Automatic personality estimation from text in two languages using natural language processing techniques

Felipe Orlando López Pabón

Tesis de maestría presentada para optar al título de Magíster en Ingeniería de Telecomunicaciones

Director
Prof. Dr.-Ing. Juan Rafael Orozco Arroyave

Universidad de Antioquia
Facultad de Ingeniería
Maestría en Ingeniería de Telecomunicaciones
Medellín, Antioquia, Colombia
2022

Cita	López Pabón [1]
Referencia **Estilo IEEE (2020)**	[1] F. O. López Pabón, "Automatic personality estimation from text in two languages using natural language processing techniques", Tesis de maestría, Maestría en Ingeniería de Telecomunicaciones, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.

Maestría en Ingeniería de Telecomunicaciones, Cohorte XV
Grupo de Investigación Telecomunicaciones Aplicadas (GITA)

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia
Sistema de Bibliotecas
Biblioteca Carlos Gaviria Díaz

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes
Decano/Director Jesús Francisco Vargas Bonilla
Jefe departamento: Augusto Enrique Salazar Jiménez

Automatic personality estimation from text in two languages using natural language processing techniques

UNIVERSIDAD DE ANTIOQUIA

1 8 0 3

Research work for the Master's degree in Telecommunications Engineering.

Felipe Orlando López Pabón

Director: Prof. Dr.-Ing. Juan Rafael Orozco Arroyave

Faculty of Engineering

Department of Electronic Engineering and Telecommunications

University of Antioquia

Acknowledgments

I am mainly grateful to my mother Luz Marina Pabón Tellez and my father José Ramiro López López, who, with unconditional love and understanding, are responsible for giving me the basis and everything necessary to obtain this great achievement, besides the motivations and the great example that each one of them has given to me. To my brothers Camilo and Cesar, who with dedication have accompanied me on this long road and advised me on more than one occasion on what was best. To my wife Adriana and my daughter Antonella, who have been my daily motivation to achieve this dream. To all my family, because at some point I needed their support.

I thank my friends and colleagues Cristian Rios and Daniel Escobar, who, on more than one occasion, extended their hand to help me both in academic and personal areas, and who also helped me in the development of this work. I would also like to thank my colleagues in the GITA research group, Luis Felipe Gómez, Paula Pérez, Luis Felipe Parra, Sebastian Guerrero, Tomas Arias and Guberney Muñeton, who, with patience, were able to help me in everything I needed.

I would like to thank my director Juan Rafael Orozco and my co-advisor Juan Camilo Vasquez, because thanks to them, the development of this work was possible, who, with theirs advices and knowledge guided me and helped me to understand that this academic goal could be achieved with a lot of effort.

Finally, I am also grateful to the Faculty of Engineering that gave me the *Estudiante Instructor* scholarship, since it covered most of my expenses in the master's program and gave me an economic support during all this time. Also, thanks to the CODI UdeA, grant # PRG2020-34068.

Contents

Chapter 1

Introduction

1.1 Motivation

Personality plays an important role in human interaction and it is defined as the combination of several behavioral characteristics, emotions, motivation, and thinking patterns of an individual [1], [2]. Personality not only reflects the consistent patterns of behavior, thinking and interpersonal communication, it also influences important aspects of life, including happiness, motivation to address tasks, preferences, emotions and mental-physical health [3], [4]. The automatic analysis of personality has gained attention and has grown a lot in the lasts years, so there exist different fields of application connected to it, including health and education [5]. In marketing, automatic personality modeling enables the prediction of preferences to improve the effectiveness in recommendation systems [6], [7]; in sentiment analysis, opinion mining and author profiling [8]; and even the analysis of marital happiness [9]. All of the above can be achieved by obtaining predictions in a more agile, fast and reliable way through artificial intelligence models that can help researchers and service providers to improve the personalized offer of products and services.

Researchers in psychology have studied how individuals differ, trying to find a general method to classify human behavioral traits into different categories. One of the most widely used models for automatic personality analysis is the Big Five model, also called the OCEAN model [10]. According to it, personality is assessed by five dimensions described as follows: **O**penness to experience: creative, curious vs. rigid, closed-mind; **C**onscientiousness: efficient, self-disciplined vs. lazy, irresponsible; **E**xtraversion: sociable, energetic vs. shy, quite; **A**greeableness: friendly, cooperative vs. selfish, unkind;

Neuroticism (the opposite of Emotional stability): insecure, nervous vs. stable, confident.

Traditional methods of personality assessment through questionnaire research or interviewing experts are expensive and less practical in cyberspace. In addition, it also has limitations in terms of participant recruitment, feedback efficiency and resource consumption [11]. For this reason, recent research has been focused on personality recognition using texts available on the web, especially on text from social media. However, since most approaches found in literature review focus on linguistic features that rely heavily on human intervention (e.g. dictionary-based features) and do not seem to take full advantage of the rich information in texts; the results obtained from most methodologies can be improved if they are allowed to use contextual information and models that can deal with it automatically, without relying on humans using dictionaries. Therefore, much can be contributed to the task of personality recognition by taking into account contextual information and word order to capture meaningful syntactic and semantic features when modeling user texts, since, according to the work done in [12], open vocabulary methods (such as those considered in this work - Word2Vec, GloVe, BERT, BETO), instead of relying on a prior judgment of words or categories, are based on the extraction of a complete collection of linguistic and contextual features from the text. These methods characterize the sample text through uncategorized single words, multi-word phrases, and semantically related word clusters identified by unsupervised methods.

Apart from the aforementioned, in most of the works that consider machine learning methods for personality assessment through user-generated content, ground-truth labels are usually obtained by asking the participants to take a survey that measures the personality traits of the Big Five model. This approach is a costly and time-consuming task, and faces privacy issues. This is why there is a bottleneck in the study of language models to perform personality prediction, which is reflected in the scarcity of large datasets appropriately labeled. Now, on the other hand, due to the scarcity of works considering languages other than English, there is a growing interest in the scientific community to develop automatic personality models in other languages as for example in Brazilian Portuguese [13], Italian [8], [14], Dutch [15], [16] and Spanish [8], [17]–[21]; in order to generate personality models and to take a step forward in the automatic evaluation of customer preferences according to their traits and also to support the processes fol-

lowed to design new products and services.

1.2 State-of-the-art

The study of systems that allow automatic personality recognition has gained attention in the last decade. After a detailed study of the state of the art, the methods that have been used can be divided into 3 types: 1) Knowledge-based Detection (KbD) methods, which use a combination of Keyword-Based methods and affective emotion lexicons. 2) Machine Learning (ML) methods and 3) Deep Learning (DL) methods. Before explaining some of the works based on the aforementioned methods, it is important to highlight which databases have been most commonly used. First, it is the *James Pennebaker and Laura King's essay dataset* [22], which consists of 2468 essays written by college students about their daily life and their thoughts, labeled with the Big Five traits. The *MyPersonality dataset* [23], which consists in approximately 10K status updates written by 250 Facebook users and five binary labels for personality traits in the OCEAN model. Another dataset consists of Twitter data [8], [24], which includes status updates, demographic information of the users (such as age and gender) and personality labels. Finally, there is the *YouTube Personality dataset* [25] that includes the transliterations of YouTube video blogs (vlogs) that were tagged with the Big Five personality scores using crowdsourcing, where participants talked about different topics such as personal issues, food and movies. The revision presented below shows not only works according to the methods they use, but also includes a couple of sections showing which are the most relevant works based on the aforementioned corpora.

1.2.1 Knowledge-based Detection (KbD) methods

Within the KbD methods, Fabio Celli et al. [14] presented personality models and classified the 5 personality traits of the OCEAN model. The authors used a corpus that consists of posts from status updates provided by a total of 748 users of the social network *FriendFeed*. Different linguistic features were considered including punctuation marks, commas, first person singular pronouns, negative adverbs, parenthesis, positive and negative emotions, prepositions and pronouns, among others. The authors reported an average accuracy of 63.1% for the OCEAN traits. In [26], the authors classify also

the traits of the OCEAN model using MyPersonality dataset. The presence or absence of each trait was labeled with a binary variable. The reported average accuracy was 64% when using semantic similarity measures based on the WordNet ontology and National Research Council Canada (NRC) affect lexicon, taking into account also morphological information like the use of nouns, adjectives, verbs and adverbs. In a similar work, [13], using Bag of Words (BoW), Psycholinguistics (obtained with Linguistic Inquiry and Word Count - LIWC toolkit), Word2Vec (Continuos BoW and Skipgram embedding models with size 600), Doc2Vec and LSTM-600 (Keras embedding model) features; they develop a system that recognizes the personality of 1039 Facebook Brazilian users with the implementation of Random Forest (RF) classifier. They obtained a highest F1-score percentage of 61% for extraversion trait and a average F1-score of 58.4% for the five traits. They conclude that no single model is capable of providing the best results for all five classes; which may suggest that not all personality traits are equally accessible from text. They also notice that the models based on word embeddings seem to outperform those based on lexical resources.

1.2.2 Classic Machine Learning (ML) methods

Regarding ML methods, in the work of Daniele Quercia, et al. [27], they predict the 5 personality traits of 335 Twitter users performing a regression analysis with a 10-fold cross-validation with 10 iterations using the Decision Trees (DT) algorithm. The authors used social network features such as the number of profiles the user follows (following), number of followers, number of times the user has been listed in others' reading lists and measure the performance of their system with the Root Mean Square Error (RMSE), where the average result for the five traits was RMSE=0.79, and the lowest RMSE value was RMSE=0.69 for the openness to experience trait. In another work, [28], they make use of the Term Frequency - Inverse Document Frequency (TF-IDF) feature and the Naive Bayes (NB), K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers for the task of personality classification in texts from social networks such as Facebook (250 users) and Twitter (40 users), obtaining the following results: for Facebook data, they achieved 60% average accuracy while for Twitter texts, the average accuracy is up to 65%. Similarly, in [11], they analyze some of the data from MyPersonality dataset and measure the performance with the F1-score.

They use the following classification methods: KNN, NB and DT with different groups of features, among which are time features (e.g. frequency of status updates per day), social network features (e.g. network size), NLTK features (e.g. frequency of adjective), statistics features (e.g. the time of first status posted per day), text style features and TF-IDF-based psychological features. They find that TF-IDF-based psychological features and text style features are helpful for classifying personality traits. The results also show that the Particle Swarm Optimization (PSO) algorithm can extract better combination of features, which contributes to high performance on their model, and the best F1-score value of the personality recognition has reach up to 79% for openness to experience trait using KNN classifier and an average F1-score of 74% for the five traits.

1.2.3 Deep Learning (DL) methods

Among the DL methods is the work of Navonil Majumder et al. [29], where the James Pennebaker and Laura King's essay dataset [22] were studied. They trained with document-level stylistic features (e.g. word count and average sentence length) and per-word semantic features (Word2Vec embeddings per word with size 300) different Convolutional Neural Networks (CNNs) and Multiple-Layer Perceptron (MLP) as binary classifiers that predicted the corresponding trait to be positive or negative (presence or absence of the trait). They reached a highest accuracy percentage of 62.68% for openness to experience trait and obtained an average accuracy percentage of 58.83% for the five traits. Related to text posts from Facebook, the work done in [30] used five different sets of features: semantic feature set extracted from a CNN-based deep neural network (which they call "Cnn"), deep semantic features extracted from the RCNN-CNNs architecture (RCC), another kind of document-level semantic feature vectors extracted through the unsupervised Doc2Vec algorithm (D2V), their own deep semantic representations (named "ARCC") and Statistical Linguistic features (SL). They implemented several machine learning methods and their best results were the following: for openness to experience and agreeableness traits, they get a Mean Absolute Error (MAE) of 0.358 and 0.386 respectively, considering only the ARCC feature set and making use of Support Vector Regression (SVR). For conscientiousness and extraversion traits, the MAE is 0.425 and 0.478 taking into account ARCC + SL feature sets and using Gradient

Boosting Regression (GBR). For neuroticism trait, MAE is equal to 0.487 taking into account ARCC + D2V + SL features sets and using GBR as well. Finally, the best average MAE (for the 5 traits) is 0.428, obtained with ARCC + SL features sets and using GBR. In [31], they analyze the essays from [22] using the pre-trained contextual embeddings obtained from the Bidirectional Encoder Representations from Transformers (BERT) and RoBERTa, and also using different linguistic features obtained with LIWC. They tested with several types of neural networks: HCNN(Hierarchical CNN model), ABCNN and ABLSTM, which represents CNN and Bidirectional LSTM models with attention mechanism and also HAN, a Hierarchical Attention Network. They conclude that compared with LIWC-based models and different Neural Networks (HCNN, ABCNN, ABLSTM), they improved the performance approximately 2.5% for 5 traits on average using BERT embeddings (agreeableness by 2.2%, conscientiousness by 2.8%, extraversion by 2.5%, openness to experience by 3.1% and neuroticism by 1.6%). With RoBERTa, they achieved the best accuracy percentages for four of the five traits: 59.7% for agreeableness, 60.6% for extraversion, 65.9% for openness to experience and 61.1% for neuroticism; and for the remaining trait, conscientiousness, the best accuracy percentage was of 60.1% with ABCNN neural network.

1.2.4 Works with YouTube Personality dataset

Besides the studies mentioned above, there are works where transliterations obtained from YouTube videos are considered as the input to the model to evaluate different personality traits. As our work is focused on the automatic evaluation of personality traits based on the transliterations provided in [25] (see subsection 3.1 for more details) we are going to mention some works related to this database. In [32], they considered each trait as a separate bi-class problem (i.e., they performed the automatic classification of presence vs. absence for each trait). Their model was based on uni-gram BoW and TF-IDF features and the classification was performed with a Logistic Regression (LR) classifier. The average F1-score reported for the OCEAN traits was 60.1%, and the highest value was obtained for the agreeableness trait (65.8%). A similar study was presented in [33], where the best result was obtained with Part-of-Speech (PoS) tagging features and the classification was performed using an SVM for each trait separately. In this case, the authors reported an

average F1-score of 60.2% for the five traits in the OCEAN model, and the highest F1-score was 69.6% for agreeableness trait.

In [34], they considered transliterations from the same dataset and used 69-dimensional LIWC vectors to represent the texts; which consists of counts the number of anger, sad, pronoun, positive emotion and negative emotion words. The authors tested popular classification algorithms like SVM, MLP, LR; and reported and average accuracy of 62.3% when they classified the different traits of the OCEAN model. Later, in [35], the authors started to approach the problem of personality detection based on unsupervised learning methods. The authors reported RMSE values of 0.68, 0.69, 0.89, 0.77, and 0.69 for OCEAN traits respectively. More recently, also working with unsupervised methods based on the skip-gram algorithm, the authors in [36], reported MAE values of 0.58, 0.57, 0.72, 0.67, and 0.60 for the same traits. In the same year, also working upon the same dataset with transliterations from YouTube vlogs, the authors in [37] considered 300-dimensional embedding vectors obtained from the GoogleNews Word2Vec pre-trained model. The authors created a neural network architecture that combined convolutional and recurrent layers to perform the classification of the traits. They obtained an average F1-score of 54.7% for the OCEAN traits, with the highest percentage of F1-score of 71.9% for extraversion trait and the lowest of 40.3% for neuroticism trait.

1.2.5 Works with PAN-AP-2015 dataset

With respect to the database proposed in [8] (see more details in subsection 3.2), since these will be one of the datasets that we will use later on for the experiments, we will mention some of the works performed with the English and Spanish language. The best result for personality traits estimation in English was obtained by the work in [38], where they combined thematic information features (obtained with Latent Semantic Analysis (LSA)) with stylistic textual features (obtained with Second Order Attributes (SOA)) in order to obtain relationships among terms, documents, profiles, and subprofiles. They used algorithms based on SVM and LR, and obtained an average RMSE value of 0.144 for the OCEAN traits. For each of the traits, the RMSE values were 0.120, 0.117, 0.128, 0.131 and 0.225 respectively. Now, the best result for the texts in Spanish was obtained by the work in [17]. In this case, the authors used the 200 most frequent terms (words and punctua-

tions) as features and used it to create personality models for each one of the traits. They reported an average RMSE of 0.123 for the OCEAN traits and for each one of the traits 0.111, 0.102, 0.137, 0.103 and 0.164 respectively.

Other works also related to PAN-AP-2015 dataset are the following: in [18], where using stylistic features such as percentage of question sentences, average sentence length, percentage of punctuations, percentage of comma, among others; and machine learning algorithms such as NB, RF, SVM and LR, they obtained for the OCEAN traits an average RMSE values of 0.231 and 0.212 for text in English and Spanish respectively. Similarly, in [19], they use TF-IDF features based on LIWC dictionaries that includes words related to linguistic dimensions (e.g., swear words), psychological processes (e.g., anger words), relativity (e.g., verbs in the past and future tense), personal concerns categories (e.g., occupation such as job) and so on. To predict the labels in the OCEAN traits, the authors used the Ensemble of Regressor Chains Corrected (ERCC) multivariate regression model, which is a multivariate technique that allows to take advantage of the prediction result for one personality trait to make a prediction for another. The reported average RMSE value for English and Spanish was 0.171 and 0.182 respectively.

And finally, as a last work related to DL methods with data from Twitter, in [39], where only analyzed the data in English, through different architectures of CNNs, they predicted the five personality traits of the OCEAN model. They use 25 and 50-dimensional pre-trained Global Vectors (GloVe) word embeddings to represent the whole tweet as an image, where each tweet was represented using a matrix $X \in m \times n$, with m the maximum number of words in the tweets and n the word embeddings dimension. They achieved RMSE values of 0.148, 0.144, 0.158, 0.150, 0.212 for OCEAN traits respectively, obtaining an average RMSE of 0.162.

1.3 Objectives

1.3.1 General Objective

To analyze, implement and evaluate machine learning and deep learning models that allow automatic personality evaluation in text signals making use of Natural Language Processing techniques.

1.3.2 Specific Objectives

- ✓ To explore and to adapt Natural Language Processing techniques that allow the extraction of relevant features to distinguish personality traits in texts.
- ✓ To study and to implement different methods of machine learning and deep learning that allow the automatic recognition of personality based on the previously described features.
- ✓ To measure and to evaluate the performance of the automatic personality recognition system using standardized metrics according to the literature.

1.4 Research Problem

Given the works mentioned in section 1.2, and also taking into account that personality is a trait that can be effectively modeled by different biosignals including facial expression, speech, language and others; this work is focused on extracting information from language signals to model personality according to the big-five criteria defined on psychology, with the caveat that, due to the nature of the two databases taken into account, the personality trait Emotional Stability was considered since both databases originally come with this label instead of the label for the Neuroticism trait (remember that, as mentioned in section 1.1, the trait Emotional Stability is the opposite trait of Neuroticism, which would mean that a high score in Emotional Stability would imply a low score in Neuroticism; and vice versa). In order to optimize the time, and decrease the resources needed to perform the personality evaluation, we intend to develop a system that allows to automatically recognize a person's personality through text signals. The idea is that, based on different types of vector representation of the texts that capture syntactic and semantic relations, known as word embeddings, each text can be characterized as a fixed (based on statistics from embeddings) length dimension vector, so that, later, using techniques of machine learning and deep learning systems, each one of the traits of the Big Five Model can be estimated automatically and also to classify the level of presence of the traits in the text.

1.5 Contribution of the research work

According to the literature review, there is a lack of works in the field of automatic personality recognition based on machine learning or deep learning models using word embeddings that do not rely on dictionaries or lexicons that are highly dependent on human intervention. Similarly, there is a lack of works using texts in a language other than English. This study is focused on the use of natural language processing techniques that allow the extraction of word embeddings that are useful for the estimation of the 5 personality traits defined in the OCEAN model of psychology (remember that the Emotional Stability trait is considered instead of the Neuroticism trait) in two languages: English and Spanish. The main contribution of this work includes: 1) different experiments are explored: i) regression methods, to predict the personality scores on the traits, classification methods, such as ii) two-class classification: weak vs. strong presence of each trait, and iii) three-class classification: low vs. medium vs. high presence of each trait; 2) implementation of word embeddings based on classical methods: Word2Vec and GloVe as well as word embeddings based on state-of-the-art methods such as BERT and BETO to train machine learning methods; 3) use of deep learning methods that allow to extract word embeddings from texts and train the embeddings layer from scratch or use pre-trained embeddings to improve the performance of the architectures; and 4) evaluation of the different methods in Spanish language taking into account text signals coming from YouTube and Twitter.

1.6 Structure of the research work

Chapter 2: Describes the methods used for the estimation of personality traits. In addition, it describes models based on natural language processing techniques that allow characterizing texts and predicting the scores for the traits belonging to the OCEAN model.
Chapter 3: Contains information about the databases used in this work: transliterations of YouTube videos and Twitter posts. It also includes information about the labels on each of the five traits of the OCEAN model.
Chapter 4: Describes the methodology implemented for the evaluation of personality traits. It also includes the methods of validation and optimization of parameters, as well as the explanation of the metrics used to measure the

performance of the models.

Chapter 5: Contains the explanation of data distribution and statistical analyses. Also, in this chapter we describe the details of the experiments performed in this work.

Chapter 6: It presents the results obtained for the different experiments. It also shows the performance of the models in English and Spanish language, taking into account machine learning and deep learning methods.

Chapter 7: It includes the conclusions based on the analysis of the results obtained in this work and mentions the line of future work.

Chapter 2

Theoretical background

Humans have learned to communicate through some form of language, whether by text or voice. Now, in order to perform human-computer interactions, computers need to understand the natural languages used by humans. Natural Language Processing (NLP), which is a field of artificial intelligence, is concerned with enabling computers to analyze, understand, manipulate and process natural languages in an intelligent and useful way. In a general way, the NLP steps that we took into account consist of: i) data pre-processing, ii) feature extraction and iii) training and optimization of classical or deep machine learning models.

Data pre-processing consists of a data mining technique that allows transforming raw data into an understandable format; in other words, it is the process that allows eliminating non-relevant information (noise) so that the model can learn. Feature extraction consists of representing texts in their equivalent numerical form so that syntactic and/or semantic relationships are preserved, so that this information can be entered into classical or deep machine learning models, since similar to computers that only understand binary digits of 0 and 1, such models tend to understand only numerical vectors or matrices. Once the features are obtained, the idea is to train and optimize a classical or deep machine learning models in order to gain knowledge of the data and generate a prediction according to the labels or targets. The following sections of this chapter will explain the theoretical details of the techniques, algorithms and methods that were considered for the preprocessing, feature extraction and training/optimization of the classical or deep machine learning models.

2.1 Pre-Processing

The first step to achieve the objective of the automatic personality evaluation through texts signals, consists in cleaning and standardizing the texts to avoid noise and getting them ready for analysis; so that once they are ready, the feature extraction can be implemented.

2.1.1 Pre-Processing of the data from YouTube

In the transliterations that come from YouTube videos, there were words like “xxxx”, “um”, “uh”, which were removed since they were only words to anonymize the data and also proper to the transcription to make it as reliable as possible. In English language, the steps to pre-process the data are: all the text is converted to lower case, then, all the punctuation present in the text is also removed, same for the numbers and non relevant information for the context, such as stopwords (meaningless words such as articles, pronouns, prepositions, etc), and finally, given the fact that there are multiple representation for a single word, it was necessary to standardize the words in an equal representation, so Lemmatization was applied to transform the words into their root form. In Spanish language, the procedure is very similar than in English, only that an extra process is added before removing stop words, which consists of removing accents (very common in Spanish).

2.1.2 Pre-Processing of the data from Twitter

The pre-processing of the data in this case is very similar to the previous one, only with some differences. In this case, both for texts written in Spanish and English, we proceed as follows: texts were lowercased and all hastags (#), mentions (@), urls, punctuation marks, emojis and numbers were removed.

2.2 Feature extraction

This process consists of creating numerical representations for the texts that allow to represent them in a vector space. The most typical techniques used in the literature are Word2Vec, GloVe, BERT, BETO (a Spanish version of BERT) and Keras Embedding Layer. All these techniques intend to create word embeddings to represent texts. Embeddings are numerical vectors with

a fixed length that keep information about coexisting words. Details of each technique are presented below:

2.2.1 Word2Vec

Word2Vec allows to represent words as a vector in a multidimensional space, where similar or related words are represented by nearby points. The model considers a single hidden layer neural network whose values encode the word representation. Such representation can be obtained using two methods: Skip Gram and Continuous Bag of Words (CBoW) [40], [41]. The network contains a hidden layer whose dimension is equal to the embedding size. At the end of the output layer, a softmax activation function is applied so that each element of the output vector describes the probability that a specific word appears in the context. The input to the neural network consist in vectors coded in the *one-hot* form, which is a vector with only one target element as 1 and the others as 0. An example of *one-hot* encoding representation is shown in Figure 2.1, where the vocabulary corresponds only to the set of words describing pets, "Cat", "Dog", "Fish", "Turtle". As can be seen in the right part of the image, for each of the words a 4-dimensional vector is generated that corresponds to the one-hot encoding of the word. Thus, one-hot vectors for words starting with "a" are expected to have the target "1" at a lower index, while those for words starting with "z" have the target "1" at a higher index. In the example mentioned above, the word "Fish" would be encoded as the *one-hot* vector 0010.

