 2006.04229 [cs.CL].

[81] J. Cañete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish pre-trained bert model and evaluation data", in *to appear in PML4DC at ICLR 2020*, 2020.

[82] A. Conneau *et al.*, *Unsupervised cross-lingual representation learning at scale*, 2019. arXiv: 1911.02116 [cs.CL].