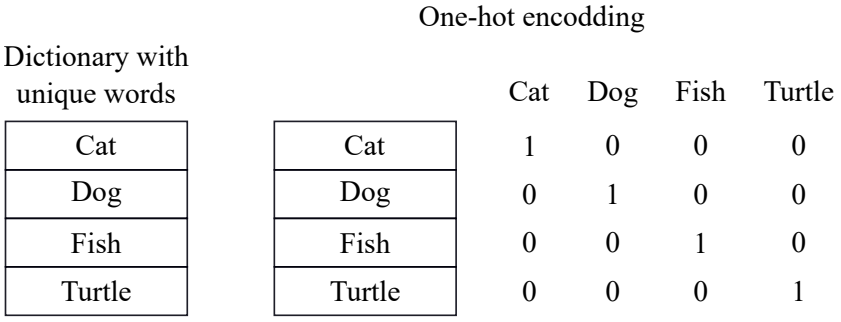

Figure 2.1. Example of *one-hot* encoding representation.

Figure 2.2 shows the structure of the network in Word2Vec for both CBoW and Skip Gram methods. For CBoW method, the model takes the context of each word as input and tries to predict the corresponding target

word according to the context. For example, in the sentence "I have a cute dog" the input would be "I", "have", "cute" and "dog", while the output will be "a", assuming a window size (context) of 2. In the case of Skip Gram, the process is very similar to CBoW, except that it exchanges input and output, i.e. the input is the target word, while the outputs are the words around the target word. For example, taking the same sentence mentioned above and taking "a" as the target word, the input would be "a", while the output would be "I", "have", "cute" and "dog", assuming that the size of the window is still 2.

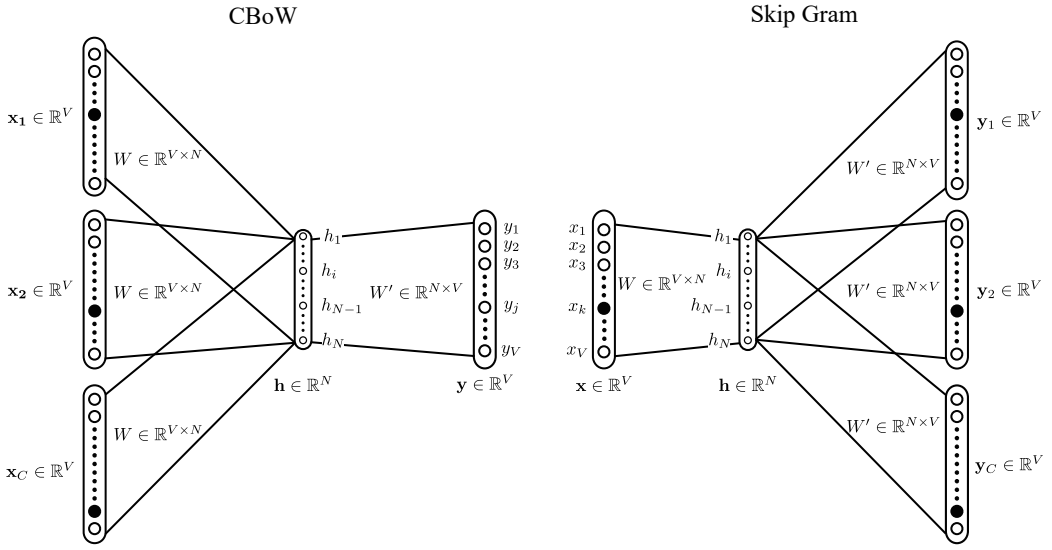

Figure 2.2. Topology of models used in Word2Vec. $W \in \mathbb{R}^{V \times N}$: weight matrix that maps the input $\mathbf{x}$ to the hidden layer. $W' \in \mathbb{R}^{N \times V}$: weight matrix that maps the hidden layer outputs to the final output layer. $\mathbf{x}$: Vector in *one-hot* format, $\mathbf{h}$: Hidden layer of N neurons. V: size of the vocabulary. C: number of context words. Figure adapted from [42].

2.2.2 Global Vectors (GloVe)

GloVe is an unsupervised learning algorithm that allows to obtain vector representations of words by capturing local and global statistics by studying the co-occurrence of words in a corpus [43]. Given a corpus having V words, the co-occurrence matrix X will be a $V \times V$ matrix, where the i-th row and the j-th column of X, X_{ij}, denotes how many times the word i has co-occurred with the word j. Once X is created, the task is to generate the

vectors in a continuous space for each word of the corpus. Vectors with a smooth constraint will be produced for each pair of words ($\boldsymbol{w_i}, \boldsymbol{w_j}$):

$$\vec{w_i}^{\mathsf{T}}\vec{w_j} + b_i + b_j = \log(X_{ij}) \tag{2.1}$$

Where $\boldsymbol{w_i}$ and b_i are the word vector and bias respectively of the i-th word, and $\boldsymbol{w_j}$ and b_j are the word vector and bias respectively of the j-th word. In Equation 2.2 can be seen the weighted mean square loss function J that GloVe implements, which minimizes the difference between the dot product of the vectors of two words and the logarithm of their co-occurrence value:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\vec{w_i}^{T}\vec{w_j} + b_i + b_j - \log(X_{ij}))^2 \tag{2.2}$$

$f(X_{ij})$ is a weighting function such that assigns lower weights to rare and frequent co-occurrences values as can be seen in Equation 2.3.

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_{\max}}\right)^{\alpha}, \text{if } X_{ij} < x_{\max} \\ 1, \text{in other case.} \end{cases} \tag{2.3}$$

Where $x_{\max}$ refers to the maximum co-occurrence value that the i-th word has with the j-th and α is a hyper-parameter that controls the sensitivity of the weights to increased co-occurrence counts.

2.2.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT makes use of the Transformer paradigm, an attention mechanism that learns contextual relations among words (or sub-words) in a text [44]. In its general form, a Transformer includes two separate mechanisms: an encoder that reads the text input and a decoder that produces a prediction for the task. BERT allows both left and right contexts to have influence in many language representations that include word predictions [44]. To effectively train a bidirectional transformer, BERT uses two techniques called Masked Language Model (MLM) and Next Sentence Prediction (NSP).

The training process with MLM is as follows: (1) 15% of the words at the input ($\mathrm{w_n}$) in each sequence are replaced with a named [MASK] token,

with the aim to predict the original value of the masked words, based on the context provided by the other non-masked words in the sequence; (2) a classification layer is added on top of the outputs (O_n) of the Transformer encoder; (3) the output vectors are multiplied by the embedded matrix $\mathbf{w}'$ which columns are given by w′ in Figure 2.3, transforming them into the vocabulary dimension, (4) the model computes the probability of each word in the vocabulary with a softmax activation function. Figure 2.3 shows the process.

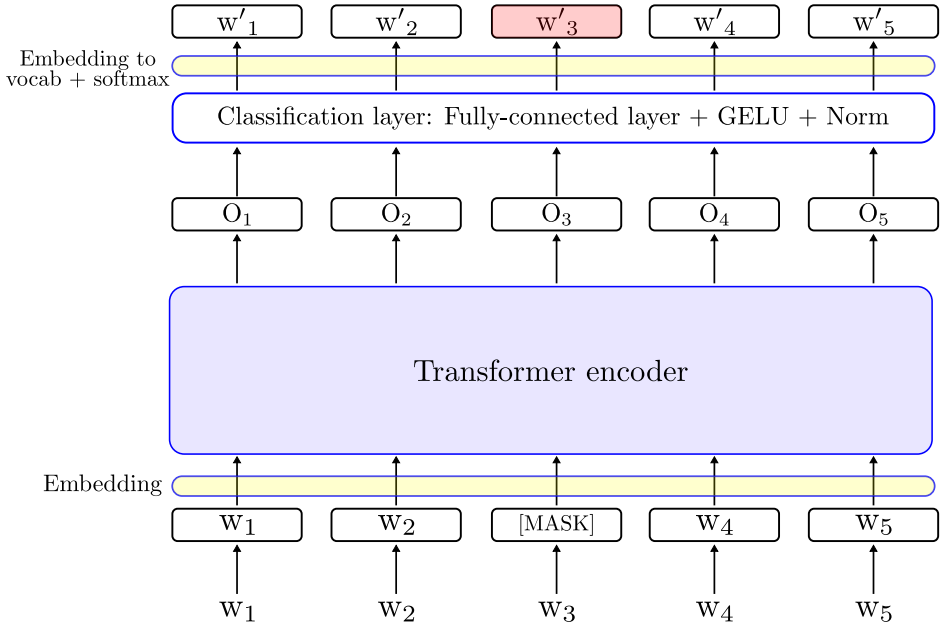

Figure 2.3. MLM proccess when training BERT. Figure adapted from [45].

Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two sentences, which is not directly captured by Language Modeling [44]. To train a model such that understands sentence relationships, authors of BERT pre-trained a binarized NSP task that can be trivially generated from any monolingual corpus. NSP works in the BERT training process using pairs of sentences. Let sentences A and B be the inputs. NSP detects whether sentence B is next to sentence A in a document, and helps the model to differentiate between two sentences. It works as follows: (1) A [CLS] token is inserted at the beginning of the first sentence (in this case sen-

tence A) and a [SEP] token is inserted at the end of each sentence (sentence A and sentence B); (2) a sentence identifier is created to identify words from both sentences A and B; (3) a positional embedding is added to each token to indicate its position in the sequence; (4) the complete sequence is parsed inside the Transformer; (5) a classification layer predicts the probabilities of next sentence. Figure 2.4 shows how the processing of the sentences is performed before being parsed inside the Transformer. When training the BERT model, MLM and NSP are trained together, with the goal of minimizing the combined loss function of the two strategies [45].

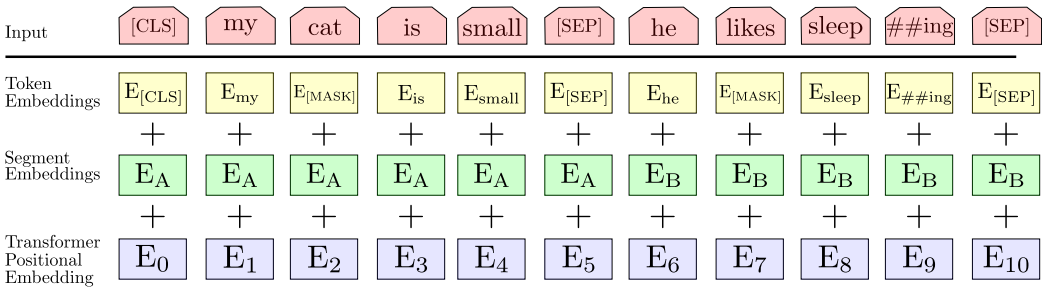

Figure 2.4. Example of BERT input representation using NSP. Figure adapted from [44].

The Transformer architecture is composed of a stack of encoders and a stack of decoders, where the encoders are composed of a Self-Attention layer and a Feed-Forward Neural Network (FFNN). Encoders are identical in structure and are connected to decoders, which include all the elements present in an encoder, and additionally, they have an Encoder-Decoder Attention layer between the Self-Attention layer and the Feed Forward layer. Figure 2.5 shows the architecture of the Transformer in BERT.

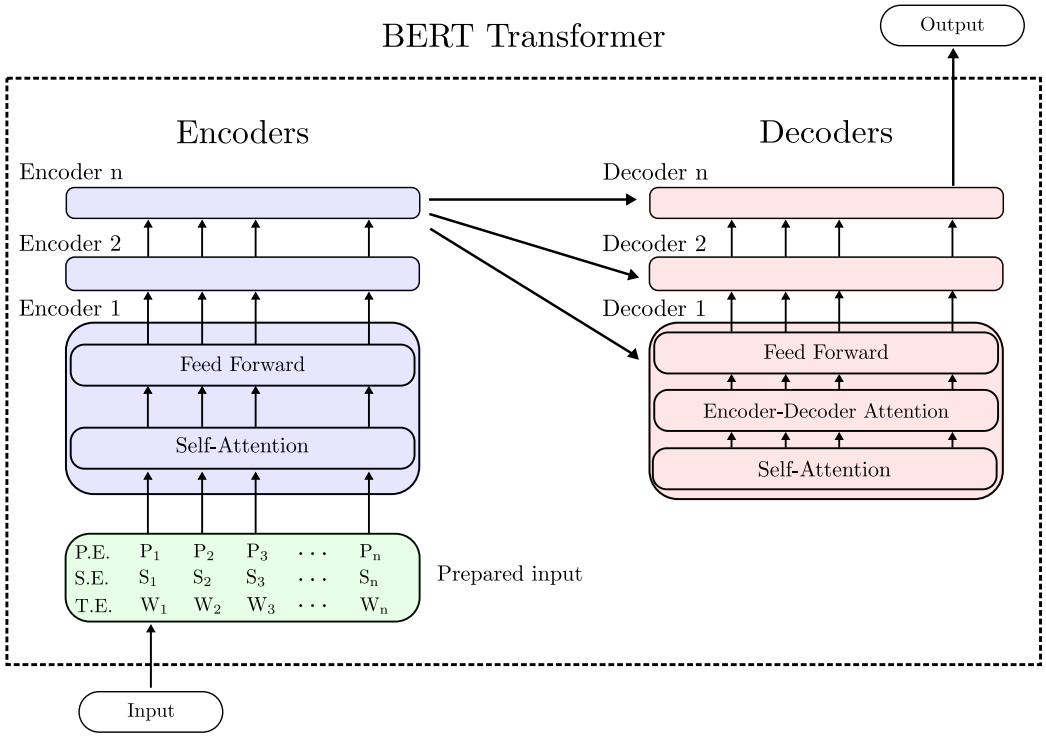

Figure 2.5. Architecture of the Transformer used in BERT.**T.E:** Text Embedding, **S.E:** Segment Embedding, **P.E:** Positional Embedding. Figure adapted from [46].

2.2.4 Keras embedding layer

Keras provides an embedding layer that can be used for neural networks on text data. This layer requires the input data to be encoded with integers, so that each word is represented by a single integer. This layer also requires that all individual documents have the same length. Therefore, shorter documents are normally padded with 0 values.

The embedding layer (which is defined as the first hidden layer of a network) is initialized with random weights and will learn an embedding for all words in the training dataset. Four arguments must be specified: *input_dim*, which is the size of the vocabulary in the text data. For example, if the data is encoded in integers with values between 1 and 100, the vocabulary size would be 100 words. *output_dim*: defines the size of the output vectors of this layer for each word. *input_length*: which corresponds to the length of the input sequences. For example, if all documents are composed of a

fixed length of 50 words, this would be 50. And finally, *mask_zero*, which is a Boolean value and indicates whether or not the input value 0 is a special "padding" value to be masked. Since the value 0 is previously used to padd shorter documents, the value *mask_zero* should be set to True. The output of the embedding layer is a 2D vector with one embedding for each word in the input document.

This layer can be used in several ways, such as: 1) alone to learn a word embedding that can be saved and used in another model. 2) it also can be used as part of a deep learning model where the embedding is learned along with the model itself. And finally, 3) it can be used to load pre-trained word embedding models (for example Word2Vec and GloVe), which is a type of transfer learning [47].

2.2.5 Word embedding models

Model training for Word2Vec and Glove

For English Language, pre-trained models were used with the python gensim module [48]. In the case of Word2Vec, it was used the "word2vec-google-news-300" model, which was trained with Google news, with a corpus of around 100 billion words and has a vector dimension of 300 [49], [50]. For GloVe, it was used the model "glove-wiki-gigaword-300", which was trained with the corpus "Wikipedia 2014 + Gigaword 5", in this case, there are 5.6 Billion tokens and a vocabulary of size 400K and the vector dimension is also 300 [50], [51].

In the case of Spanish Language, we trained our own two models with embeddings with dimension 300 using a machine with 256GB of RAM memory, 96 processing cores and making use of the python gensim module [52]. Both models (Word2Vec and GloVe) were trained with the Spanish language Wikipedia 2018 Corpus, which contains approximately 709 million words [53]. For the Word2Vec model it was used Skip-Gram method and a window size of 8. In the case of GloVe model, it was used a window size of 7 and a value of $\alpha = 0.75$, which is a default hyper parameter that controls the sensitivity of the weights to increased co-occurrence counts. The choice of window size and embedding dimension in this way is made because according to the state of the art, these values in the parameters brings some of the best results in similar tasks like the one we are addressing in this, which involves semantic questions (typically analogies about people or places, such as "Madrid is

to Spain as Berlin is to ?") and also in tasks involving syntactic questions (analogies about verb tenses or adjective forms, for example "play is to playing as read is to ?"). We took into account a minimum count of 5 occurrences for each word, to avoid not relevant words be included [43], [54].

Pretrained models for BERT and BETO

To obtain word embeddings based on BERT, we used the WEBERT Toolkit [55], which is a Python tool typically used to obtain BERT embeddings in English and Spanish language. For BERT embeddings in Spanish, the translated version to Spanish of the corpus named Multi-Genre Natural Language Inference was used. The same framework was used to extract the BETO embeddings, which is a pre-trained BERT model that used a Spanish corpus named Spanish Unannotated Corpora [56]. Both BERT and BETO embeddings are 768-dimensional, we took this dimension because, despite it exists a large BERT model that brings 1024-dimensional embeddings, taking large BERT model this would have taken more computational cost and a direct comparison with BETO could not be made.

Because the word embeddings are calculated per word, and the texts correspond to spontaneous narrations (which means that the number of words is different for each subject), it was decided to obtain a fixed dimension vector for each subject. This was performed by taking six functional statistics from the word embeddings: mean, standard deviation, skewness, kurtosis, minimum and maximum. In summary, these are the resulting feature matrices that we took into account: with respect to **Word2Vec** and **GloVe** embeddings, each one with $\mathbf{X} \in \mathbb{R}^{n\times 1800}$; **Fusion** embeddings, which are the simple concatenation of Word2Vec and GloVe embeddings, resulting in $\mathbf{X} \in \mathbb{R}^{n\times 3600}$; and $\mathbf{X} \in \mathbb{R}^{n\times 4608}$ for **BERT** and **BETO** embeddings, where n represents the number of samples (texts) in the considered database. For example, $n = 404$ subjects for the case of YouTube transliterations, $n = 294$ subjects for the case of the English Twitter data, and $n = 171$ subjects for the case of Spanish Twitter data.

2.3 Learning methods

2.3.1 Support Vector Machine (SVM)

The goal of an SVM is to discriminate data samples by finding a separating hyperplane such that maximizes the margin between classes. Soft Margin SVM allows errors in the process of finding the optimal hyperplane. These allowed errors are data samples located on the wrong side of the hyperplane but within the optimal margin. An example of Soft Margin SVM is shown in Figure 2.6.

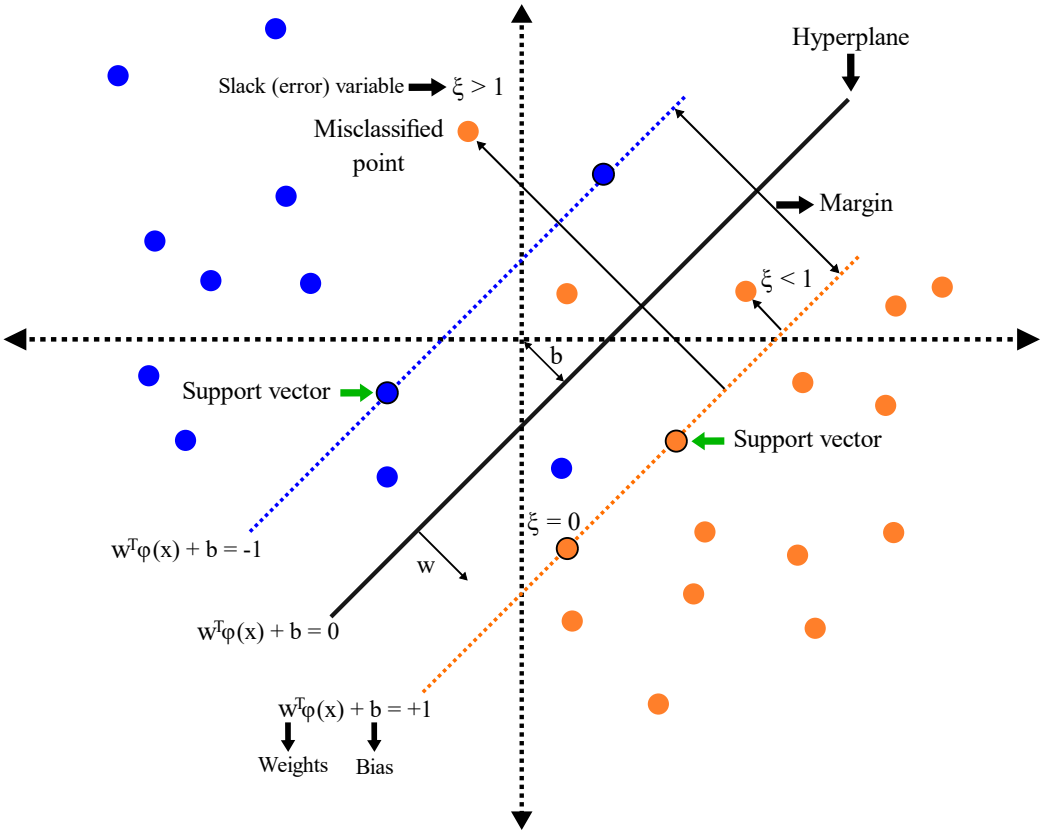

Figure 2.6. Soft-Margin SVM. Figure adapted from [57].

The decision function of a Soft Margin SVM is expressed according to the Equation 2.4, where ξ_n is a slack variable that penalizes the amount of errors allowed in the optimization process. $y_n \in \{-1, +1\}$ are the class labels, $\phi(\mathbf{x}_n)$ is a Kernel Function to transform the feature space $\mathbf{x}$ into a higher

dimensional space where a linear solution to the problem can be found. The weight vector $\mathbf{w}$ and the bias value b define the separating hyperplane.

$$y_n \cdot (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n, \quad n = 1, 2, 3, \cdots, N \tag{2.4}$$

The optimization problem for finding the hyperplane is defined in Equation 2.5, where the hyperparameter C controls the offset between ξ_n and the margin width. The samples $\mathbf{x}_n$ that satisfy the condition of equality in the Equation 2.4 are called support vectors ($\mathbf{x}_m$).

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \tfrac{1}{2} ||\mathbf{w}||^2 + C \textstyle\sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & y_n \cdot (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned} \tag{2.5}$$

In this Work, the Gaussian kernel function is considered $\phi(\mathbf{x}_n) = e^{-\gamma^2 ||\mathbf{x}_n - \mathbf{x}_m||^2}$, where the hyper parameter γ is the kernel bandwidth. More details about the process of optimizing an SVM with a kernel function can be found in [58].

2.3.2 Support Vector Regression (SVR)

When SVMs are used for regression they turn into SVR. The main change is that instead of predicting binary labels, a regressor is optimized to predict a real value. In ε-SVR, the goal is to find a function $f(\mathbf{x})$ that has at most ε deviation from the actually targets y_i for all the training data and at the same time is as flat as possible [59]. When we are describing linear functions, $f(\mathbf{x})$ has the form:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{2.6}$$

Where $\langle \cdot, \cdot \rangle$ denotes the dot product and $b \in \mathbb{R}$. Flatness in 2.6 means that one seeks a small $\mathbf{w}$, which is obtained minimizing its norm, i.e. $||\mathbf{w}||^2 = \langle \mathbf{w}, \mathbf{w} \rangle$. This problem could be write as a convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2} ||\mathbf{w}||^2 \\ \text{subject to} \quad & y_i - \langle \mathbf{w}, \mathbf{x_i} \rangle - b \leq \varepsilon \\ & \langle \mathbf{w}, \mathbf{x_i} \rangle + b - y_i \leq \varepsilon \end{aligned} \tag{2.7}$$

The assumption in Equation 2.7 is that the function $f(\mathbf{x})$ exists and the convex optimization problem is feasible. However this is not always the case. Thus, similarly to the soft margin SVM, one can introduce slack variables ξ_i and ξ_i^* to cope with otherwise infeasible constraints of the optimization problem in Equation 2.7. The resulting optimization problem is as follows [60]:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\,||\mathbf{w}||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle \mathbf{w}, \mathbf{x_i} \rangle - b \leq \varepsilon + \xi_i \\ & \langle \mathbf{w}, \mathbf{x_i} \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{2.8}$$

The constant $C > 0$ determines the trade-off between the flatness of $f(\mathbf{x})$ and the maximum allowed deviation ε. This corresponds to the so called ε-insensitive loss function $|\xi|_\varepsilon$, which is described as:

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon; & \text{otherwise} \end{cases} \tag{2.9}$$

Figure 2.7 illustrates the concept. Note that only points outside the region between the dotted line (the "tube") contribute to the cost. Deviations are linearly penalized although it is possible to extend the SVR to nonlinear functions [59], [61].

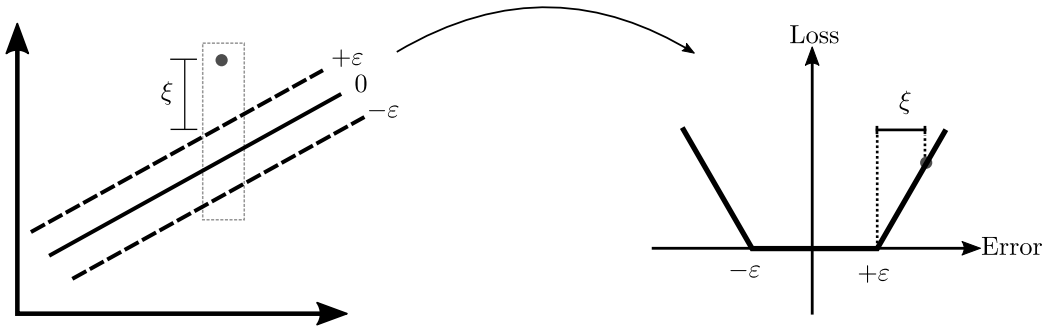

Figure 2.7. Linear SVR with ε-insensitive loss function. Figure adapted from [59].

2.3.3 Convolutional Neural Network (CNN)

The idea of a CNN model was firstly applied in the field of Image Recognition, where a picture can be transformed into a matrix, then, a convolution operation is performed on the input data with the use of a filter or kernel to then produce a feature map or also named activation map. The main benefit of CNN is the learning of the kernels. For years, kernels (filters) were designed by human to detect particular features (vertical edges, small spots, etc). With the CNNs, the kernels are learned automatically and it is the task of the network to select the more relevant values for the given task. The convolution is executed by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result into the feature map [62]. This filter scans from two directions (left to right, and from top to down). When the filter detects its feature on a subpart of the image, the activation value will be high. A CNN is constructed of multiple convolutional layers. After each layer, we end up with a feature map that will be passed to the next layer. What happens with the aggregation of layers is the following: simple features are extracted in shallow layers and complex features can be captured in the deeper ones.

In NLP applications, the filter scans from top to down. Here, a dataframe is prepared with one word in each row, and columns are filled up with features generated from a word embedding model (normally using a Word2Vec model or GloVe pre-trained model) [63]. Normally, the filter size can be set from $n = 2$ to 5, where larger filter sizes are computationally more demanding. As is shown in Figure 2.8, K2 to K5 represent each feature map produced varying the filter size from 2 to 5. These feature maps are also known as convolutional layers. After the convolution, one of the pooling methods (for example, global max pooling) is applied to these feature maps. Global max pooling consists in extracting the maximum value from each feature map to produce a pooling layer, as can be seen in Figure 2.9.

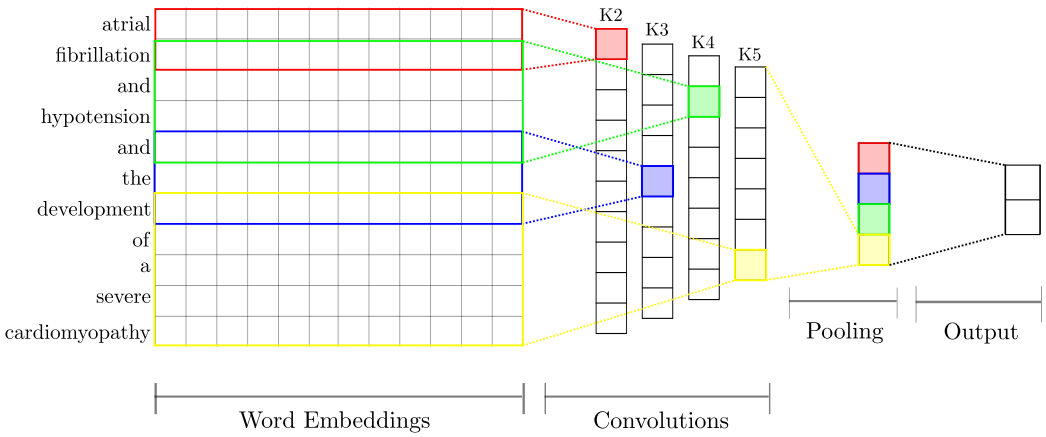

Figure 2.8. Example of 1D CNN architecture. Figure adapted from [64].

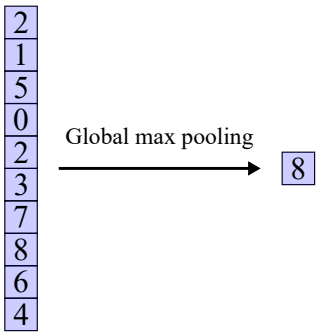

Figure 2.9. Example of 1D Global Max Pooling.

The output of the pooling layer is flattened and then goes as input to a fully connected layer to generate the output layer. In the output layer, an activation function (normally an softmax activation function) is applied before producing the predicted labels. So far, the layers are fully connected, which might have the following two disadvantages: high computation time and over-fitting. The drop-out technique is a common regularization technique to avoid these two disadvantages, and consists of preventing complex co-adaptations on training data, ignoring (dropping) randomly selected neurons during training. This can be done using a dropout value from 0 to 1, where using a value of 0.20 will mean randomly ignore (drop) 20% percent of the nodes (neurons) of the layer during the training of the model.

2.3.4 Recurrent Neural Networks (RNN)

The Recurrent Neural Networks (RNNs) are different to the other Neural Networks architectures because they incorporate feedback such that it introduces memory into the net, which is really useful when the input corresponds to a sequence such as in speech or language. The decision of a recurrent net in a time step t is affected by the decision in a time step $t-1$. Thus, RNNs have two inputs, the present and the recent past. The networks will be connected to their past decisions (feedback loop), having as input their own outputs from previous moments. RNNs finds correlation with the events that have occurred several moments ago. These correlations are called "long-term dependencies". In this case, language is full of words that depend upon the previous word, as a feedback loop.

In Figure 2.10 the traditional model of an RNN is shown. On the left side is the compact form of an RNN, which does not describe the temporal character of the RNN in detail. On the right side its extended form: for a record of the data set, x_t, the input will be multiplied by a weight transition matrix W_{xh} and will be added the value of the previous state h_{t-1} multiplied by the weight transition matrix W_{hh}. The previous expression $(x_t \cdot W_{xh} + h_{t-1} \cdot W_{hh})$ will pass through a neuron that contains an activation function $\tanh(\cdot)$ (in green), with which we will obtain the value of the state h_t. With this state, we will obtain two possible functions, (1) we will be able to determine the output z_t, by multiplying the current state by the output transition matrix W_{hz} and going through the activation function $\text{softmax}(\cdot)$ (in blue), and (2) the state h_t will go to the next neuron for the next state computation h_{t+1}, repeating the process described above. Additionally, the red lines shows the gradient path by the method of *back propagation* over time.

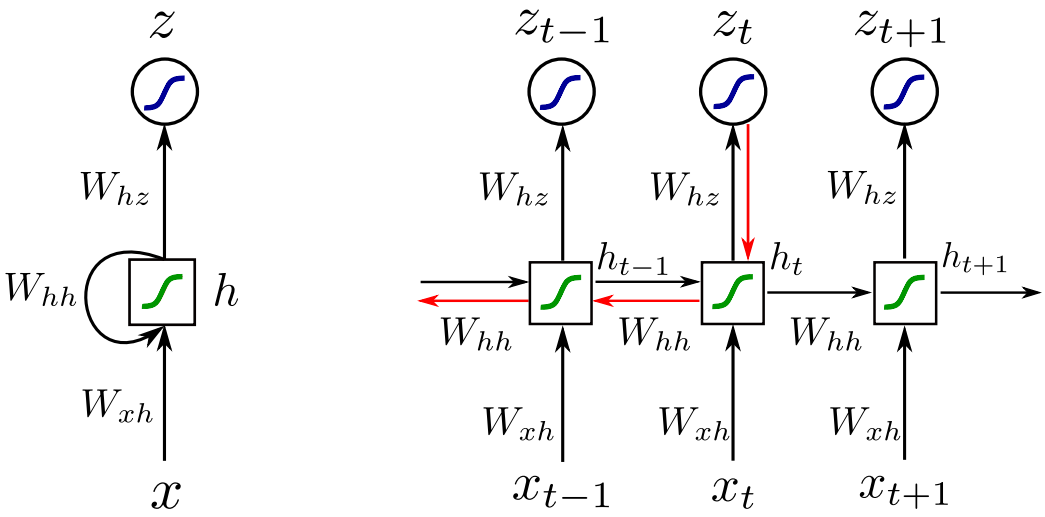

Figure 2.10. Diagram of an RNN layer and its temporal expansion. Figure adapted from [65].

RNNs use a method of *back propagation* over time for the optimization of network parameters at the training stage. Because RNNs can propagate errors in a long sequence of data, the error value will decrease layer by layer, and will eventually vanish after t states. Therefore, neurons in the more distant states will not contribute to the optimization of the RNN parameters. Considering this weakness of RNNs, Long Short Term Memory (LSTM) units were created to handle the problem of vanishing gradient that appears in conventional RNNs. These architectures have been used for Deep Learning, they have shown to improve results compared to traditional RNN, and have also been used for activity recognition, emotion detection, among others.

Long Short Term Memory (LSTM):

Basic RNN architectures incorporate activation cycles with the previous network inputs for decision making on the current input. A problem with basic RNNs is the vanishing of the gradient over time. LSTMs extend the basic RNN model and offer two main advantages: (1) they introduce the information from the memory or cell, and (2), they remain stable with respect to long input sequences, compensating for the problem of vanishing gradient. LSTMs are commonly called memory units or c_t cells, which have the same inputs (h_{t-1} and x_t) and the same h_t output from a common RNN, but this network has more control units for the flow of information. Additionally, it

is regulated and controlled by four gates: The input gate i_t, the cell status gate $\tilde{c}_t$, the output gate o_t and the forget gate f_t, which implement read, write and cell reset functions respectively.

In Figure 2.11, it can be observed the structure of an LSTM neural network for a time t, where i_t, $\tilde{c}_t$, f_t and o_t denote the control gates and $\otimes$ denotes a multiplication operation. Initially, the information flow starts with the cell state gate $\tilde{c}_t$, which takes an estimated value of what the new cell state may be, multiplying the data x_t and the previous state h_{t-1} with their corresponding transition matrix W_{xc} and W_{hc} and additionally they go through a $\tanh(\cdot)$ activation function. The input gate i_t with the data x_t and h_{t-1} and with the multiplication of transition matrices W_{xi} and W_{hi}, and additionally passing through an activation function sigmoid$(\cdot)$, maps the result between values $\in [0, 1]$, where 0 indicates the inhibition of the $\tilde{c}_t$ gate and 1 indicates the total activation of the $\tilde{c}_t$ gate. The forget gate f_t is activated with the data x_t and h_{t-1}, with their respective matrices W_{xf} and W_{hf} and passing through a sigmoid$(\cdot)$ activation function, which inhibits or activates the state of the previous cell c_{t-1}. With the results of the previous gates, the value c_t of the cell can be estimated. The output gate o_t with the data x_t and h_{t-1}, with their respective matrices W_{xo} and W_{ho} and passing through a sigmoid$(\cdot)$ activation function, modulates the state of the cell c_t, enabling a new state h_t for subsequent cells. Finally, the output z_t is determined by multiplying h_t by the output transition matrix W_{hz} and passing through a softmax$(\cdot)$ activation function.

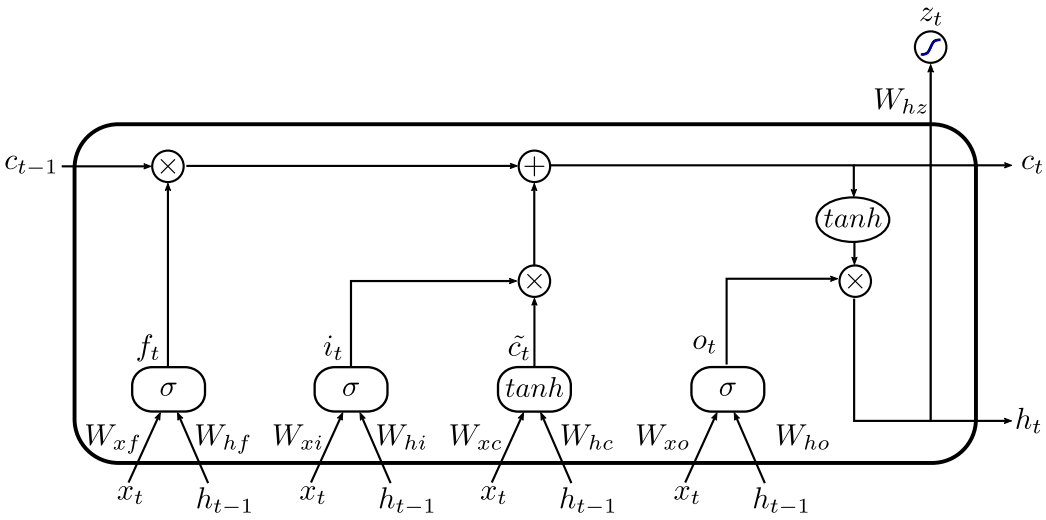

Figure 2.11. Structure of an LSTM unit with 4 gates: cell status gate, input gate, output gate, forget gate. Figure adapted from [66].

Chapter 3

Databases

The details of the two databases used in this work are described below: YouTube transliterations and Twitter statuses.

3.1 YouTube Personality dataset

This dataset consists of manual transliterations of audio-visual recordings generated by 404 YouTube vloggers that explicitly show themselves in front of a webcam talking about a variety of topics including personal issues, politics, movies, and books. The corpus was originally presented in Biel et al. (2013). There is no content-related restrictions in the videos and the language is natural, diverse and informal. The transliterations contain approximately 10K unique words and 240K word tokens. The data is gender-balanced (52% female). The transliterations are originally produced in English Language and the videos in the database were automatically labeled according to the five traits of the OCEAN model. The labeling process was performed using the Amazon Mechanical Turk [67] and the Ten-Item Personality Inventory [68], giving values for the scores in the range [1.9 and 6.6]. Figure 3.1 shows histograms with the scores assigned to each trait. Note that the corpus contains the Emotional stability label instead of **N**euroticism, which is its opposite trait. Some statistical information of the scores in the traits is also provided in Table 3.1.

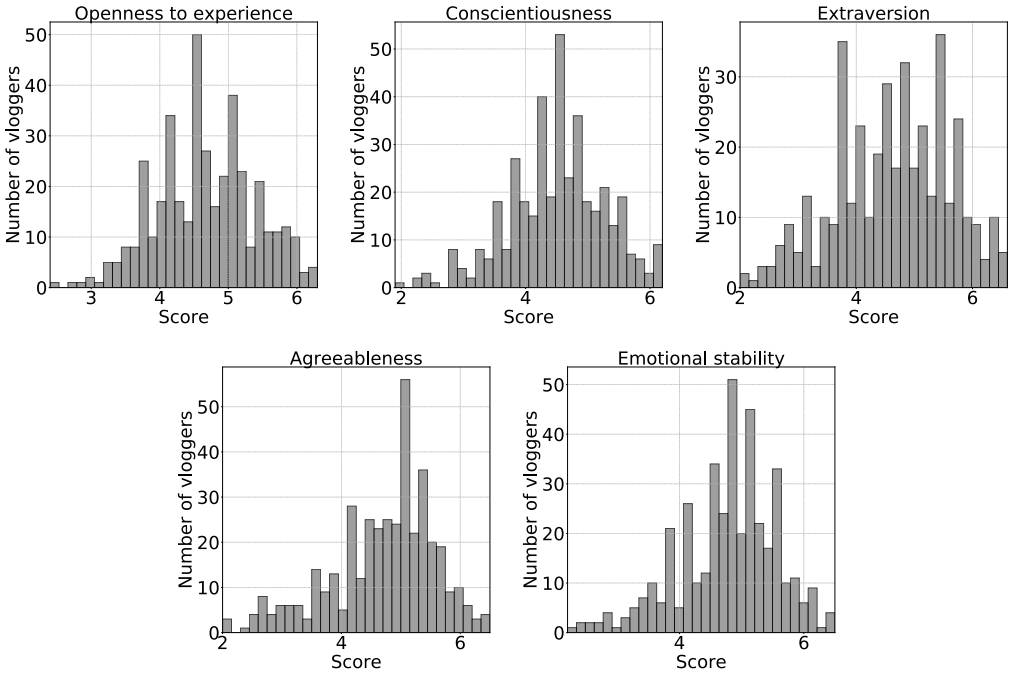

Figure 3.1. Histogram of the score in the 5 traits for YouTube database.

Table 3.1. Statistical information of the scores in the personality traits for YouTube database.

Trait	Min.	T1	Med.	T2	Max.	Std.
Openness to experience	2.40	4.40	4.70	5.00	6.30	0.72
Conscientiousness	1.90	4.20	4.50	4.80	6.20	0.77
Extraversion	2.00	4.20	4.70	5.20	6.60	0.98
Agreeableness	2.00	4.40	4.90	5.10	6.50	0.88
Emotional stability	2.20	4.50	4.80	5.10	6.50	0.78

Med.: Median; Min.: Minimum; Max.: Maximum;

T1: 1st tertitle; T2: 2nd tertile; Std.: Standard deviation.

Since the data is originally in English and one of the goals of this work is to evaluate the models with languages other than English, the transliterations were automatically translated into Spanish using the *TextBlob* Python

library, which internally uses the Google Translate API [69] to create the database that we will call from now on "Spanish YouTube Personality dataset". Statistical information about the number of words per text for both English and Spanish languages can be seen in Table 3.2; and the distribution of the number of words before and after the preprocessing (described in subsection 2.1.1) is shown in Figure 3.2.

Table 3.2. Statistical information of the number of words per text for YouTube database.

Data	Min.	Med.	Max.
English raw	24.0	466.0	2031.0
English pre-processed	17.0	233.5	1037.0
Spanish raw	24.0	443.5	1945.0
Spanish pre-processed	15.0	213.5	975.0

Med.: Median; Min.: Minimum; Max.: Maximum.

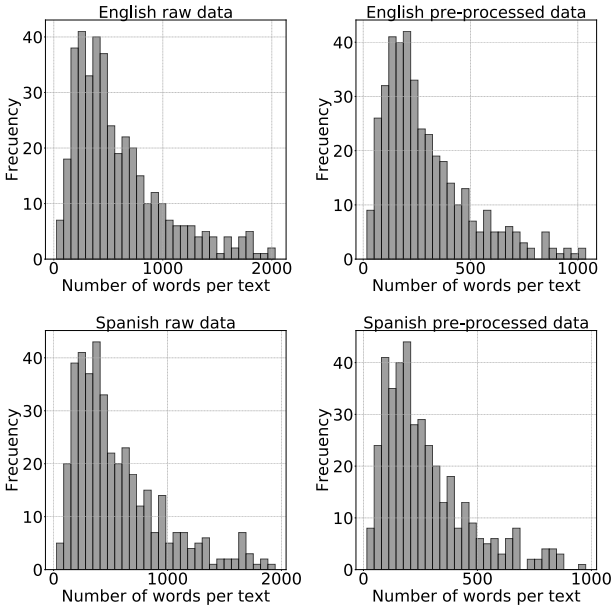

Figure 3.2. Histogram of the number of words per text for YouTube database.

3.2 PAN-AP-2015 dataset

The data from Twitter consist in the PAN-AP-2015 corpus [8], which is a database with tweets in 4 different languages: English, Spanish, Italian and German. In this work, we will focus on the texts in English (294 subjects in total, 152 in the training set and 142 in the test set) and Spanish Language (188 subjects in total). Because exploring the Spanish database we found subjects with tweets in a language different to Spanish, we removed them and a total of 171 subjects remained (92 subjects for the training set and 79 for the test set).

The corpus is balanced with respect to gender and has personality annotations that consist in scores in the big five personality traits: extraversion, emotional stability (the inverse of neuroticism), agreeableness, conscientiousness and openness to experience. Personality traits were self-assessed with the Big Five Inventory (BFI-10) online test [70] and reported as standardized scores between -0.5 and +0.5. Table 3.3 shows the number of instances per language in both languages for the training and test sets, where each of the subjects has approximately 100 Tweets.

Table 3.3. Number of instances for PAN-AP-2015 corpus in English and Spanish language.

	Training data		**Test data**	
Language	**# Subjects**	**#Tweets**	**# Subjects**	**#Tweets**
English	152	14152	142	13178
Spanish	92	9132	79	7729

Statistical information about personality trait scores for the two languages can be found in Table 3.4. Similarly, information about the number of words per text after the pre-processing stage mentioned in 2.1.2 is found in Table 3.5.

Table 3.4. Statistical information of the scores in the personality traits for Twitter database.

Trait		English						Spanish					
		Min.	**T1**	**Med.**	**T2**	**Max.**	**Std**	**Min.**	**T1**	**Med.**	**T2**	**Max.**	**Std**
Openness to experience	Training	-0.1	0.2	0.2	0.3	0.5	0.146	-0.1	0.1	0.1	0.2	0.5	0.155
	Test	-0.1	0.2	0.2	0.4	0.5	0.157	-0.1	0.1	0.2	0.2	0.5	0.135
Conscientiousness	Training	-0.2	0.1	0.2	0.3	0.5	0.151	-0.2	0.2	0.25	0.4	0.5	0.188
	Test	-0.2	0.1	0.2	0.2	0.5	0.148	-0.2	0.2	0.2	0.3	0.5	0.169
Extraversion	Training	-0.3	0.1	0.2	0.2	0.5	0.167	-0.3	0.1	0.2	0.2	0.5	0.177
	Test	-0.3	0.1	0.2	0.2	0.5	0.158	-0.3	0.1	0.2	0.2	0.5	0.199
Agreeableness	Training	-0.3	0.1	0.1	0.2	0.5	0.158	-0.2	0.1	0.2	0.2	0.5	0.168
	Test	-0.3	0.1	0.2	0.2	0.5	0.152	-0.2	0.1	0.2	0.2	0.5	0.179
Emotional stability	Training	-0.3	0.1	0.2	0.2	0.5	0.223	-0.3	-0.1	0.1	0.2	0.5	0.202
	Test	-0.3	0.0	0.1	0.3	0.5	0.230	-0.3	-0.1	0.1	0.2	0.5	0.209

Med.: Median; Min.: Minimum; Max.: Maximum; T1: 1st tertitle; T2: 2nd tertile; Std.: Standard deviation.

Table 3.5. Statistical information of the number of words per text for Twitter database.

	Data per subject			**Data per tweet**		
Data	**Min.**	**Med.**	**Max.**	**Min.**	**Med.**	**Max.**
English - Training	203	892	1607	1	9	30
English - Test	191	873	1784	1	9	32
Spanish - Training	645	1203.5	1644	1	12	32
Spanish - Test	278	1179	1898	1	11	31

Med.: Median; Min.: Minimum; Max.: Maximum.

Chapter 4

Methodology

A summary of the methodology implemented in this study is shown in Figure 4.1. In general terms, the procedure starts with the pre-processing of the texts, which consists in noise removal and lexicon normalization (see section 2.1 for details). Then, feature extraction process is performed (considering the features described in the section 2.2). Later, two automatic learning approaches will be explored: (1) classical machine learning methods and (2) deep learning methods, which were explained in section 2.3. The personality scores of the OCEAN model will be considered as the reference/gold standard labels to be predicted. Since the methods for feature extraction, classification and regression were presented in the theoretical background, the following subsections will introduce the algorithms used in this work to evaluate the performance of the proposed approaches.

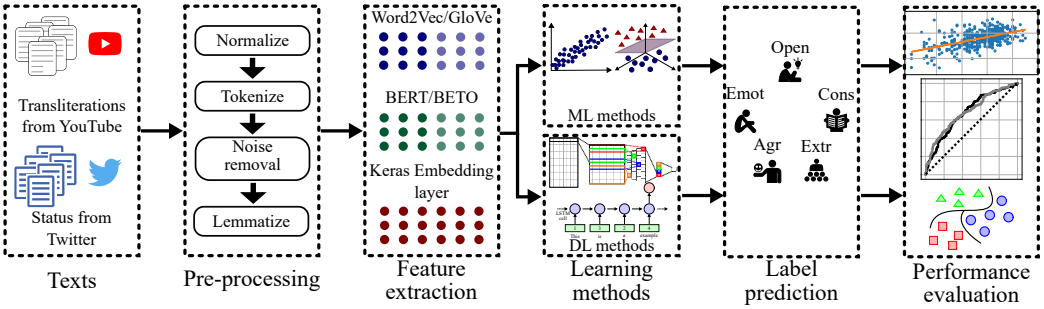

Figure 4.1. Block diagram of the methodology implemented in this study.

4.1 Validation methods

4.1.1 k-Fold Cross-Validation

For the optimization and validation of machine and deep learning methods in the *YouTube Personality dataset*, we use the k-Fold Cross-Validation (CV) technique. CV is a model validation technique to evaluate how the results of a statistical analysis will be generalized to a set of independent data [71]. The purpose of CV is to test the ability of the model to predict new data that was not used in the estimation, to point out problems such as overfitting and give an idea of how the model will be generalized to an independent data set. In the CV of k-divisions, also called k-Fold Cross-Validation, as it is shown in Figure 4.2, the data set is divided into k mutually exclusive subsets of approximately the same size, then the system is tested with each of these divisions and finally, the performance will be the average of the performance of each of these tests [71].

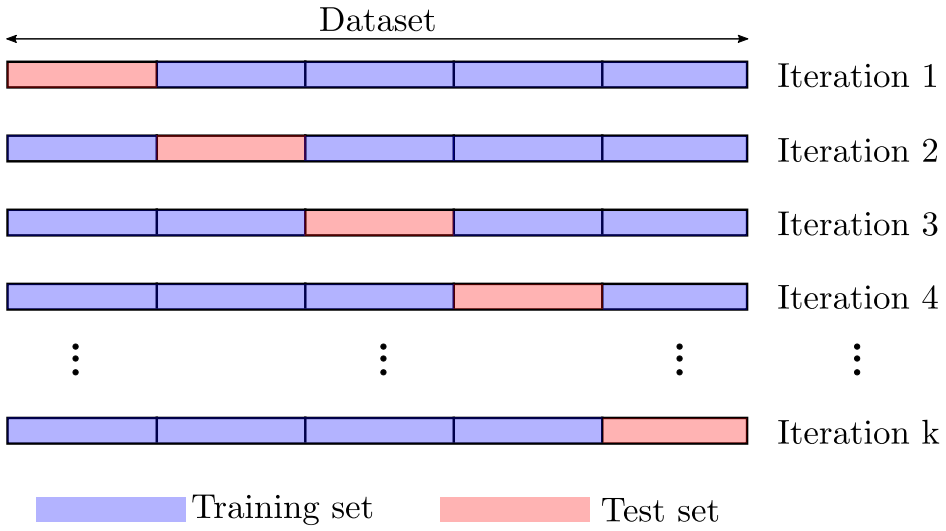

Figure 4.2. Diagram of k-Fold Cross-Validation technique.

4.1.2 Hold-out validation

Now, for the implementation of learning methods using the PAN-AP-2015 dataset, since it was previously divided into two sets: train and test, we used the Hold-out validation technique to adjust the hyperparameters. Based on Figure 4.3, the Hold-out validation method works in the following way:

1) the data set is divided into three parts: training data set, validation data set and test data set; 2) different models are generated with the training data set resulting from different combinations of the hyperparameters; 3) the performance of each of these models is tested on the validation data set; 4) the optimal model is selected from the models tested on the validation data set, which will have the optimal hyperparameters for the implemented algorithm; and finally, 5) to measure the performance of the optimal model on the test data set.

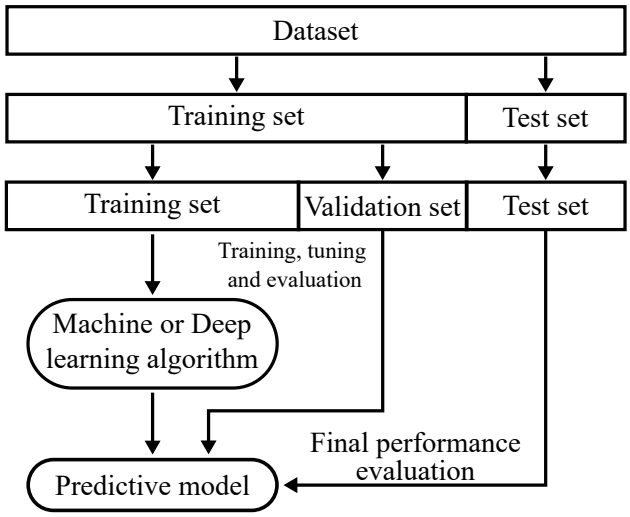

Figure 4.3. Diagram of Hold-out validation technique for hyperparameters tuning. Figure adapted from [72].

4.2 Parameters optimization

4.2.1 Parameters optimization for SVR and SVM

In this work, with respect to classical ML methods, we use the ε-SVR for regression and the Soft-Margin SVM for classification. Gaussian kernels were used for both tasks, and hyper-parameters C, γ, and ε are optimized through a grid-search up to powers of ten in the range between $[1 \times 10^{-4}$ and $1 \times 10^{4}]$.

4.2.2 Parameters optimization for CNN and LSTM

Now, with respect to DL methods, we implemented some architectures based on CNNs and LSTMs. Due to the fact that in some cases there was overfit-

ting, it was necessary to implement some regularization techniques such as the use of L2 regularization and Dropout. As with the classical methods, in this case a grid-search was used in order to find the optimal values for the hyperparameters: learning rate, the value for L2 regularization and the value for Dropout. The grid-search for the learning rate and for the L2 regularization is given by powers of 10 in the range $[1 \times 10^{-5}$ and $1 \times 10^{-3}]$, while for the dropout value, the grid-search is in the range $[0.1, 0.3, 0.5]$.

4.3 Performance metrics

4.3.1 Metrics used for Regression

Pearson's correlation coefficient (r):

The Pearson product-moment correlation coefficient or Pearson's correlation coefficient is a measure of the strength of a linear association between two variables and it is denoted by r. Basically, a Pearson product-moment correlation attempts to draw a line of best fit through the data of two variables, and r indicates how far away all these data points are from this line of best fit (i.e., how well the data points fit this new model/line of best fit) [73]. It has the following characteristics: $r \in [-1, 1]$, a value of $r = 0$ indicates that there is no association between the two variables; a value $r \geq 0$ indicates a positive association, that means that if the value of a variable increases, the value of the other variable also increases; and a value $r \leq 0$ indicates a negative association, which means that if a variable increases, the other variable decreases. Pearson's correlation coefficient is computed as follows:

$$r_{x,y} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{4.1}$$

Where σ_{xy} is the covariance between x and y; σ_x and σ_y are the variance of x and y respectively.

Spearman's correlation coefficient(ρ):

Spearman's rank correlation is a non parametric test that is used to measure the degree of association between two variables. The Spearman's rank correlation test does not carry any assumptions about the distribution of the data and it is the appropriate correlation measure when the variables are

measured on a scale that is at least ordinal. It has the following characteristics: its value ranges from -1 to +1 and has and advantage over Pearson correlation which consist in find out nonlinear correlations between variables. Spearman's correlation coefficient, ρ, is defined as:

$$\rho_{x,y} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{4.2}$$

Where d_i is the difference in the ranges of x and y, and n is the number of elements in x and y.

Mean Absolute Error (MAE):

MAE quantifies the difference between two continuous variables. MAE measures the average magnitude of the errors in a set of predictions without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation (real value), where all individual differences have equal weight. MAE is computed as is shown in Equation 4.3.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.3}$$

Where y_i corresponds to the actual value, $\hat{y}_i$ to the predicted value and n is the numbers of data samples.

Root Mean Squared Error (RMSE):

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It is defined as the square root of the average of squared differences between prediction and actual observation. RMSE is computed as is shown in Equation 4.4.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.4}$$

4.3.2 Metrics used for Bi-class Classification

Confusion matrix:

In general, when a process involving pattern recognition is performed, the so-called contingency or confusion matrix is used. The performance of the system is evaluated depending on the number of hits and the number of misses in the classification stage of new data [74]. A confusion matrix for two-class classification systems is shown in Table 4.1.

Table 4.1. Confusion matrix.

	True class	
Estimated class	Class 0	Class 1
Class 0	TP	FP
Class 1	FN	TN

According to this matrix and taking class 0 as the target one, the following terms are defined:

- ✓ True positive (TP): refers to the number (or percentage) of class 0 patterns that the system correctly classifies as belonging to class 0.
- ✓ False negative (FN): corresponds to the number (or percentage) of class 0 patterns that the system incorrectly classifies as belonging to class 1.
- ✓ False positive (FP): is the number (or percentage) of class 1 patterns that the system incorrectly classifies as belonging to class 0.
- ✓ True negative (TN): is the number (or percentage) of class 1 patterns that the system correctly classifies as belonging to class 1.

Accuracy (ACC):

This measure is the proportion of patterns correctly classified by the system:

$$ACC = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}} \tag{4.5}$$

Sensitivity (SEN):

Sensitivity indicates the system's ability to detect reference class patterns. For example, the percentage of sick people who are correctly identified as carriers of the condition.

$$SEN = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.6}$$

Specificity (SPE):

Specificity indicates the system's ability to reject patterns that do not belong to the reference class. For example, the percentage of healthy people who are correctly identified as not having the condition.

$$SPE = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{4.7}$$

Precision (PRE):

It refers to the proportion of positive results that are truly positive.

$$PRE = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.8}$$

F1-Score (F1):

It is the harmonic mean of precision and sensitivity.

$$F1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{4.9}$$

Receiver Operating Characteristic curve (ROC):

The ROC curve is a graphical representation which shows the binary classifier performance, while its discrimination threshold is varied. Usually, y-label are the True Positives (Sensitivity) and x-label are the False Positives ($1 - \text{Specificity}$). Figure 4.4 shows the derivation of a ROC curve (right) for different threshold points in the probability distribution of the two classes (left). From the distribution that is shown, the inevitable presence of false regions is indicated. The placement of the line at the threshold, also called Cutoff Value (CV) determines how the total error is distributed. By placing

the CV at point E, there is a 0 FP rate and 0 TP rate (sensitivity). Therefore, it has no applicable value, everyone is going negative. Also, at the other extreme (point A), the sensitivity is 1 and the FP rate is close to 1. The optimal balance is somewhere between points A and E, like point C, where most of the two classes are receiving accurate results.

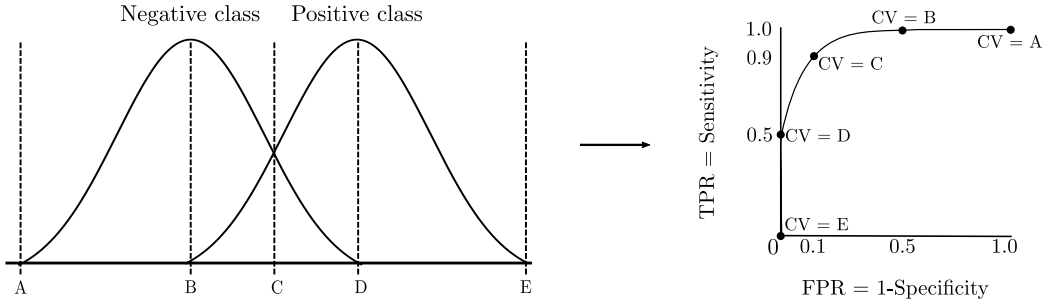

Figure 4.4. Distribution figure and ROC curve. **CV:** Cutoff Value, **TPR:** True Positive Rate, **FPR:** False Positive Rate. Figure adapted from [75].

4.3.3 Metrics used for Tri-class Classification

Unweighted Average Recall (UAR):

It is the accuracy per class divided by the number of classes without considerations of instances per class. It is also the mean of the recall values of all the classes considered.

Cohen's kappa coefficient (κ):

The Cohen's kappa coefficient, is a robust statistic useful for reliability testing among evaluators (degree of agreement among evaluators). Similar to some correlation coefficients, it can vary from -1 to +1, where 0 represents the amount of agreement that can be expected from a random opportunity, and 1 represents perfect agreement among evaluators. As with all correlation statistics, κ is a standardized value and is therefore interpreted in the same way in multiple studies [76] and is defined as:

$$\kappa = \frac{p_o - p_\epsilon}{1 - p_\epsilon} \tag{4.10}$$

Where p_o is the empirical probability of agreement on a label assigned to any sample, and p_ϵ is the expected agreement when both evaluators assign

labels at random. Cohen suggested that the result of κ be interpreted as follows: $\kappa \leq 0$ indicate no agreement, $0.01 \leq \kappa \leq 0.20$ as no agreement at all, $0.21 - 0.40$ as fair, $0.41 - 0.60$ as moderate, $0.61 - 0.80$ as substantial, and $0.81 - 1.00$ as near perfect agreement.

Chapter 5

Experiments

5.1 Data distribution and statistical analyses

5.1.1 Data distribution

As mentioned in previous chapters, the three main experiments in this work are: experiments of personality trait estimation using regression systems; bi-class classification experiments between weak presence and strong presence of personality traits; and finally, tri-class classification experiments to classify the presence of personality traits into 3 different levels: low presence (LP), medium presence (MP), and high presence (HP).

For the regression systems, on the YouTube and Twitter databases, we used the scores on each of the five traits of the OCEAN model (remember that the databases came with the scores for the emotional stability trait instead of the neuroticism trait). Now, for the two-class and three-class classification systems, the labels were constructed as follows: for the binary classification scenario, the scores of each trait are divided around their median, i.e., samples with values below the median are considered to have weak presence of the trait (codified as 0), while those above are labeled as strong presence (codified as 1). This distribution criterion allows us to have a balanced number of samples per trait in most of the cases. The median threshold is shown in Figure 5.1, Figure 5.2, and Figure 5.3 as a red dotted line. The distribution of the data for the tri-class classification problem is according to the tertiles of the scores distribution per trait. This strategy guarantees the balance among the three resulting subgroups. The distribution of these three subgroups is shown in Figure 5.1, Figure 5.2, and Figure 5.3 as the three

shadowed regions.

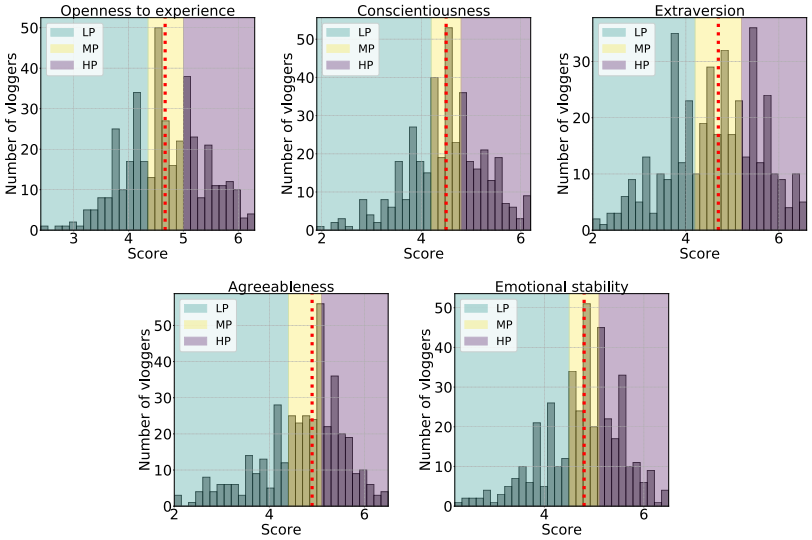

Figure 5.1. Score thresholds for the bi-class and tri-class classification problems in YouTube Personality dataset. LP: low presence; MP: medium presence; HP: high presence.

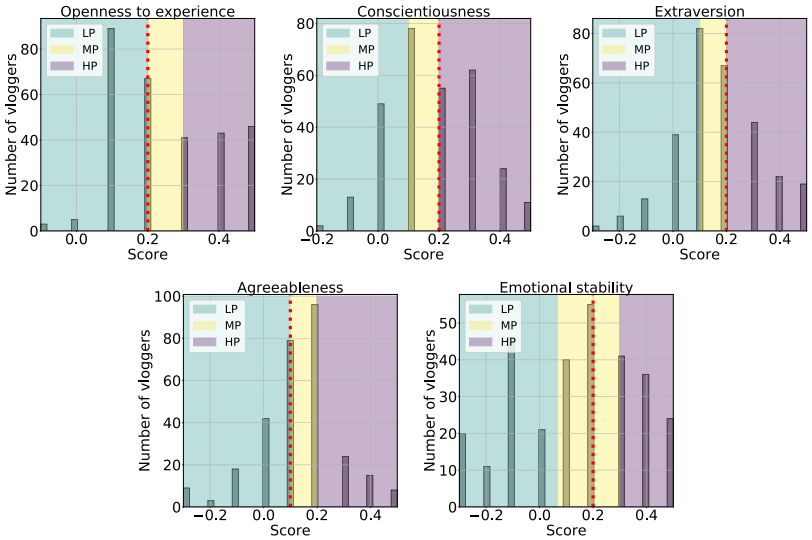

Figure 5.2. Score thresholds for the bi-class and tri-class classification problems in PAN-AP-2015 English dataset. LP: low presence; MP: medium presence; HP: high presence.

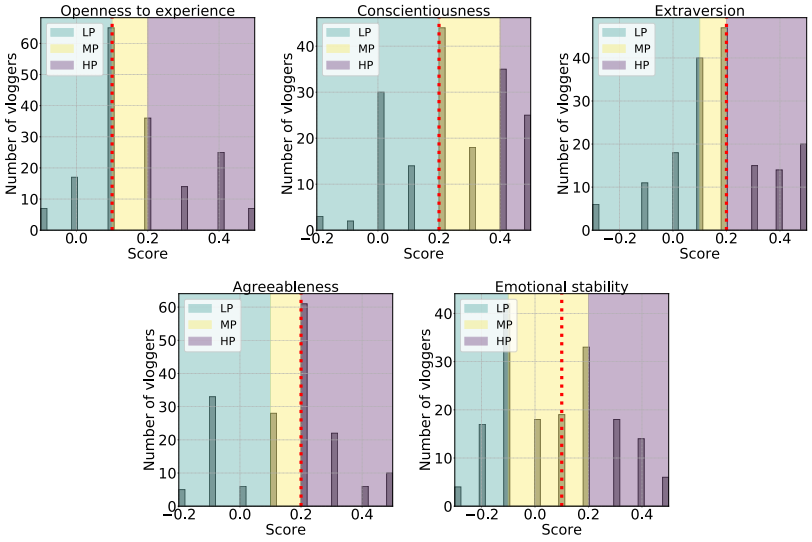

Figure 5.3. Score thresholds for the bi-class and tri-class classification problems in PAN-AP-2015 Spanish dataset. LP: low presence; MP: medium presence; HP: high presence.

For both YouTube and Twitter datasets, the number of samples per class and subgroup (two for the bi-class problem and three for the tri-class problem) are summarized in Table 5.1, Table 5.2 and Table 5.3.

Table 5.1. Number of subjects for Bi-class and Tri-class classification problem in YouTube Personality dataset.

	Number of subjects for Bi-class problem		**Number of subjects for Tri-class problem**		
Trait	**Weak presence**	**Strong presence**	**Low presence**	**Medium presence**	**High presence**
Openness to experience	203	201	135	148	121
Conscientiousness	209	195	146	132	126
Extraversion	209	195	144	137	123
Agreeableness	218	186	137	138	129
Emotional stability	203	201	136	137	131

Table 5.2. Number of subjects for Bi-class and Tri-class classification problem in PAN-AP-2015 English dataset.

	Training data					Test data				
	# Subjects Bi-class problem		# Subjects Tri-class problem			# Subjects Bi-class problem		# Subjects Tri-class problem		
Trait	**WP**	**SP**	**LP**	**MP**	**HP**	**WP**	**SP**	**LP**	**MP**	**HP**
Openness to experience	50	102	50	39	63	47	95	47	46	49
Conscientiousness	73	79	35	65	52	69	73	29	40	73
Extraversion	73	79	32	41	79	69	73	28	41	73
Agreeableness	38	114	38	44	70	69	73	34	35	73
Emotional stability	66	86	47	19	86	51	91	39	51	52

WP: Weak presence; SP: Strong presence; LP: Low presence; MP: Medium presence; HP: High presence.

Table 5.3. Number of subjects for Bi-class and Tri-class classification problem in PAN-AP-2015 Spanish dataset.

	Training data					Test data				
	# Subjects Bi-class problem		# Subjects Tri-class problem			# Subjects Bi-class problem		# Subjects Tri-class problem		
Trait	**WP**	**SP**	**LP**	**MP**	**HP**	**WP**	**SP**	**LP**	**MP**	**HP**
Openness to experience	15	77	50	15	27	39	40	39	21	19
Conscientiousness	46	46	46	29	17	23	56	47	9	23
Extraversion	37	55	37	29	26	38	41	38	18	23
Agreeableness	39	53	39	35	18	33	46	33	26	20
Emotional stability	44	48	35	37	20	37	42	28	33	18

WP: Weak presence; SP: Strong presence; LP: Low presence; MP: Medium presence; HP: High presence.

5.1.2 Statistical analyses

Two statistical tests were performed. The first one is the Kruskal-Wallis test regarding the feature matrices extracted per sample and trait. The test was performed for the two scenarios: weak vs. strong presence of each trait, and the three levels of manifestation of the traits (LP, MP, HP). The second statistical test was χ^2 tests for both the bi-class and the tri-class scenarios, which intends to evaluate whether the gender of subjects biases the distribution of the extracted features.

With respect to the Kruskal-Wallis test, both for the YouTube Personality and PAN-AP-2015 datasets considering English and Spanish languages; in all of the cases, the null hypothesis H_0, "the median of the populations considered are equal", was rejected with $p \ll 0.01$. Now, with respect to the χ^2 tests, we found the following: for YouTube Personality dataset in English and Spanish languages; possible bias regarding gender was discarded for

extraversion, conscientiousness, emotional stability and openness to experience traits, while a possible bias was found for the agreeableness trait. With respect to PAN-AP-2015 dataset, in both languages English and Spanish, the gender bias was discarded for openness to experience, conscientiousness, extraversion and agreeableness traits, but it was found a possible bias for emotional stability trait.

5.2 Classification and regression experiments

As explained above, the 3 major experiments of this work are: regression, bi-class classification and triclass classification. For each one of theses experiments and taking into account the YouTube Personality dataset and the PAN-AP-2015 dataset, classical machine learning methods and deep learning methods were considered, where the details of the obtained results are in Chapter 6.

5.2.1 Experiments with classical machine learning methods

Since support vector machine (SVM) is one of the most used classification methods in the state of the art and considering its robustness when considering high-dimensional representation spaces [58], we decided to adopt this as our main framework for classical machine learning methods. For the regression experiments, we used the ε-SVR (see 2.3.2 for details), a Soft Margin SVM for the bi-class classification experiments (see 2.3.1 for details); and a SVM with the One vs All (OvA) (also called One vs Rest - OvR) approach for the tri-class classification experiments. This method consists of building one SVM per class, which is trained to distinguish the samples in one class from the samples of all the other classes and the decision is made according to the maximum output among all SVMs.

5.2.2 Experiments with deep learning methods

For the case of deep learning methods, neural networks based on convolutional layers and LSTM layers were considered (see 2.3.3 and 2.3.4 for details). In Figure 5.4 and Figure 5.5 the architectures implemented in this work are shown. Both architectures were created with the objective in mind to make them as simple as possible in order to reduce the number of parameters to be optimized due to the small number of samples to train the architectures

for both YouTube and Twitter data. The specific parameters for each architecture are shown in Table 5.4 and Table 5.5. With the previous objective in mind, we used different methods for the training of the weights of the Embedding layer (layer with the largest number of parameters to be optimized): i) the keras Embedding layer trained from scratch, ii) pre-trained Word2Vec and GloVe word embeddings with embedding layer freezing and iii) pre-trained Word2Vec and GloVe word embeddings without embedding layer freezing i.e., retraining of the word embeddings.

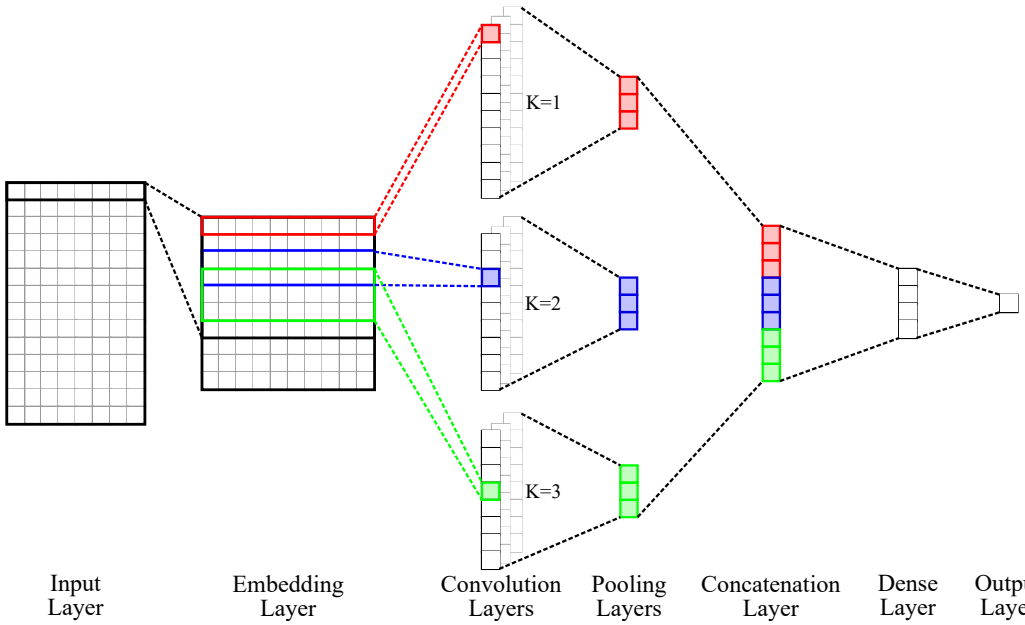

Figure 5.4. Architecture based on CNNs implemented in this study. K: size (high) of the filter.

For the case of the CNN-based architecture, 3 filter sizes were considered simultaneously to simulate the n-gram relationships of the words in the texts: n=1, n=2 and n=3. For the pooling layer, the global max pooling was taken into account, where the results are then concatenated, passed through a dense layer to reduce the dimension, and finally passed to the output layer, where the number of neurons and the activation function (with respect to the experiment) of these layer are shown in Table 5.5. For the case of the LSTM-based architecture, the number of units in the LSTM layer was optimized in the range of 32 to 512 in powers of 2. The embedding dimension and the number of neurons in the dense layer were kept the same as for the case of

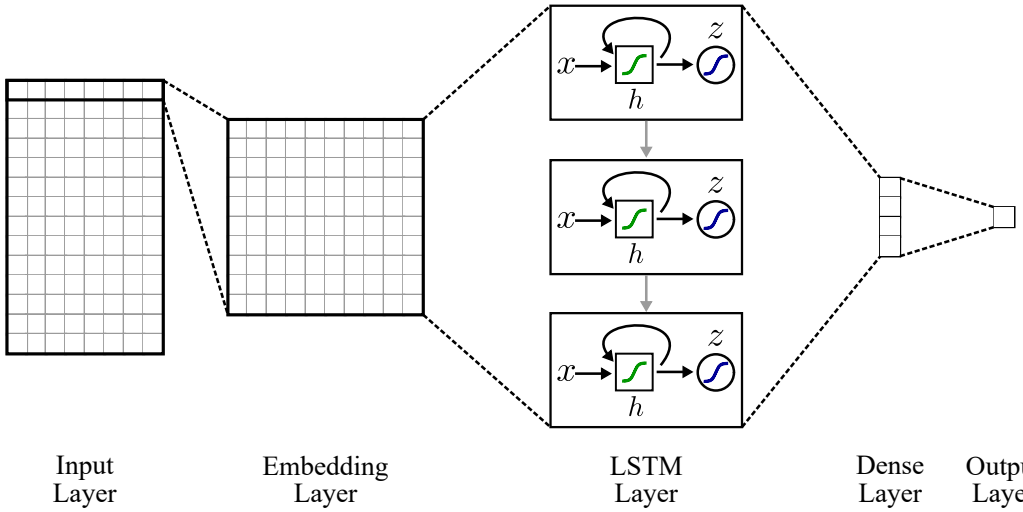

Figure 5.5. Architecture based on LSTMs implemented in this study.

the architecture with CNNs.

Table 5.4. Parameters for the architectures based on CNNs and LSTMs.

Architecture	Parameter	Value
CNN	Embedding dimension	300
	Number offilters	64
	Size of the filter	1(uni-gram), 2(bi-gram) and 3(tri-gram) simultaneously.
	Number of neurons dense layer	64
LSTM	Embedding dimension	300
	Number of units	Options: 32, 64, 128, 256 and 512.
	Number of neurons dense layer	64

Table 5.5. Number of neurons and activation function in the Output layer for regression, bi-class classification and tri-class classification experiments.

	Output Layer	
Experiment	**Number of neurons**	**Activation function**
Regression	1	Linear
Bi-class Classification	1	Sigmoid
Tri-class Classification	3	Softmax

Chapter 6

Results and discussion

This chapter shows the results obtained for the experiments explained in 5.2, as well as the analysis of the results and comparisons with works that took into account the same databases as our work.

6.1 Results with classical machine learning methods

We start with the results obtained with classical methods for the different experiments explained in 5.2.1.

6.1.1 Results with the English dataset from YouTube Personality

Personality trait estimation

This experiment was mainly based on SVR systems with Gaussian kernel. To allow comparisons with respect to other works in the literature that use the same corpus as we used here, results are reported in Table 6.1 in terms of the four metrics mentioned in subsection 4.3.1. Note that in three out of the five traits, the best result was obtained when merging the Word2Vec and GloVe embeddings, except extraversion which best result was obtained with the BERT-base embeddings and openness to experience which best result was obtained with GloVe embeddings. When observing r and ρ, the best result among all was obtained for the agreeableness trait ($r = 0.49$ and $\rho = 0.43$), followed by conscientiousness ($r = 0.40$ and $\rho = 0.41$), while the lowest result was for the openness to experience trait with $r = 0.22$ and $\rho = 0.21$. The lowest MAE = 0.55 was obtained with conscientiousness and

the lowest RMSE value is obtained with openness to experience with a value of RMSE = 0.70.

Table 6.1. Results for personality trait estimation with the English dataset from YouTube Personality considering a SVR.

Trait	Feature	r	ρ	MAE	RMSE
Open	Word2Vec	0.19 ± 0.02	0.18 ± 0.02	0.57 ± 0.00	0.70 ± 0.00
	GloVe	**0.22 ± 0.02**	**0.21 ± 0.02**	**0.56 ± 0.00**	**0.70 ± 0.00**
	Fusion	0.14 ± 0.02	0.16 ± 0.02	0.57 ± 0.00	0.71 ± 0.00
	BERT	0.06 ± 0.05	0.06 ± 0.04	0.58 ± 0.01	0.72 ± 0.01
Cons	Word2Vec	0.40 ± 0.01	0.40 ± 0.01	0.55 ± 0.00	0.71 ± 0.00
	GloVe	0.36 ± 0.02	0.37 ± 0.02	0.56 ± 0.01	0.72 ± 0.01
	Fusion	**0.40 ± 0.01**	**0.41 ± 0.01**	**0.55 ± 0.00**	**0.71 ± 0.00**
	BERT	0.39 ± 0.01	0.40 ± 0.01	0.55 ± 0.00	0.71 ± 0.00
Extr	Word2Vec	0.32 ± 0.01	0.30 ± 0.01	0.77 ± 0.00	0.93 ± 0.00
	GloVe	0.31 ± 0.01	0.31 ± 0.02	0.77 ± 0.01	0.93 ± 0.01
	Fusion	0.35 ± 0.01	0.35 ± 0.01	0.75 ± 0.01	0.92 ± 0.01
	BERT	**0.38 ± 0.01**	**0.37 ± 0.02**	**0.74 ± 0.00**	**0.91 ± 0.01**
Agr	Word2Vec	0.49 ± 0.02	0.43 ± 0.01	0.61 ± 0.00	0.79 ± 0.00
	GloVe	0.40 ± 0.03	0.38 ± 0.02	0.63 ± 0.01	0.81 ± 0.01
	Fusion	**0.49 ± 0.02**	**0.43 ± 0.02**	**0.60 ± 0.00**	**0.77 ± 0.01**
	BERT	0.47 ± 0.01	0.44 ± 0.01	0.62 ± 0.00	0.78 ± 0.01
Emot	Word2Vec	0.22 ± 0.03	0.18 ± 0.02	0.60 ± 0.01	0.76 ± 0.01
	GloVe	0.27 ± 0.02	0.22 ± 0.02	0.59 ± 0.00	0.75 ± 0.00
	Fusion	**0.29 ± 0.02**	**0.24 ± 0.02**	**0.59 ± 0.01**	**0.75 ± 0.01**
	BERT	0.24 ± 0.02	0.21 ± 0.02	0.59 ± 0.01	0.76 ± 0.01

Weak vs. strong presence of each trait

In this case, SVM classifiers with Gaussian kernel are used and results are reported in Table 6.2. Note that four out of the five traits (except openness to experience) obtained the best result when considering word embeddings based on BERT model. Also, note that three out of the five traits exhibit accuracies above around 60%, the best result was obtained for extraversion with an accuracy of 64.7% followed by agreeableness with an accuracy of 64.3% and conscientiousness with 63.9%. As we will see few lines below, these results are similar to most of the results reported in the literature. The results also show that there is no a clear model that leads to the best results. This means that there is still a lot of work to do in this field, which apart from the challenge of extracting information from text, imposes an additional constraint due to the consistency of the labels, i.e., the evaluation of personality is very hard task for humans and machines.

Table 6.2. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from YouTube Personality considering a SVM.

Trait	Feature	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Word2Vec	56.4 ± 1.9	51.7 ± 2.9	60.9 ± 1.9	56.3 ± 1.9	0.58 ± 0.02
	GloVe	56.4 ± 1.2	52.8 ± 2.3	59.9 ± 2.1	56.3 ± 1.3	0.58 ± 0.02
	Fusion	**56.5 ± 1.5**	**49.9 ± 2.5**	**63.0 ± 3.3**	**56.3 ± 1.5**	**0.58 ± 0.02**
	BERT	55.2 ± 1.2	48.5 ± 2.4	61.9 ± 3.8	55.0 ± 1.2	0.57 ± 0.01
Cons	Word2Vec	62.5 ± 0.8	53.6 ± 1.2	70.8 ± 1.6	62.2 ± 0.8	0.67 ± 0.01
	GloVe	63.4 ± 0.7	57.9 ± 1.2	68.6 ± 1.3	63.3 ± 0.7	0.67 ± 0.01
	Fusion	63.0 ± 1.1	66.5 ± 1.7	59.8 ± 1.4	62.9 ± 1.1	0.69 ± 0.01
	BERT	**63.6 ± 1.7**	**64.4 ± 1.8**	**62.9 ± 1.8**	**63.6 ± 1.7**	**0.68 ± 0.01**
Extr	Word2Vec	60.9 ± 0.8	53.2 ± 1.7	68.1 ± 1.6	60.7 ± 0.9	0.63 ± 0.01
	GloVe	63.8 ± 1.2	54.7 ± 1.0	72.3 ± 2.1	63.5 ± 1.2	0.67 ± 0.01
	Fusion	63.4 ± 1.1	62.0 ± 1.6	64.7 ± 1.8	63.4 ± 1.1	0.68 ± 0.01
	BERT	**64.7 ± 0.6**	**63.5 ± 0.9**	**65.8 ± 1.6**	**64.7 ± 0.6**	**0.70 ± 0.01**
Agr	Word2Vec	59.8 ± 1.4	53.3 ± 3.3	65.2 ± 1.9	59.6 ± 1.4	0.64 ± 0.01
	GloVe	60.3 ± 1.5	52.2 ± 2.3	67.2 ± 2.8	60.0 ± 1.5	0.64 ± 0.01
	Fusion	60.9 ± 1.6	56.7 ± 2.7	64.5 ± 2.4	60.8 ± 1.6	0.67 ± 0.02
	BERT	**64.3 ± 0.8**	**59.4 ± 1.7**	**68.5 ± 1.5**	**64.2 ± 0.8**	**0.69 ± 0.08**
Emot	Word2Vec	56.7 ± 1.9	52.4 ± 2.9	60.9 ± 3.5	56.6 ± 1.9	0.59 ± 0.02
	GloVe	55.5 ± 1.2	53.8 ± 1.2	57.1 ± 2.1	55.5 ± 1.2	0.57 ± 0.02
	Fusion	55.9 ± 1.1	54.2 ± 2.1	57.6 ± 1.8	55.9 ± 1.1	0.59 ± 0.02
	BERT	**56.8 ± 1.0**	**54.0 ± 1.4**	**59.6 ± 1.6**	**56.8 ± 1.0**	**0.60 ± 0.01**

Results are shown more compactly in Figure 6.1 where the ROC curves resulting from the bi-class experiments are included. Each panel in the figure includes results obtained with the three feature extraction approaches (Word2Vec, GloVe and BERT). The AUC values show that, in the majority of the cases, better results were obtained for extraversion and conscientiousness traits, and also that the best AUC values were obtained taking into

account the Fusion of Word2Vec and GloVe embeddings, and also embeddings based on BERT.

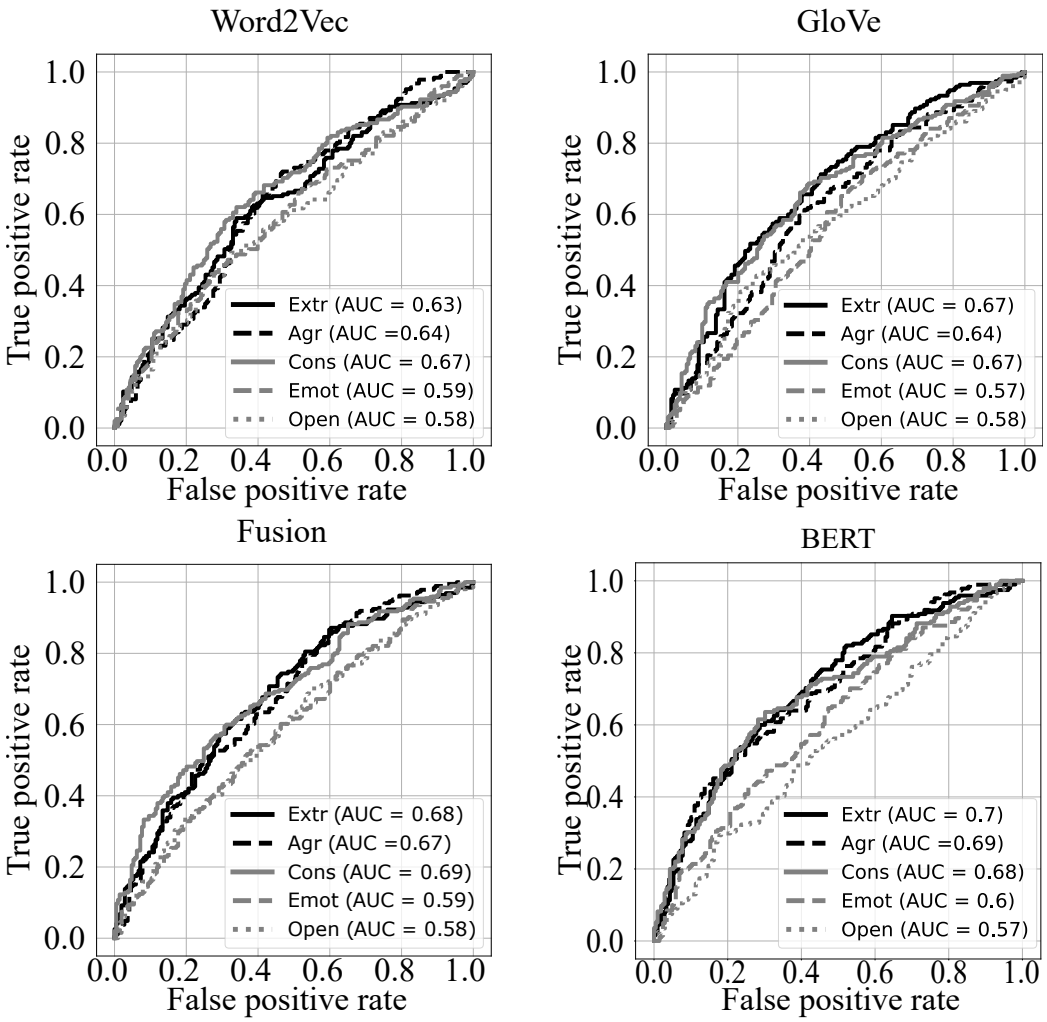

Figure 6.1. ROC curves obtained with Word2Vec, GloVe, Fusion: Word2Vec + GloVe and BERT embeddings for English dataset from YouTube Personality.

Classification of personality traits into 3 levels

Three groups were created according to the scores in the personality traits as it was explained in chapter 5: LP: low presence; MP: medium presence; and HP: high presence. Results of the tri-class classification are presented in Table 6.3 in terms of accuracy, F1-score, UAR and κ. Note that in four out

of the five traits, the best results were obtained with the fusion of Word2Vec and GloVe embeddings, only for the case of conscientiousness the best result was obtained considering only Word2Vec embeddings. It can be observed also that the models trained with BERT embeddings did not improve the performance of the models compared with the classical word embeddings (Word2Vec and GloVe) in any of the OCEAN traits. However, when having a close look at the numbers, one can notice that the difference among different approaches is not that high, and the results are similar across different traits, i.e., between 40% and 46% in accuracy and F1-score percentages.

Table 6.3. Tri-class classification results with the English dataset from YouTube Personality considering a SVM.

Trait	Feature	Accuracy	F1-score	UAR	κ
Open	Word2Vec	37.9 ± 1.9	37.6 ± 1.9	37.4 ± 1.9	0.06 ± 0.03
	GloVe	40.6 ± 2.6	40.1 ± 2.6	39.9 ± 2.6	0.09 ± 0.04
	Fusion	**41.2 ± 0.8**	**41.0 ± 0.9**	**40.8 ± 0.8**	**0.11 ± 0.01**
	BERT	35.0 ± 2.1	34.2 ± 2.1	34.3 ± 2.0	0.01 ± 0.03
Cons	**Word2Vec**	**46.6 ± 1.2**	**45.1 ± 1.2**	**45.8 ± 1.2**	**0.19 ± 0.02**
	GloVe	46.7 ± 0.8	44.9 ± 0.9	45.8 ± 0.9	0.19 ± 0.01
	Fusion	45.6 ± 1.4	45.8 ± 1.4	45.5 ± 1.4	0.18 ± 0.02
	BERT	45.5 ± 1.5	45.3 ± 1.5	45.1 ± 1.5	0.18 ± 0.02
Extr	Word2Vec	42.6 ± 1.4	42.0 ± 1.3	42.2 ± 1.3	0.13 ± 0.02
	GloVe	44.2 ± 1.4	43.5 ± 1.4	43.7 ± 1.4	0.15 ± 0.02
	Fusion	**44.3 ± 1.2**	**44.1 ± 1.2**	**44.7 ± 1.3**	**0.17 ± 0.02**
	BERT	41.8 ± 1.1	41.7 ± 1.1	41.8 ± 1.1	0.12 ± 0.02
Agr	Word2Vec	46.0 ± 1.5	45.9 ± 1.5	45.8 ± 1.5	0.19 ± 0.02
	GloVe	46.1 ± 2.0	46.0 ± 2.0	45.9 ± 2.0	0.19 ± 0.03
	Fusion	**46.2 ± 1.3**	**46.3 ± 1.3**	**46.2 ± 1.3**	**0.19 ± 0.02**
	BERT	45.9 ± 0.9	45.8 ± 0.82	45.9 ± 0.8	0.19 ± 0.01
Emot	Word2Vec	39.2 ± 1.4	38.9 ± 1.3	39.1 ± 1.4	0.09 ± 0.02
	GloVe	39.1 ± 1.3	39.0 ± 1.3	39.1 ± 1.3	0.09 ± 0.02
	Fusion	**40.4 ± 1.3**	**40.4 ± 1.3**	**40.5 ± 1.3**	**0.11 ± 0.02**
	BERT	38.3 ± 0.7	38.0 ± 0.6	38.2 ± 0.7	0.07 ± 0.01

To have a detailed look at the results, confusion matrices are presented in Table 6.4. The results do not show a clear pattern when comparing the three levels of traits. One would expect to see a relatively clear separation

between LP and HP samples; however, in all cases the target class shows the largest percentage but the remaining portion is almost always equivalently distributed between the other two classes. Although being aware that the models presented here could be improved and also acknowledging that the addressed problem is very challenging, the behavior of the observed results may be reflecting possible labeling problems. We believe that there is a big gap in the study of personality traits based on linguistic patterns, which make it necessary to work on collecting and labeling data considering the knowledge of expert psychologists and psycholinguists [35], [36].

Table 6.4. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from YouTube Personality considering a SVM (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Word2Vec	LP	36	45	19	**66**	21	13	58	31	11	56	26	18	47	32	21
	MP	35	47	18	51	**26**	23	46	32	22	25	45	30	38	37	25
	HP	32	38	30	34	20	**46**	36	28	36	23	40	37	32	34	34
GloVe	LP	44	37	19	70	18	12	60	28	12	53	25	22	42	32	26
	MP	31	48	21	53	24	23	47	33	20	27	44	29	36	34	30
	HP	28	45	27	31	25	44	34	28	38	23	36	41	29	30	41
Fusion	LP	**42**	36	22	49	35	16	**45**	33	22	**51**	26	23	**41**	33	26
	MP	29	**46**	25	31	41	28	31	**35**	34	25	**43**	32	35	**37**	28
	HP	30	36	**34**	18	35	47	19	26	**55**	19	37	**44**	27	29	**44**
BERT	LP	41	39	20	56	29	15	49	36	15	55	28	17	47	27	26
	MP	40	41	19	33	34	33	44	32	24	32	38	30	37	34	29
	HP	36	37	27	24	31	45	28	28	44	23	33	44	32	34	34

LP: Low presence, **MP:** Medium presence, **HP:** High presence.

Comparison with respect to recent works

The results reported in this study are compared with respect to different works in the state of the art. We did not find any study working on the tri-class classification problem, so comparisons are only reported for the regression and the bi-class results.

The summary of our regression results and those reported by others in the literature are included in Table 6.5. According to the average results reported in the last row of the table, our approach shows similar performance to others reported in the literature, actually our results are better according to the MAE value in about 0.02. Now, in the case of RMSE, our results are 0.03 below (worst). Although other works in the literature do not report results in terms of the Pearson's and Spearman's correlation, we decided to do it because this measure is more intuitive, especially when the "actual vs. predicted" plot is shown (see for example Figure 6.9).

Table 6.5. Comparison of our regression model w.r.t. recent works with the English dataset from YouTube Personality.

	Our approach				**[35]**	**[36]**
Trait	r	ρ	RMSE	MAE	RMSE	MAE
Open	0.22	0.21	0.70	0.56	0.68	0.58
Cons	0.40	0.41	0.71	0.55	0.69	0.57
Extr	0.38	0.37	0.91	0.74	0.89	0.72
Agre	0.49	0.43	0.77	0.60	0.77	0.67
Emot	0.29	0.24	0.75	0.59	0.69	0.60
Average	0.36	0.33	0.77	0.61	0.74	0.63

The comparison in the bi-class classification scenario is reported in Table 6.6. Note that on average, our results are slightly better than the other works except for the work in [34]. Unfortunately, the authors of that work only report the average accuracy along with the five traits, which does not allow us to make direct comparisons in specific traits. If we consider the average performance in terms of F1-score compared to the performance reported by Salminen, et al [37], we are able to improve the performance by 6.4%, which is relevant considering that in this case we did not use neural networks. This gives us an idea that for certain traits (agreeableness, conscientiousness, and emotional stability), classical methods like those used in

this work (SVM with Gaussian kernel) yield better results than those found with other methods like the one reported in [37].

Table 6.6. Comparison of our Bi-class classification model w.r.t. recent works with the English dataset from YouTube Personality.

			[32]	[33]	[34]	[37]
Trait	Acc	F1-score	F1-score	F1-score	Acc	F1-score
Open	56.5	56.3	60.8	57.3	-	68.6
Cons	63.6	63.6	65.8	54.3	-	48.5
Extr	64.7	64.7	60.5	57.8	-	71.9
Agre	64.3	64.2	65.7	69.6	-	44.4
Emot	56.8	56.8	47.7	61.9	-	40.3
Average	61.2	61.1	60.1	60.2	62.3	54.7

6.1.2 Results with the Spanish dataset from YouTube Personality

Personality trait estimation

The main idea in these experiment is to predict the score on each trait of the OCEAN model. The results are reported in Table 6.7. Note that in four out of the five traits, the transformer-based methods (BERT and BETO) yield better results. BERT seems to be the best model for conscientiousness and extraversion, while BETO works better for agreeableness and emotional stability. The only case where transformer-based methods are not the best is for openness to experience trait, where the Word2Vec model works best, and actually, it is the only one that shows positive correlation values. This is the case with the lowest correlation coefficient values ($r = 0.13$ and $\rho = 0.12$). This likely indicates that the openness to experience trait is the most difficult to predict among the five included in the OCEAN model. The other best correlation coefficients obtained for the other traits range between 0.28 and 0.45 for the Pearson's coefficient and from 0.24 to 0.40 for the Spearman's coefficient, indicating that, to some extent, it is possible to predict the level at which the participants express each trait. On the other hand, when comparing the performance of Word2Vec and GloVe, except for con-

scientiousness trait, Word2Vec model achieves better results. The regression results obtained in this study are in line with those reported in [13], where it is stated that no single model can provide good results for all five traits.

Table 6.7. Results for personality trait estimation with the Spanish dataset from YouTube Personality.

Trait	Feature	r	ρ	MAE	RMSE
Open	**Wor2Vec**	**0.13 ± 0.02**	**0.12 ± 0.03**	**0.57 ± 0.00**	**0.71 ± 0.00**
	GloVe	-0.02 ± 0.03	-0.01 ± 0.03	0.58 ± 0.00	0.72 ± 0.00
	Fusion	-0.02 ± 0.04	-0.02 ± 0.03	0.58 ± 0.00	0.72 ± 0.00
	BERT	-0.06 ± 0.03	-0.06 ± 0.04	0.58 ± 0.00	0.72 ± 0.00
	BETO	-0.05 ± 0.04	-0.05 ± 0.05	0.58 ± 0.00	0.72 ± 0.00
Cons	Wor2Vec	0.24 ± 0.02	0.23 ± 0.02	0.58 ± 0.00	0.75 ± 0.00
	GloVe	0.29 ± 0.01	0.32 ± 0.01	0.57 ± 0.00	0.74 ± 0.00
	Fusion	0.26 ± 0.01	0.28 ± 0.01	0.58 ± 0.00	0.75 ± 0.00
	BERT	**0.36 ± 0.01**	**0.38 ± 0.01**	**0.55 ± 0.00**	**0.72 ± 0.00**
	BETO	0.36 ± 0.01	0.38 ± 0.01	0.56 ± 0.00	0.72 ± 0.00
Extr	Wor2Vec	0.24 ± 0.03	0.23 ± 0.02	0.78 ± 0.01	0.95 ± 0.01
	GloVe	0.18 ± 0.03	0.18 ± 0.02	0.79 ± 0.01	0.96 ± 0.01
	Fusion	0.25 ± 0.01	0.24 ± 0.01	0.77 ± 0.00	0.95 ± 0.00
	BERT	**0.35 ± 0.02**	**0.36 ± 0.01**	**0.75 ± 0.01**	**0.92 ± 0.01**
	BETO	0.30 ± 0.01	0.30 ± 0.02	0.76 ± 0.00	0.93 ± 0.00
Agr	Wor2Vec	0.35 ± 0.02	0.30 ± 0.02	0.65 ± 0.01	0.82 ± 0.01
	GloVe	0.24 ± 0.01	0.21 ± 0.01	0.66 ± 0.00	0.85 ± 0.00
	Fusion	0.37 ± 0.01	0.32 ± 0.02	0.64 ± 0.01	0.82 ± 0.01
	BERT	0.41 ± 0.02	0.38 ± 0.02	0.63 ± 0.01	0.80 ± 0.01
	BETO	**0.45 ± 0.01**	**0.40 ± 0.01**	**0.62 ± 0.00**	**0.79 ± 0.01**
Emot	Wor2Vec	0.27 ± 0.03	0.22 ± 0.02	0.59 ± 0.00	0.75 ± 0.00
	GloVe	0.02 ± 0.03	0.01 ± 0.04	0.61 ± 0.00	0.79 ± 0.01
	Fusion	0.16 ± 0.04	0.12 ± 0.04	0.61 ± 0.01	0.77 ± 0.01
	BERT	0.19 ± 0.03	0.18 ± 0.03	0.60 ± 0.01	0.77 ± 0.01
	BETO	**0.28 ± 0.01**	**0.24 ± 0.02**	**0.59 ± 0.01**	**0.75 ± 0.00**

Weak vs. strong presence of each trait

Table 6.8 shows the results of the bi-class classification experiments. Note that the behavior is similar to the one observed in the regression results, where transformer-based models outperformed other word embeddings models. The trait with the best results was agreeableness, with an accuracy of 63.9% obtained with the BERT model. Conversely, the trait with the lowest accuracy is openness (as it was the case in the regression experiments) with an accuracy of 52.3%. When comparing BERT and BETO results, we realize that they are quite similar in all of the traits. Additionally, the fusion of embeddings is not showing significant improvements with respect to individual models.

Table 6.8. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from YouTube Personality.

Trait	Feature	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Word2Vec	52.0 ± 1.6	53.1 ± 2.6	50.9 ± 2.0	52.0 ± 1.6	0.53 ± 0.02
	GloVe	50.2 ± 2.3	46.7 ± 4.7	53.7 ± 5.2	50.1 ± 2.3	0.50 ± 0.02
	Fusion	49.5 ± 1.0	45.5 ± 3.1	53.5 ± 2.9	49.4 ± 1.0	0.50 ± 0.01
	BERT	**52.3 ± 1.7**	**49.9 ± 3.3**	**54.6 ± 3.1**	**52.2 ± 1.7**	**0.53 ± 0.02**
	BETO	51.0 ± 1.1	49.1 ± 3.0	52.8 ± 3.6	50.9 ± 1.1	0.51 ± 0.01
Cons	Word2Vec	57.7 ± 1.3	54.9 ± 2.3	60.4 ± 2.0	57.7 ± 1.3	0.59 ± 0.01
	GloVe	59.7 ± 1.0	66.9 ± 2.1	53.0 ± 1.7	59.5 ± 1.0	0.64 ± 0.01
	Fusion	60.3 ± 0.9	65.9 ± 2.6	55.1 ± 1.9	60.2 ± 0.8	0.64 ± 0.01
	BERT	60.0 ± 1.2	57.8 ± 1.6	62.0 ± 2.8	60.0 ± 1.2	0.67 ± 0.01
	BETO	**61.3 ± 1.2**	**60.4 ± 1.9**	**62.1 ± 1.5**	**61.3 ± 1.2**	**0.65 ± 0.01**
Extr	Word2Vec	58.5 ± 1.3	50.4 ± 2.1	66.0 ± 2.1	58.2 ± 1.3	0.61 ± 0.01
	GloVe	54.7 ± 1.3	52.9 ± 1.9	56.4 ± 1.9	54.7 ± 1.3	0.57 ± 0.01
	Fusion	57.5 ± 2.1	56.2 ± 2.6	58.8 ± 2.7	57.5 ± 2.1	0.62 ± 0.02
	BERT	**62.1 ± 1.1**	**66.9 ± 1.4**	**57.6 ± 1.8**	**62.0± 1.2**	**0.66 ± 0.01**
	BETO	59.3 ± 1.0	64.5 ± 2.6	54.4 ± 2.0	59.2 ± 1.0	0.65 ± 0.01
Agr	Word2Vec	58.8 ± 0.7	42.3 ± 1.7	72.9 ± 1.7	57.8 ± 0.7	0.63 ± 0.01
	GloVe	56.0 ± 1.5	48.5 ± 2.1	62.4 ± 2.6	55.9 ± 1.5	0.58 ± 0.01
	Fusion	57.7 ± 1.2	50.9 ± 2.7	63.6 ± 1.3	57.6 ± 1.3	0.63 ± 0.01
	BERT	**63.9 ± 0.9**	**54.3 ± 1.8**	**72.0 ± 1.3**	**63.6 ± 0.9**	**0.69 ± 0.01**
	BETO	60.0 ± 1.3	52.6 ± 2.1	66.2 ± 1.4	59.8 ± 1.3	0.64 ± 0.01
Emot	Word2Vec	56.3 ± 1.1	55.4 ± 1.7	57.1 ± 1.9	56.3 ± 1.1	0.57 ± 0.01
	GloVe	52.2 ± 1.9	48.0 ± 3.2	56.3 ± 3.2	52.0 ± 1.9	0.52 ± 0.02
	Fusion	53.5 ± 2.6	50.2 ± 2.5	56.7 ± 4.3	53.4 ± 2.5	0.55 ± 0.02
	BERT	54.1 ± 1.8	53.8 ± 2.4	54.3 ± 3.9	54.1 ± 1.8	0.55 ± 0.02
	BETO	**57.3 ± 0.7**	**55.0 ± 1.5**	**59.6 ± 2.1**	**57.3 ± 0.7**	**0.61 ± 0.01**

In Figure 6.2 it is shown the ROC curves resulting from the bi-class

classification experiments. We can observe a general behavior: the best performance with respect to the area under the curve (AUC) is generally obtained with BERT's word embeddings, since, in 4 of the five traits (except emotional stability), it has the highest AUC value. Following BERT, the BETO model provides some of the best performances, and after it, the fusion of Word2Vec and GloVe embeddings.

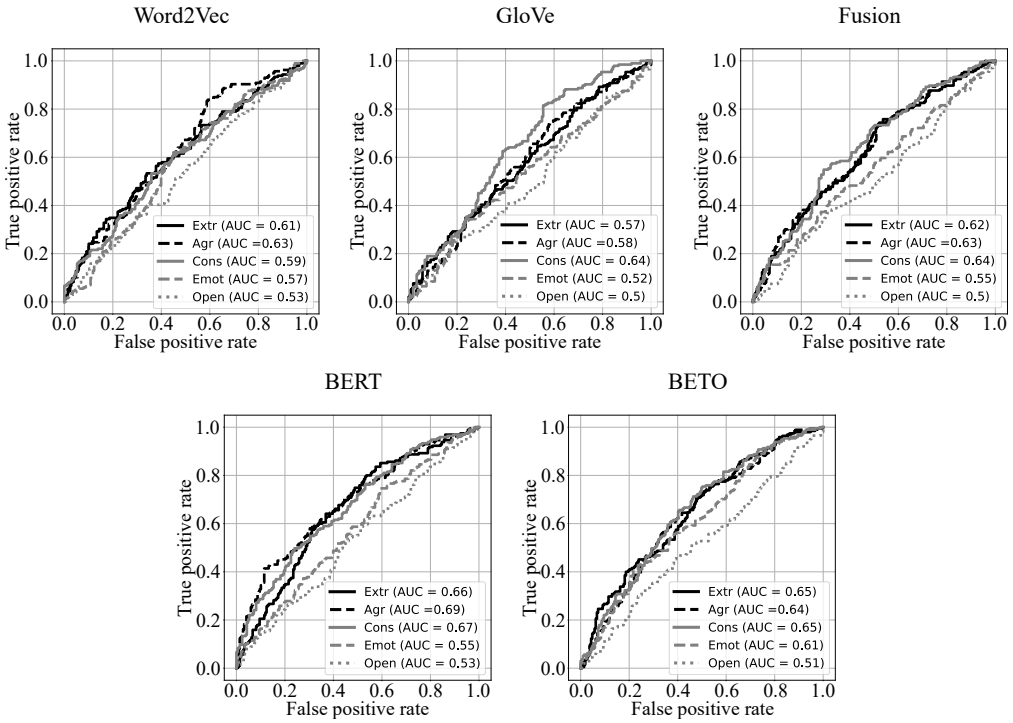

Figure 6.2. ROC curves obtained with Word2Vec, GloVe, Fusion: Word2Vec + GloVe, BERT and BETO embeddings for Spanish dataset from YouTube Personality.

Classification of personality traits into 3 levels

Table 6.9 shows the results of the experiment that classifies the personality traits into 3 different levels for Spanish language. Comparing these results with those obtained in English language, it is observed that for conscientiousness, extraversion and agreeableness traits, the reported metrics are close by approximately 2%, while for the traits openness to experience and emotional stability, the results in Spanish are notoriously lower than those obtained in English. The best result for this experiment is presented for agreeableness

trait taking into account the embeddings obtained with the BERT model, obtaining a maximum F1-score of 45.3% and a Cohen's kappa coefficient $\kappa = 0.18$. The second best result was obtained for the conscientiousness trait with a maximum F1-score of 44.7% and a $\kappa = 0.17$ and the lowest result was obtained for the emotional stability trait (F1-score of 38.1% and $\kappa = 0.07$).

Table 6.9. Tri-class classification results with the Spanish dataset from YouTube Personality.

Trait	Feature	Accuracy	F1-score	UAR	κ
Open	Word2Vec	34.5 ± 1.3	33.0 ± 1.6	33.3 ± 1.3	0.00 ± 0.02
	GloVe	**36.8 ± 1.0**	**36.0 ± 1.0**	**35.9 ± 1.0**	**0.04 ± 0.02**
	Fusion	34.1 ± 1.7	33.5 ± 1.7	33.4 ± 1.7	0.00 ± 0.03
	BERT	34.0 ± 1.8	32.8 ± 1.9	33.1 ± 1.8	-0.00 ± 0.03
	BETO	33.6 ± 1.3	31.5 ± 1.6	32.4 ± 1.3	-0.01 ± 0.02
Cons	Word2Vec	40.1 ± 1.2	39.9 ± 1.3	39.8 ± 1.3	0.10 ± 0.02
	GloVe	43.3 ± 1.5	42.9 ± 1.6	42.7 ± 1.5	0.15 ± 0.02
	Fusion	42.7 ± 0.8	42.4 ± 0.8	42.3 ± 0.8	0.14 ± 0.01
	BERT	**45.2 ± 2.0**	**44.8 ± 1.9**	**44.7 ± 2.0**	**0.17 ± 0.03**
	BETO	41.7 ± 1.2	41.2 ± 1.2	41.3 ± 1.2	0.12 ± 0.02
Extr	**Word2Vec**	**42.4 ± 1.8**	**41.4 ± 1.7**	**41.7 ± 1.7**	**0.13 ± 0.03**
	GloVe	38.4 ± 1.9	38.4 ± 1.9	38.5 ± 1.9	0.08 ± 0.03
	Fusion	41.8 ± 1.5	41.8 ± 1.5	41.7 ± 1.5	0.12 ± 0.02
	BERT	38.9 ± 1.4	38.8 ± 1.5	38.7 ± 1.5	0.08 ± 0.02
	BETO	41.8 ± 0.8	41.7 ± 0.8	41.6 ± 0.8	0.12 ± 0.01
Agr	Word2Vec	41.5 ± 1.4	41.6 ± 1.4	41.4 ± 1.4	0.12 ± 0.02
	GloVe	41.5 ± 1.0	41.4 ± 1.1	41.3 ± 1.0	0.12 ± 0.01
	Fusion	43.6 ± 1.6	43.2 ± 1.7	43.3 ± 1.6	0.15 ± 0.02
	BERT	**45.5 ± 1.7**	**45.2 ± 1.7**	**45.3 ± 1.7**	**0.18 ± 0.03**
	BETO	43.0 ± 1.1	42.9 ± 1.2	42.9 ± 1.1	0.14 ± 0.02
Emot	Word2Vec	33.9 ± 1.5	33.1 ± 1.5	33.8 ± 1.5	0.01 ± 0.02
	GloVe	31.5 ± 1.0	31.3 ± 1.0	31.5 ± 1.0	-0.03 ± 0.01
	Fusion	32.4 ± 2.1	32.2 ± 2.2	32.3 ± 2.1	-0.01 ± 0.03
	BERT	37.2 ± 1.0	36.5 ± 1.0	37.0 ± 1.0	0.06 ± 0.02
	BETO	**38.1 ± 1.5**	**36.9 ± 1.4**	**37.9 ± 1.5**	**0.07 ± 0.02**

The confusion matrices for the tri-class experiment are shown in the Table 6.10. As can be seen, the results obtained with embeddings based on models with transformers (i.e. BERT and BETO) yield the best results, because in 3 out of 5 traits, gives the best percentages in the confusion matrices.

Table 6.10. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from YouTube Personality (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Word2Vec	LP	30	48	22	49	31	20	**60**	26	14	45	32	23	48	32	20
	MP	28	52	20	42	29	29	46	**36**	18	28	42	30	38	32	30
	HP	24	58	18	27	31	42	41	30	**29**	19	44	37	35	44	21
GloVe	LP	**34**	42	24	58	24	18	37	36	27	48	31	21	35	35	30
	MP	32	**51**	17	34	31	35	26	40	34	32	41	27	35	30	35
	HP	37	40	**23**	24	36	40	26	34	40	25	40	35	35	35	30
Fusion	LP	30	45	25	55	28	17	45	33	22	49	34	17	35	36	29
	MP	34	46	20	37	31	32	36	39	25	29	49	22	31	33	36
	HP	33	43	24	26	33	41	27	31	42	23	45	32	28	44	28
BERT	LP	33	45	22	**58**	28	14	46	35	19	**57**	28	15	52	29	19
	MP	32	47	21	39	**33**	28	42	31	27	30	**41**	29	37	34	29
	HP	32	49	19	24	33	**43**	27	34	39	28	34	**38**	34	40	26
BETO	LP	34	48	18	54	27	19	48	34	18	52	28	20	**57**	27	16
	MP	38	49	13	36	26	38	41	39	20	31	40	29	34	**35**	31
	HP	36	50	14	22	33	45	30	32	38	23	40	37	33	46	**21**

LP: Low presence, **MP:** Medium presence, **HP:** High presence.

6.1.3 Results with the English dataset from PAN-AP-2015

Now, in order to test the proposed methodology with a different database, we used the English PAN-AP-2015 database from Twitter explained in 3.2, taking into account the database per subject (152 samples for training and 142 for testing), where the reported metrics correspond to the performance of the models in the test set.

Personality trait estimation

The results of the estimation of the 5 personality traits of the OCEAN model (regression experiments) are in the Table 6.11, which correspond to the results obtained on the test set originally provided in the database in order to compare them with other results of works reported in the literature. From Table 6.11 we can see that no single group of features provides the best result for the five traits, since for openness to experience and agreeableness the best result was obtained with GloVe embeddings; for conscientiousness and extraversion, the best result was obtained with Fusion group of features (Word2Vec + GloVe embeddings) and for emotional stability, the best result was obtained with BERT embeddings. It should also be noted that for openness to experience, conscientiousness and emotional stability traits, there are strong correlations between the actual labels and the labels predicted by the regressor ($r, \rho > 0.6$). With respect to MAE and RMSE, the best (lowest) values were obtained for conscientiousness trait with MAE = 0.095 and RMSE = 0.118, while the worst (highest) values were obtained for emotional stability trait with MAE = 0.155 and RMSE = 0.184.

Table 6.11. Results for personality trait estimation with the English dataset from PAN-AP-2015 considering a SVR.

Trait	Feature	r	ρ	MAE	RMSE
Open	Word2Vec	0.586	0.561	0.104	0.128
	GloVe	**0.649**	**0.635**	**0.096**	**0.119**
	Fusion	0.655	0.627	0.099	0.121
	BERT	0.635	0.614	0.105	0.125
Cons	Word2Vec	0.542	0.466	0.102	0.125
	GloVe	0.595	0.622	0.095	0.121
	Fusion	**0.667**	**0.624**	**0.095**	**0.118**
	BERT	0.578	0.560	0.102	0.127
Extr	Word2Vec	0.460	0.414	0.118	0.147
	GloVe	0.491	0.442	0.109	0.138
	Fusion	**0.546**	**0.485**	**0.108**	**0.136**
	BERT	0.544	0.503	0.109	0.136
Agr	Word2Vec	0.403	0.397	0.110	0.143
	GloVe	**0.449**	**0.375**	**0.104**	**0.137**
	Fusion	0.444	0.384	0.105	0.137
	BERT	0.403	0.414	0.105	0.140
Emot	Word2Vec	0.519	0.496	0.163	0.197
	GloVe	0.543	0.535	0.158	0.193
	Fusion	0.567	0.560	0.159	0.191
	BERT	**0.646**	**0.632**	**0.155**	**0.184**

Weak vs. strong presence of each trait

The result of the binary classifiers taking into account the SVMs with Gaussian kernel are shown in Table 6.12. In this case, the word embeddings obtained with GloVe provided the best result for the traits conscientiousness, agreeableness and emotional stability; for the remaining traits (openness to experience and extraversion) the best result was obtained with the fusion of Word2Vec and GloVe embeddings. The best result in this bi-class classification scenario was obtained for conscientiousness trait with an accuracy of 82.4% followed by openness to experience trait (80.3%). The extraversion and emotional stability traits have a similar performance (70.1% approximately) and the lowest result is for agreeableness trait, with an accuracy percentage of 62.7%.

Table 6.12. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from PAN-AP-2015.

Trait	Feature	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Word2Vec	77.5	81.1	70.2	77.7	0.84
	GloVe	77.5	97.9	36.2	74.1	0.88
	Fusion	**80.3**	**88.4**	**63.8**	**79.9**	**0.89**
	BERT	78.9	91.6	53.2	77.7	0.89
Cons	Word2Vec	73.9	72.6	75.4	73.9	0.81
	GloVe	**82.4**	**86.3**	**78.3**	**82.4**	**0.89**
	Fusion	79.6	80.8	78.3	79.6	0.88
	BERT	79.6	80.8	78.3	79.6	0.86
Extr	Word2Vec	59.9	74.0	44.9	59.0	0.72
	GloVe	70.4	83.4	56.5	69.8	0.76
	Fusion	**71.1**	**69.9**	**72.5**	**71.1**	**0.78**
	BERT	68.3	75.3	60.9	68.1	0.78
Agr	Word2Vec	54.9	90.4	17.4	47.9	0.57
	GloVe	**62.7**	**95.9**	**27.5**	**57.6**	**0.58**
	Fusion	59.2	95.9	20.3	52.2	0.58
	BERT	58.5	97.3	17.4	50.4	0.55
Emot	Word2Vec	70.4	80.2	52.9	70.0	0.72
	GloVe	**71.1**	**80.2**	**54.9**	**70.8**	**0.75**
	Fusion	64.8	65.9	62.7	65.4	0.71
	BERT	66.9	72.5	56.9	67.1	0.75

The ROC curves from this bi-class classification experiment are shown in Figure 6.3. Regarding the performance in terms of AUC, we observed a similar behavior for the traits openness to experience and conscientiousness, being generally higher the values obtained for the first mentioned trait. Now, regarding the word embeddings with the best performance also in terms of AUC, GloVe or Fusion embeddings generally obtained the best performances for all five traits. The lowest performance (also in terms of AUC value) was

obtained for the agreeableness trait and for the word embeddings obtained with Word2Vec.

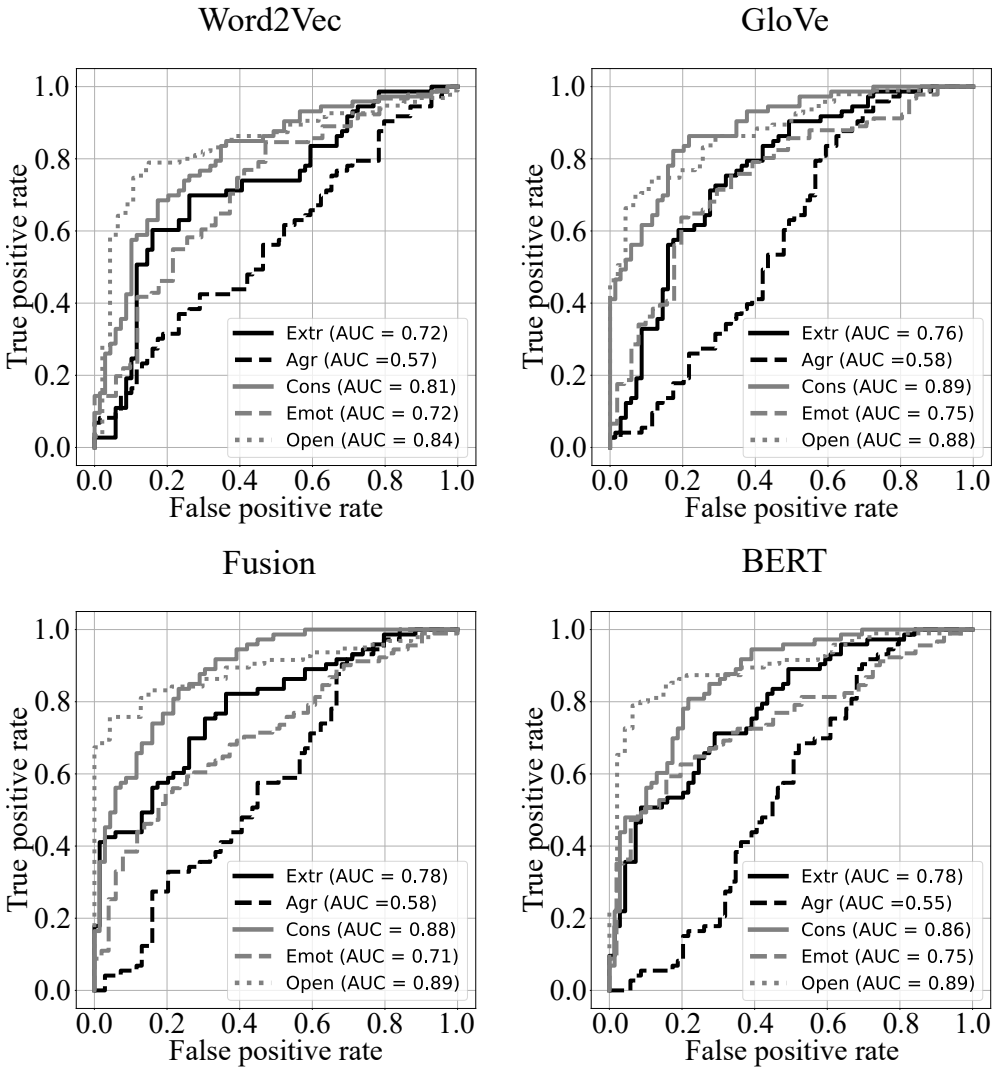

Figure 6.3. ROC curves obtained with Word2Vec, GloVe, Fusion: Word2Vec + GloVe and BERT embeddings for English dataset from PAN-AP-2015.

Classification of personality traits into 3 levels

The results of the tri-class classification experiment (LP vs MP vs HP) are presented in Table 6.13 and Table 6.14 in therms of accuracy, F1-score, UAR, κ and confusion matrix. Note that, as in the bi-class classification experi-

ments, the best results were obtained with GloVe or Fusion embeddings, but, contrary to what was obtained in the regression and bi-class classification experiments, in this experiment the trait agreeableness obtained the best performance (Accuracy = 70.4%, F1-score = 69.7% and $\kappa = 0.49$), for the other four traits, the accuracy percentages range from 52.1% and 66.9%.

Table 6.13. Tri-class classification results with the English dataset from PAN-AP-2015.

Trait	Feature	Accuracy	F1-score	UAR	κ
Open	Word2Vec	59.9	58.7	59.6	0.40
	GloVe	53.5	50.0	52.8	0.30
	Fusion	**63.4**	**62.7**	**63.1**	**0.45**
	BERT	60.6	58.1	59.9	0.40
Cons	Word2Vec	54.2	54.5	58.5	0.31
	GloVe	**59.9**	**60.5**	**64.0**	**0.39**
	Fusion	59.2	59.9	62.4	0.37
	BERT	50.7	51.6	53.5	0.26
Extr	Word2Vec	57.7	56.4	50.4	0.28
	GloVe	**66.9**	**65.9**	**60.7**	**0.43**
	Fusion	63.4	61.6	55.2	0.36
	BERT	62.0	57.6	49.9	0.30
Agr	Word2Vec	59.9	59.0	57.0	0.34
	GloVe	65.5	65.5	63.9	0.43
	Fusion	68.3	67.8	63.7	0.46
	BERT	**70.4**	**69.7**	**65.0**	**0.49**
Emot	Word2Vec	48.6	44.3	48.0	0.22
	GloVe	**52.1**	**47.4**	**51.8**	**0.27**
	Fusion	50.7	46.0	50.1	0.25
	BERT	49.3	42.2	48.2	0.22

From Table 6.14 we can see that the best values in the confusion matrices were obtained with the GloVe embeddings and that the BERT embeddings allowed to classify with a high percentage ($> 85\%$) the subjects belonging to the HP class in four of the five traits (except for conscientiousness trait).

Table 6.14. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from PAN-AP-2015 (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Word2Vec	LP	72	7	21	66	17	17	29	21	50	35	6	59	44	0	56
	MP	24	37	39	10	68	22	10	49	41	0	69	31	33	18	49
	HP	16	14	70	8	49	43	5	21	74	16	16	68	17	0	83
GloVe	LP	55	0	45	**69**	21	10	**39**	7	54	65	0	35	**51**	0	49
	MP	17	18	65	5	**75**	20	5	**63**	32	0	57	43	23	**18**	59
	HP	14	0	86	5	47	**48**	4	16	**80**	10	20	70	13	0	**87**
Fusion	LP	**72**	0	28	65	21	14	32	14	54	56	3	41	46	0	54
	MP	15	**41**	44	5	73	22	2	51	47	0	54	46	25	18	57
	HP	14	10	**76**	4	47	49	1	17	82	7	12	80	13	0	87
BERT	LP	66	4	30	48	45	7	18	11	71	**53**	6	41	38	0	62
	MP	11	26	63	5	73	22	5	41	54	0	**57**	43	25	12	63
	HP	10	2	88	4	56	40	1	8	91	4	11	**85**	6	0	94

LP: Low presence, **MP:** Medium presence, **HP:** High presence.

Comparison with respect to works in the literature

Now, with the aim of comparing our work with others found in the literature, we will show a comparison of our results with those obtained by different works mentioned in 1.2.5. In Table 6.15 we can see the results of our work and the results in different papers reported in the literature. The work done in [38] ranked first in the Author Profiling Task at PAN 2015 [8] taking into account English language Tweets, obtaining also the best performance for agreeableness and openness to experience traits. In the work [77], the authors achieved the best performance for emotional stability trait, in [78] it was obtained the best performance for extraversion trait and in [79], the authors obtained the best performance for conscientiousness trait. Our work

outperforms with respect to the average RMSE value for the five traits of the OCEAN model all the aforementioned works and also the performance of the works done in [18], [19], [39] improving the average RMSE for the five traits by 0.005. Also, our work improves over previous works by 0.001 and 0.011 the RMSE value for openness to experience and emotional stability traits respectively.

Table 6.15. Comparison of our regression model w.r.t. recent works with the English dataset from PAN-AP-2015.

	Our approach				**[38]**	**[18]**	**[19]**	**[39]**	**[77]**	**[78]**	**[79]**
Trait	r	ρ	MAE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
Open	0.649	0.635	0.096	**0.119**	0.120	0.215	0.158	0.148	0.125	0.123	0.142
Cons	0.667	0.624	0.095	0.118	0.117	0.196	0.148	0.144	0.130	0.133	**0.110**
Extr	0.546	0.485	0.108	0.136	0.128	0.213	0.164	0.158	0.132	**0.125**	0.130
Agre	0.449	0.375	0.104	0.137	**0.131**	0.215	0.151	0.150	0.140	0.132	0.148
Emot	0.646	0.632	0.155	**0.184**	0.225	0.317	0.235	0.212	0.195	0.225	0.215
Average	0.591	0.550	0.112	**0.139**	0.144	0.231	0.171	0.162	0.144	0.148	0.149

6.1.4 Results with the Spanish dataset from PAN-AP-2015

In this case, similar to the English Tweets, the database per subject was taken into account (92 samples for training and 79 for testing), where the reported metrics correspond to the performance of the models on the test set.

Personality trait estimation

In Table 6.16 we can observe the performance of the regression models for the Twitter database in Spanish language. For 3 of the 5 traits (openness to experience, conscientiousness and extraversion) the best result was obtained with the word embeddings of transformer-based models (i.e. BERT or BETO). The best result was obtained for the extraversion trait with $r = 0.801$, $\rho = 0.819$, MAE $= 0.107$ and RMSE $= 0.139$. It should also be noted that for the 5 traits there is a moderate to strong correlation ($r, \rho > 0.5$) and that the MAE and RMSE values are in the ranges $0.095 - 0.137$ and $0.118 - 0.173$ respectively.

Table 6.16. Results for personality trait estimation with the Spanish dataset from PAN-AP-2015.

Trait	Feature	r	ρ	MAE	RMSE
Open	Wor2Vec	0.448	0.400	0.103	0.133
	GloVe	0.511	0.530	0.100	0.130
	Fusion	0.393	0.366	0.101	0.134
	BERT	**0.542**	**0.500**	**0.095**	**0.118**
	BETO	0.541	0.479	0.095	0.119
Cons	Wor2Vec	0.663	0.677	0.101	0.127
	GloVe	0.554	0.551	0.119	0.144
	Fusion	0.650	0.665	0.106	0.129
	BERT	0.719	0.718	0.102	0.126
	BETO	**0.717**	**0.703**	**0.099**	**0.126**
Extr	Wor2Vec	0.732	0.771	0.109	0.142
	GloVe	0.540	0.582	0.134	0.177
	Fusion	0.735	0.756	0.111	0.144
	BERT	0.774	0.778	0.111	0.146
	BETO	**0.801**	**0.819**	**0.107**	**0.139**
Agr	Wor2Vec	0.691	0.679	0.106	0.134
	GloVe	0.517	0.528	0.121	0.156
	Fusion	**0.682**	**0.693**	**0.105**	**0.134**
	BERT	0.726	0.714	0.112	0.140
	BETO	0.707	0.695	0.111	0.139
Emot	**Wor2Vec**	**0.579**	**0.561**	**0.137**	**0.173**
	GloVe	0.358	0.318	0.165	0.203
	Fusion	0.540	0.516	0.144	0.179
	BERT	0.468	0.485	0.152	0.188
	BETO	0.524	0.534	0.149	0.184

Weak vs. strong presence of each trait

The results of the bi-class classification system between weak presence and strong presence of the 5 personality traits of the OCEAN model taking into account the Spanish language Tweets database can be found in Table 6.17. In this case, there is no clear word embeddings model that provides the best result for the five personality traits, since for 2 traits (conscientiousness and extraversion) the best result was obtained with BERT, for other 2 (agreeableness and emotional stability) the best result was achieved with the fusion of Word2Vec and GloVe, and for the missing trait (openness to experience) using GloVe embeddings leads to the best result. For extraversion, agreeableness and emotional stability traits, an accuracy percentage $> 80\%$ was achieved, which is a promising result for the task of automatic recognition of personality traits through Spanish language texts; while for the traits openness to experience and conscientiousness, the accuracy percentage is $\leq 61\%$, where a kind of overfitting towards one of the two classes is observed, as for example in the case of openness to experience trait, where the results show that the classifier is good at discriminating the subjects belonging to the class 'strong presence' of the trait but is bad at classifying the subjects belonging to the class 'weak presence' of the trait, which may be due to an imbalance between the number of samples for both classes when training the models (see Table 5.3).

Table 6.17. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from PAN-AP-2015.

Trait	Feature	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Word2Vec	53.2	97.5	7.7	41.2	0.52
	GloVe	**55.7**	**92.5**	**17.9**	**48.5**	**0.59**
	Fusion	53.2	92.5	12.8	44.3	0.58
	BERT	51.9	97.5	5.1	38.7	0.58
	BETO	53.2	100.0	5.1	39.4	0.57
Cons	Word2Vec	59.5	51.8	78.3	61.1	0.71
	GloVe	54.4	44.6	78.3	55.8	0.66
	Fusion	59.5	50.0	82.6	60.9	0.72
	BERT	**60.8**	**50.0**	**87.0**	**62.0**	**0.73**
	BETO	59.5	51.8	78.2	61.1	0.73
Extr	Word2Vec	82.3	85.4	78.9	82.2	0.91
	GloVe	86.1	95.1	76.3	85.9	0.93
	Fusion	86.1	90.2	81.6	86.0	0.93
	BERT	**87.3**	**92.7**	**81.6**	**87.3**	**0.95**
	BETO	**87.3**	**92.7**	**81.6**	**87.3**	**0.95**
Agr	Word2Vec	78.5	78.3	78.8	78.6	0.88
	GloVe	77.2	80.4	72.7	77.2	0.79
	Fusion	**82.3**	**84.8**	**78.8**	**82.3**	**0.89**
	BERT	81.0	95.7	60.6	80.1	0.85
	BETO	77.2	87.0	63.6	76.8	0.83
Emot	Word2Vec	79.7	78.6	81.1	79.8	0.88
	GloVe	67.1	66.7	67.6	67.1	0.76
	Fusion	**83.5**	**85.7**	**81.1**	**83.5**	**0.87**
	BERT	82.3	85.7	78.4	82.2	0.88
	BETO	82.3	85.7	78.4	82.2	0.90

In Figure 6.4 we can observe the ROC curves for this bi-class classification experiment with respect to different word embeddings and different personality traits. As can be seen, the best performance in terms of AUC values for 3 of the 5 traits (conscientiousness, extraversion and emotional stability) were obtained with the embeddings obtained from transformer-based

models (BERT or BETO), and in general, comparing the performance in terms of AUC of the five personality traits, the best performing traits were extraversion, agreeableness and emotional stability (in that respective order).

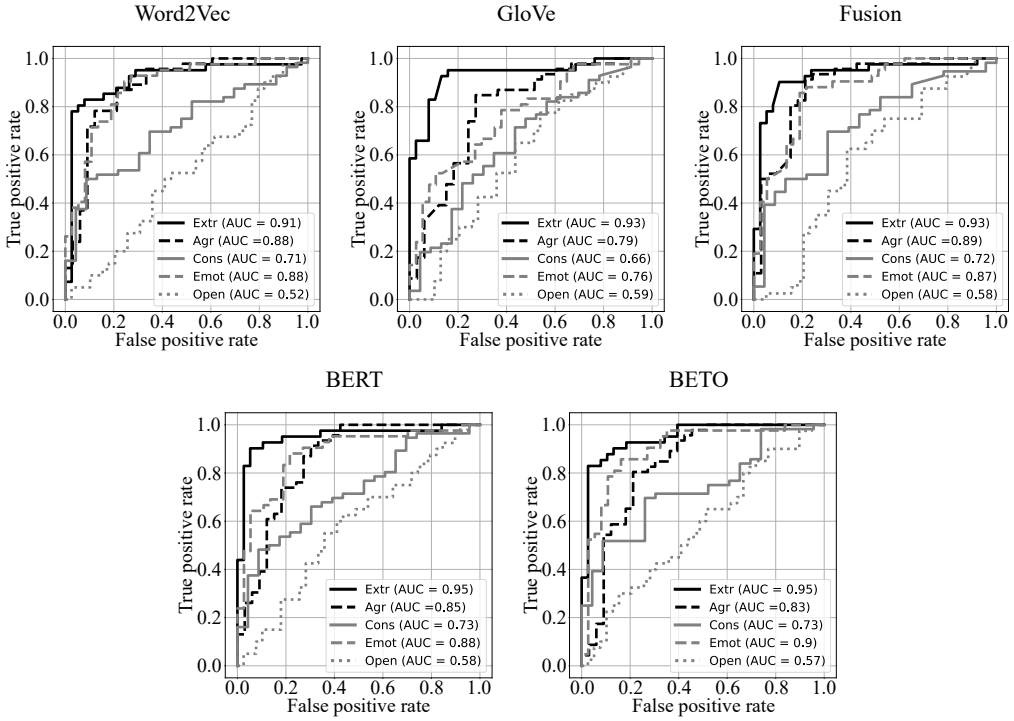

Figure 6.4. ROC curves obtained with Word2Vec, GloVe, Fusion: Word2Vec + GloVe, BERT and BETO embeddings for Spanish dataset from PAN-AP-2015.

Classification of personality traits into 3 levels

In Table 6.18 are the results of the LP vs MP vs HP classification of personality traits. Similar to the bi-class classification case, there is no single word embeddings model that serves to obtain the best result for all five traits. The best result was obtained for extraversion trait with a F1-score of 80.2% and $\kappa = 0.69$, followed by agreeableness trait with F1-score = 73.4% and $\kappa = 0.59$. For the other 3 traits, promising results were also obtained in this task of classifying the presence of the trait into 3 levels (accuracy $\geq 67\%$).

Table 6.18. Tri-class classification results with the Spanish dataset from PAN-AP-2015.

Trait	Feature	Accuracy	F1-score	UAR	κ
Open	Word2Vec	62.0	59.0	56.3	0.35
	GloVe	53.2	49.7	47.0	0.22
	Fusion	**67.1**	**62.6**	**59.9**	**0.42**
	BERT	62.0	54.5	50.5	0.29
	BETO	65.8	62.0	56.0	0.38
Cons	**Word2Vec**	**69.6**	**70.3**	**62.9**	**0.47**
	GloVe	67.1	67.0	57.7	0.41
	Fusion	69.6	69.6	62.2	0.47
	BERT	64.6	60.1	52.6	0.36
	BETO	68.4	64.0	61.5	0.43
Extr	Word2Vec	77.2	76.4	72.0	0.63
	GloVe	69.6	67.1	61.4	0.49
	Fusion	79.7	78.7	74.3	0.67
	BERT	**81.0**	**80.2**	**76.2**	**0.69**
	BETO	81.0	79.4	74.8	0.69
Agr	Word2Vec	69.6	69.3	68.8	0.53
	GloVe	60.8	59.3	57.3	0.38
	Fusion	**73.4**	**73.1**	**71.7**	**0.59**
	BERT	67.1	66.9	65.3	0.49
	BETO	68.4	68.5	67.3	0.51
Emot	**Word2Vec**	**72.2**	**71.3**	**68.6**	**0.56**
	GloVe	54.4	53.6	52.1	0.29
	Fusion	64.6	63.6	61.4	0.44
	BERT	65.8	63.7	60.5	0.45
	BETO	65.8	64.3	61.2	0.45

Table 6.19 shows the confusion matrices for this tri-class experiment. It can be seen that the classifier was able to discriminate subjects in the 'LP' class (low presence of the trait) with high classification rates (from 70% to 100%). The best result in terms of confusion matrix was obtained for extraversion trait, followed by agreeableness and emotional stability traits.

Table 6.19. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from PAN-AP-2015 (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Word2Vec	LP	82	3	15	**87**	11	2	92	5	3	79	15	6	**86**	11	3
	MP	71	24	5	33	**67**	0	33	50	17	27	58	15	12	**76**	12
	HP	37	0	63	17	48	**35**	13	13	74	20	10	70	22	33	**45**
GloVe	LP	74	8	18	87	9	4	97	0	3	73	27	0	71	18	11
	MP	57	14	29	44	56	0	44	39	17	31	69	0	33	52	15
	HP	42	5	53	22	48	30	26	26	48	30	40	30	17	50	33
Fusion	LP	**92**	0	8	89	9	2	95	0	5	**85**	12	3	79	21	0
	MP	81	**19**	0	33	67	0	33	50	17	23	**65**	12	21	67	12
	HP	32	0	**68**	17	52	31	13	9	78	15	20	**65**	22	39	39
BERT	LP	100	0	0	94	6	0	**94**	3	3	73	27	0	75	21	4
	MP	90	10	0	44	56	0	28	**55**	17	15	73	12	21	79	0
	HP	58	0	42	26	65	9	13	9	**78**	25	25	50	22	50	28
BETO	LP	97	0	3	94	4	2	97	0	1	73	27	0	71	25	4
	MP	71	29	0	22	78	0	33	45	22	23	69	8	21	79	0
	HP	58	0	42	30	57	13	13	4	83	30	10	60	17	50	33

LP: Low presence, **MP:** Medium presence, **HP:** High presence.

Comparison with respect to works in the literature

Similar to the comparison made with the English Tweets, in Table 6.20 it is shown the comparison of our results taking into account the Tweets in Spanish language with those results obtained in other works reported in the literature. As can be seen, our results could not outperform the results of the works reported in [38] and [17], but our work improved the results of the works reported in [18] and [19] as follows: the average RMSE value for the

five traits improved by 0.044 and the improvement of the RMSE value for the OCEAN traits was by 0.029, 0.015, 0.059, 0.009, 0.04 respectively.

Table 6.20. Comparison of our regression model w.r.t. recent works with the Spanish dataset from PAN-AP-2015.

	Our approach				**[38]**	**[17]**	**[18]**	**[19]**
Trait	r	ρ	MAE	RMSE	RMSE	RMSE	RMSE	RMSE
Open	0.542	0.500	0.095	0.118	0.126	**0.111**	0.214	0.147
Cons	0.717	0.703	0.099	0.126	0.117	**0.102**	0.141	0.179
Extr	0.801	0.819	0.107	0.139	**0.132**	0.137	0.279	0.198
Agr	0.682	0.693	0.105	0.134	0.111	**0.103**	0.143	0.173
Emot	0.579	0.561	0.137	0.173	**0.163**	0.164	0.281	0.213
Average	0.664	0.655	0.109	0.138	0.130	**0.123**	0.212	0.182

6.2 Results with deep learning methods

In this section we will show the results obtained with the experiments taking into account deep learning methods such as the architectures based on CNNs and LSTMs, explained in 5.2.2. We will start with the results obtained with the YouTube transliterations and then we will show the results with the Twitter data.

6.2.1 Results with the English dataset from YouTube Personality

Personality trait estimation

In Table 6.21 and Table 6.22 we can observe the results of the regression experiment where the scores for each of the five personality traits are predicted. These results are lower for all five traits if we compare them with those obtained taking into account an SVR (see Table 6.1). From the results we can observe a general behavior: better results are obtained when considering pre-trained embeddings and freezing the embedding layer, because for the case of the CNN architecture, the five traits obtained the best result

considering this configuration (4 traits with pre-trained Word2Vec embeddings and 1 trait with pre-trained GloVe embeddings); and for the case of the LSTM architecture, three of the five traits achieved the best result with pre-trained Word2Vec embeddings and freezing the embedding layer. For the case of the CNN architecture $r, \rho \geq 0.25$ were achieved for conscientiousness, extraversion and agreeableness traits, and the lowest result is for openness to experience trait with $r = 0.18$, $\rho = 0.18$, MAE $= 0.58$ and RMSE $= 0.72$. While for the case of the LSTM architecture, four of the five traits have $r, \rho \leq 0.21$, the exception being the agreeableness trait (best result) with $r = 0.28$ and $\rho = 0.23$.

Table 6.21. Results for personality trait estimation with the English dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	No	0.10 ± 0.03	0.09 ± 0.04	0.62 ± 0.01	0.78 ± 0.01
	Word2Vec	No	0.16 ± 0.01	0.15 ± 0.02	0.59 ± 0.01	0.73 ± 0.00
		Yes	**0.18 ± 0.02**	**0.18 ± 0.02**	**0.58 ± 0.01**	**0.72 ± 0.01**
	GloVe	No	0.15 ± 0.03	0.14 ± 0.02	0.60 ± 0.01	0.75 ± 0.01
		Yes	0.15 ± 0.03	0.15 ± 0.03	0.60 ± 0.01	0.75 ± 0.01
Cons	Keras	No	0.17 ± 0.01	0.16 ± 0.02	0.64 ± 0.01	0.81 ± 0.01
	Word2Vec	No	0.23 ± 0.02	0.23 ± 0.02	0.60 ± 0.00	0.76 ± 0.01
		Yes	**0.28 ± 0.02**	**0.28 ± 0.02**	**0.58 ± 0.01**	**0.75 ± 0.01**
	GloVe	No	0.25 ± 0.03	0.25 ± 0.03	0.60 ± 0.01	0.76 ± 0.01
		Yes	0.26 ± 0.03	0.26 ± 0.03	0.60 ± 0.01	0.76 ± 0.01
Extr	Keras	No	0.20 ± 0.02	0.22 ± 0.03	0.81 ± 0.01	0.99 ± 0.01
	Word2Vec	No	0.29 ± 0.02	0.29 ± 0.02	0.77 ± 0.01	0.94 ± 0.01
		Yes	**0.32 ± 0.03**	**0.32 ± 0.03**	**0.76 ± 0.01**	**0.93 ± 0.01**
	GloVe	No	0.27 ± 0.02	0.27 ± 0.02	0.78 ± 0.01	0.95 ± 0.01
		Yes	0.26 ± 0.01	0.26 ± 0.01	0.78 ± 0.00	0.95 ± 0.01
Agr	Keras	No	0.21 ± 0.03	0.19 ± 0.04	0.71 ± 0.01	0.89 ± 0.01
	Word2Vec	No	0.25 ± 0.02	0.23 ± 0.02	0.68 ± 0.00	0.86 ± 0.00
		Yes	**0.33 ± 0.02**	**0.30 ± 0.02**	**0.66 ± 0.01**	**0.83 ± 0.01**
	GloVe	No	0.24 ± 0.04	0.22 ± 0.05	0.68 ± 0.01	0.86 ± 0.01
		Yes	0.28 ± 0.02	0.24 ± 0.02	0.68 ± 0.00	0.85 ± 0.01
Emot	Keras	No	0.13 ± 0.02	0.11 ± 0.03	0.67 ± 0.01	0.84 ± 0.01
	Word2Vec	No	0.19 ± 0.02	0.16 ± 0.03	0.62 ± 0.01	0.78 ± 0.01
		Yes	0.21 ± 0.02	0.18 ± 0.02	0.61 ± 0.00	0.77 ± 0.00
	GloVe	No	0.18 ± 0.04	0.15 ± 0.04	0.63 ± 0.01	0.80 ± 0.01
		Yes	**0.23 ± 0.02**	**0.20 ± 0.03**	**0.61 ± 0.01**	**0.77 ± 0.01**

Table 6.22. Results for personality trait estimation with the English dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	**No**	**0.21 ± 0.03**	**0.19 ± 0.04**	**0.56 ± 0.01**	**0.70 ± 0.00**
	Word2Vec	No	0.16 ± 0.03	0.16 ± 0.03	0.57 ± 0.00	0.71 ± 0.00
		Yes	0.17 ± 0.04	0.18 ± 0.04	0.57 ± 0.00	0.71 ± 0.00
	GloVe	No	0.15 ± 0.03	0.14 ± 0.03	0.57 ± 0.01	0.72 ± 0.01
		Yes	0.15 ± 0.03	0.15 ± 0.02	0.57 ± 0.00	0.72 ± 0.01
Cons	Keras	No	0.13 ± 0.03	0.11 ± 0.02	0.61 ± 0.00	0.77 ± 0.00
	Word2Vec	No	0.15 ± 0.03	0.11 ± 0.03	0.61 ± 0.00	0.77 ± 0.00
		Yes	**0.19 ± 0.03**	**0.17 ± 0.05**	**0.60 ± 0.01**	**0.76 ± 0.01**
	GloVe	No	0.09 ± 0.04	0.07 ± 0.04	0.63 ± 0.01	0.79 ± 0.01
		Yes	0.13 ± 0.04	0.10 ± 0.05	0.62 ± 0.01	0.78 ± 0.01
Extr	Keras	No	0.16 ± 0.05	0.12 ± 0.05	0.79 ± 0.01	0.97 ± 0.01
	Word2Vec	No	0.17 ± 0.02	0.14 ± 0.02	0.80 ± 0.05	0.97 ± 0.00
		Yes	**0.19 ± 0.03**	**0.18 ± 0.02**	**0.79 ± 0.00**	**0.96 ± 0.00**
	GloVe	No	0.13 ± 0.04	0.12 ± 0.04	0.81 ± 0.01	0.98 ± 0.01
		Yes	0.17 ± 0.02	0.17 ± 0.03	0.80 ± 0.01	0.97 ± 0.01
Agr	Keras	No	0.23 ± 0.03	0.20 ± 0.03	0.68 ± 0.01	0.86 ± 0.01
	Word2Vec	No	0.26 ± 0.03	0.22 ± 0.03	0.68 ± 0.01	0.86 ± 0.01
		Yes	**0.28 ± 0.04**	**0.23 ± 0.04**	**0.68 ± 0.01**	**0.85 ± 0.01**
	GloVe	No	0.25 ± 0.04	0.22 ± 0.04	0.69 ± 0.01	0.87 ± 0.01
		Yes	0.26 ± 0.03	0.22 ± 0.03	0.69 ± 0.01	0.86 ± 0.01
Emot	Keras	**No**	**0.17 ± 0.05**	**0.14 ± 0.03**	**0.60 ± 0.01**	**0.77 ± 0.01**
	Word2Vec	No	0.09 ± 0.02	0.06 ± 0.02	0.62 ± 0.00	0.78 ± 0.00
		Yes	0.12 ± 0.03	0.08 ± 0.03	0.61 ± 0.00	0.78 ± 0.00
	GloVe	No	0.10 ± 0.04	0.06 ± 0.03	0.63 ± 0.01	0.79 ± 0.01
		Yes	0.15 ± 0.02	0.11 ± 0.02	0.62 ± 0.00	0.78 ± 0.00

Weak vs. strong presence of each trait

The results of the bi-class experiment considering deep learning architectures can be found in Table 6.23 and Table 6.24. Similar to what happened in the regression experiments, the fact of using pre-trained embeddings and

freezing the embedding layer provides the best results, since for the case of the CNN architecture, the 5 traits obtained a better performance with this configuration, and for the case of the LSTM architecture, 4 of the 5 traits (except for conscientiousness) obtained the best performance making use of the mentioned configuration. For this case of classification between weak presence and strong presence of the traits, the performance was improved compared to the performance obtained using a SVM with Gaussian kernel (see Table 6.2) for the traits openness to experience (using a LSTM architecture) and emotional stability (using a CNN architecture) by 2.2% and 0.4% respectively.

Table 6.23. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	53.7 ± 1.3	49.8 ± 6.1	57.6 ± 5.1	53.5 ± 1.4	0.56 ± 0.02
	Word2Vec	No	56.3 ± 2.7	56.8 ± 5.5	55.7 ± 6.6	56.1 ± 2.8	0.60 ± 0.03
		Yes	**58.2 ± 1.0**	**58.4 ± 4.1**	**58.1 ± 4.1**	**58.2 ± 1.0**	**0.61 ± 0.01**
	GloVe	No	56.0 ± 2.5	54.5 ± 4.8	57.5 ± 6.1	55.9 ± 2.5	0.59 ± 0.03
		Yes	57.9 ± 1.8	55.0 ± 3.0	60.7 ± 4.6	57.8 ± 1.8	0.61 ± 0.02
Cons	Keras	No	59.0 ± 1.8	50.6 ± 6.4	66.8 ± 3.5	58.6 ± 2.1	0.64 ± 0.02
	Word2Vec	No	59.8 ± 1.2	55.3 ± 2.7	63.9 ± 3.1	59.7 ± 1.2	0.64 ± 0.02
		Yes	**62.2 ± 1.5**	**58.4 ± 4.0**	**65.6 ± 2.1**	**62.1 ± 1.6**	**0.67 ± 0.01**
	GloVe	No	60.3 ± 2.3	55.5 ± 3.0	64.7 ± 4.9	60.1 ± 2.2	0.65 ± 0.02
		Yes	58.7 ± 1.9	52.7 ± 5.2	64.4 ± 4.9	58.5 ± 2.0	0.63 ± 0.02
Extr	Keras	No	57.3 ± 2.0	46.7 ± 4.8	67.2 ± 5.7	56.7 ± 1.9	0.61 ± 0.02
	Word2Vec	No	61.3 ± 2.1	54.8 ± 3.8	67.6 ± 3.5	61.2 ± 2.2	0.66 ± 0.02
		Yes	**61.5 ± 1.7**	**55.6 ± 2.5**	**67.0 ± 4.3**	**61.3 ± 1.6**	**0.67 ± 0.02**
	GloVe	No	59.7 ± 1.9	52.5 ± 4.4	66.4 ± 4.6	59.4 ± 1.9	0.64 ± 0.02
		Yes	59.3 ± 2.2	54.5 ± 5.2	63.8 ± 4.5	59.2 ± 2.3	0.63 ± 0.02
Agr	Keras	No	60.3 ± 2.2	44.7 ± 5.3	73.6 ± 2.5	59.3 ± 2.5	0.65 ± 0.02
	Word2Vec	No	60.6 ± 2.3	48.3 ± 5.0	71.1 ± 3.0	60.0 ± 2.5	0.66 ± 0.02
		Yes	**62.3 ± 1.7**	**53.1 ± 3.9**	**70.2 ± 1.8**	**62.0 ± 1.9**	**0.68 ± 0.01**
	GloVe	No	59.9 ± 2.4	47.8 ± 6.5	70.2 ± 3.6	59.3 ± 2.7	0.66 ± 0.02
		Yes	61.1 ± 1.4	51.0 ± 5.1	69.7 ± 3.6	60.6 ± 1.6	0.66 ± 0.02
Emot	Keras	No	56.3 ± 1.6	56.7 ± 7.7	55.9 ± 6.2	56.1 ± 1.6	0.59 ± 0.02
	Word2Vec	No	55.7 ± 1.7	54.9 ± 6.3	56.5 ± 6.9	55.5 ± 1.8	0.59 ± 0.01
		Yes	56.8 ± 1.7	57.5 ± 6.2	56.2 ± 7.4	56.6 ± 1.7	0.60 ± 0.02
	GloVe	No	55.7 ± 2.1	55.5 ± 5.8	55.9 ± 4.4	55.9 ± 2.1	0.58 ± 0.03
		Yes	**57.2 ± 1.6**	**56.5 ± 3.8**	**57.8 ± 4.8**	**57.1 ± 1.6**	**0.60 ± 0.01**

Table 6.24. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	56.0 ±1.5	48.9 ± 5.0	63.1 ± 4.6	55.7 ± 1.6	0.60 ± 0.01
	Word2Vec	No	57.7 ± 1.6	60.2 ± 5.8	55.1 ± 5.0	57.5 ± 1.6	0.61 ± 0.01
		Yes	58.2 ± 1.2	63.3 ± 7.9	53.1 ± 7.9	57.8 ± 1.4	0.62 ± 0.01
	GloVe	No	58.6 ± 1.6	63.0 ± 5.7	54.3 ± 6.3	58.4 ± 1.6	0.62 ± 0.01
		Yes	**58.7 ± 2.0**	**60.8 ± 4.3**	**56.6 ± 4.7**	**58.6 ± 2.0**	**0.63 ± 0.02**
Cons	Keras	No	55.0 ± 1.2	33.4 ± 5.1	75.2 ± 4.0	52.8 ± 1.8	0.59 ± 0.02
	Word2Vec	No	56.0 ± 2.7	37.8 ± 5.8	73.1 ± 3.9	54.5 ± 3.0	0.61 ± 0.03
		Yes	56.6 ± 1.2	39.1 ± 4.1	72.9 ± 2.2	55.2 ± 1.6	0.61 ± 0.02
	GloVe	**No**	**57.2 ± 2.2**	**47.3 ± 4.8**	**66.5 ± 4.2**	**56.7 ± 2.3**	**0.62 ± 0.02**
		Yes	56.5 ± 1.7	45.7 ± 4.3	66.7 ± 2.1	56.0 ± 1.9	0.60 ± 0.02
Extr	Keras	No	52.6 ± 1.6	24.7 ± 3.7	78.7 ± 3.7	48.8 ± 1.9	0.54 ± 0.02
	Word2Vec	No	54.8 ± 1.4	48.1 ± 4.1	61.1 ± 3.8	54.5 ± 1.5	0.58 ± 0.01
		Yes	56.5 ± 1.9	49.1 ± 5.0	63.3 ± 5.4	56.2 ± 1.9	0.60 ± 0.01
	GloVe	No	56.3 ± 1.9	46.1 ± 3.6	65.8 ± 3.0	55.8 ± 1.9	0.59 ± 0.01
		Yes	**56.6 ± 1.4**	**52.2 ± 5.6**	**60.8 ± 3.2**	**56.5 ± 1.6**	**0.60 ± 0.01**
Agr	Keras	No	58.6 ± 1.2	34.2 ± 7.0	79.4 ± 5.0	56.1 ± 2.2	0.62 ± 0.02
	Word2Vec	No	60.1 ± 1.9	39.5 ± 6.2	77.7 ± 3.7	58.4 ± 2.4	0.64 ± 0.03
		Yes	**60.1 ± 2.4**	**42.7 ± 6.9**	**75.0 ± 4.4**	**58.8 ± 2.7**	**0.65 ± 0.02**
	GloVe	No	58.9 ± 2.0	41.5 ± 4.0	73.8 ± 1.4	57.7 ± 2.3	0.63 ± 0.03
		Yes	59.4 ± 1.5	40.4 ± 4.0	75.6 ± 4.0	58.0 ± 1.5	0.63 ± 0.02
Emot	Keras	No	54.4 ± 1.6	52.5 ± 5.6	56.2 ± 5.5	54.2 ± 1.7	0.57 ± 0.02
	Word2Vec	No	52.7 ± 2.3	46.5 ± 7.0	58.8 ± 3.8	52.4 ± 2.6	0.56 ± 0.03
		Yes	**54.9 ± 3.1**	**51.2 ± 6.9**	**58.6 ± 6.3**	**54.7 ± 3.1**	**0.59 ± 0.03**
	GloVe	No	51.4 ± 1.3	50.1 ± 4.4	52.8 ± 4.0	51.4 ± 1.3	0.53 ± 0.02
		Yes	53.6 ± 2.9	50.0 ± 4.1	57.1 ± 5.5	53.4 ± 2.9	0.55 ± 0.03

ROC curves of this bi-class classification experiment considering CNN and LSTM architectures can be found in Figure 6.5. As can be seen, the CNN architecture taking into account the pre-trained embeddings of Word2Vec and freezing the embedding layer provides the best results in terms of AUC for 4 of the 5 traits, with the openness to experience trait being the exception, for which the best result in terms of AUC was obtained with the LSTM

architecture and taking into account the pre-trained embeddings of GloVe and freezing the embedding layer as well.

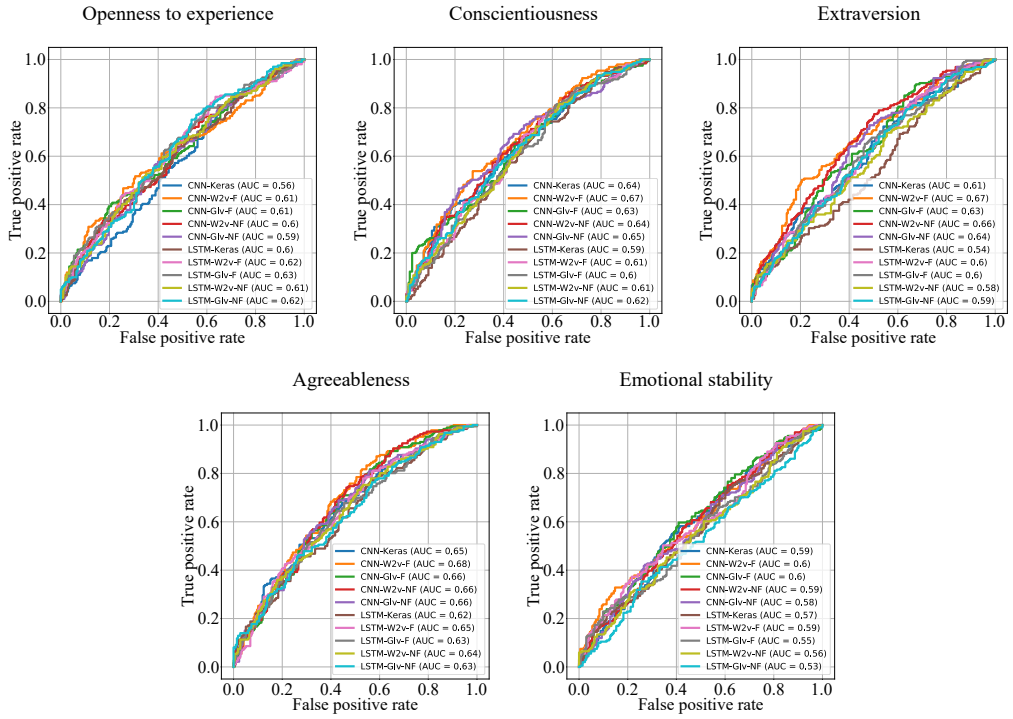

Figure 6.5. ROC curves obtained for English dataset from YouTube Personality considering CNN and LSTM architectures. Keras: Keras embedding layer trained from scratch, W2v: pre-trained Word2Vec embeddings, Glv: pre-trained GloVe embeddings, F: Freezing the embedding layer, NF: No freezing the embedding layer.

Classification of personality traits into 3 levels

The results of the LP vs MP vs HP tri-class classification experiment considering CNN and LSTM architectures can be found in Table 6.25 and Table 6.27. Similar to the case of regression and bi-class classification experiments, the best results were obtained when the embedding layer is frozen, and in general, when the pre-trained Word2Vec embeddings are taken into account, since for the CNN architecture for 4 of the 5 traits the best results were achieved with this configuration and for the LSTM architecture this happens with 3 of the 5 traits. Comparing these results with those obtained with the SVM (see Table 6.3), we can observe that in general, the SVM

obtained better results, with the only difference being that with the CNN architecture we were able to improve the performance for the openness to experience trait by 0.6% in terms of the F1-score metric. Similarly, in general terms, better results were obtained with the CNN architecture than with the LSTM architecture, since only for the emotional stability trait the LSTM architecture provided the best performance, for the other 4 features CNN won. The confusion matrices are shown in Table 6.26 and Table 6.28. We can observe similar behavior to the triclass classification case with a SVM with Gaussian kernel (see Table 6.4), where the classifier tends to confuse the samples among the 3 classes. We also observe that the CNN architecture gives better results in terms of confusion matrix compared to the LSTM architecture for the first 3 personality traits of the OCEAN model, where it achieves to classify the target class with slightly higher accuracy.

Table 6.25. Tri-class classification results with the English dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	39.1 ± 2.1	38.2 ± 2.1	38.2 ± 2.1	0.07 ± 0.03
	Word2Vec	No	41.0 ± 2.0	40.8 ± 2.2	40.6 ± 2.1	0.11 ± 0.03
		Yes	**41.8 ± 1.6**	**41.6 ± 1.6**	**41.3 ± 1.6**	**0.12 ± 0.02**
	GloVe	No	40.2 ± 2.1	39.8 ± 2.2	39.6 ± 2.2	0.09 ± 0.03
		Yes	41.7 ± 2.0	41.3 ± 2.1	41.2 ± 2.0	0.12 ± 0.03
Cons	Keras	No	40.7 ± 1.7	39.9 ± 1.5	40.1 ± 1.6	0.10 ± 0.02
	Word2Vec	No	42.1 ± 2.7	41.6 ± 2.8	41.7 ± 2.7	0.13 ± 0.04
		Yes	**44.9 ± 1.8**	**44.5 ± 2.0**	**44.6 ± 1.8**	**0.17 ± 0.03**
	GloVe	No	42.9 ± 1.2	42.4 ± 1.2	42.6 ± 1.1	0.14 ± 0.02
		Yes	42.6 ± 1.7	42.1 ± 1.6	42.2 ± 1.7	0.13 ± 0.03
Extr	Keras	No	38.3 ± 1.5	37.9 ± 1.5	37.9 ± 1.5	0.07 ± 0.02
	Word2Vec	No	41.8 ± 2.2	41.6 ± 2.2	41.6 ± 2.2	0.12 ± 0.03
		Yes	**42.3 ± 1.3**	**42.1 ± 1.2**	**42.1 ± 1.3**	**0.13 ± 0.02**
	GloVe	No	41.8 ± 2.2	41.7 ± 2.3	41.5 ± 2.3	0.12 ± 0.03
		Yes	40.9 ± 3.0	40.6 ± 3.0	40.7 ± 3.0	0.11 ± 0.05
Agr	Keras	No	42.1 ± 2.1	42.1 ± 2.0	42.0 ± 2.0	0.13 ± 0.03
	Word2Vec	No	44.6 ± 2.2	44.4 ± 2.2	44.4 ± 2.2	0.17 ± 0.03
		Yes	**45.2 ± 1.3**	**45.0 ± 1.3**	**45.0 ± 1.3**	**0.18 ± 0.02**
	GloVe	No	42.6 ± 1.5	42.4 ± 1.6	42.5 ± 1.5	0.14 ± 0.02
		Yes	42.8 ± 2.5	42.8 ± 2.6	42.7 ± 2.6	0.14 ± 0.04
Emot	Keras	No	35.0 ± 1.6	34.9 ± 1.7	35.0 ± 1.7	0.02 ± 0.02
	Word2Vec	No	37.4 ± 1.2	37.2 ± 1.1	37.3 ± 1.2	0.06 ± 0.02
		Yes	37.6 ± 1.5	37.5 ± 1.5	37.6 ± 1.5	0.06 ± 0.02
	GloVe	**No**	**38.0 ± 2.7**	**37.9 ± 2.6**	**37.9 ± 2.7**	**0.07 ± 0.04**
		Yes	37.9 ± 1.8	37.7 ± 1.9	37.8 ± 1.8	0.07 ± 0.03

Table 6.26. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from YouTube Personality considering a CNN architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	36	44	20	57	25	18	47	36	17	47	31	22	40	34	26
	MP	29	53	18	43	27	30	42	37	21	28	42	30	34	34	32
	HP	33	42	25	33	31	37	36	36	28	22	40	38	30	39	31
W2v-F	LP	**42**	40	18	**55**	27	18	**48**	33	19	**56**	26	18	43	33	24
	MP	30	**49**	21	36	**33**	31	38	**37**	25	26	**43**	31	31	37	32
	HP	26	40	**34**	25	28	**47**	27	32	**41**	23	40	**37**	30	37	33
Glv-F	LP	46	37	17	53	30	17	48	31	21	51	29	20	43	31	26
	MP	31	49	20	45	29	26	38	36	26	26	41	33	34	36	30
	HP	30	42	28	29	26	45	33	29	38	23	41	36	31	35	34
W2v-NF	LP	38	41	21	53	29	18	51	31	18	55	25	20	44	32	24
	MP	28	49	23	41	29	30	40	36	24	28	41	31	34	35	31
	HP	29	36	35	29	29	42	32	30	38	25	37	38	31	35	34
Glv-NF	LP	42	38	20	53	27	20	48	33	19	51	27	22	**42**	34	24
	MP	30	48	22	42	29	29	37	39	24	29	40	31	32	**37**	31
	HP	30	41	29	29	26	45	28	35	37	24	39	37	30	35	**35**

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

Table 6.27. Tri-class classification results with the English dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	37.6 ± 1.5	32.4 ± 1.9	35.6 ± 1.5	0.03 ± 0.02
	Word2Vec	No	38.9 ± 1.3	35.6 ± 1.5	37.2 ± 1.2	0.06 ± 0.02
		Yes	39.6 ± 1.7	36.8 ± 2.4	38.0 ± 1.9	0.07 ± 0.03
	GloVe	**No**	**39.6 ± 1.6**	**38.2 ± 1.5**	**38.5 ± 1.5**	**0.08 ± 0.02**
		Yes	38.1 ± 2.4	36.8 ± 2.4	37.0 ± 2.4	0.06 ± 0.04
Cons	Keras	No	39.5 ± 1.9	35.0 ± 3.9	38.0 ± 2.2	0.07 ± 0.03
	Word2Vec	No	39.7 ± 2.6	37.1 ± 3.2	38.5 ± 2.7	0.08 ± 0.04
		Yes	**42.8 ± 1.7**	**41.7 ± 1.9**	**42.2 ± 1.8**	**0.13 ± 0.03**
	GloVe	No	40.4 ± 2.4	39.0 ± 2.8	39.6 ± 2.5	0.10 ± 0.04
		Yes	40.3 ± 1.8	39.2 ± 1.7	39.6 ± 1.8	0.10 ± 0.03
Extr	Keras	No	37.0 ± 1.8	33.4 ± 2.3	35.7 ± 1.9	0.04 ± 0.03
	Word2Vec	No	38.5 ± 2.7	37.5 ±2.9	37.8 ± 2.8	0.06 ± 0.04
		Yes	39.2 ± 2.0	38.3 ± 2.4	38.6 ± 2.1	0.08 ± 0.03
	GloVe	No	39.2 ± 2.2	38.7 ± 2.1	38.7 ± 2.1	0.08 ± 0.03
		Yes	**40.0 ± 2.4**	**39.7 ± 2.5**	**39.6 ± 2.4**	**0.10 ± 0.04**
Agr	Keras	No	38.5 ± 1.4	37.7 ± 1.7	38.3 ± 1.4	0.07 ± 0.02
	Word2Vec	No	40.1 ± 1.4	39.5 ± 1.7	39.9 ± 1.4	0.09 ± 0.02
		Yes	**43.1 ± 2.2**	**42.6 ± 2.3**	**43.0 ± 2.2**	**0.14 ± 0.03**
	GloVe	No	40.0 ± 1.9	39.3 ± 2.1	39.7 ± 2.0	0.10 ± 0.03
		Yes	42.1 ± 2.0	41.5 ± 2.1	41.9 ± 2.1	0.13 ± 0.03
Emot	Keras	No	36.6 ± 1.2	35.2 ± 1.0	36.4 ± 1.1	0.05 ± 0.02
	Word2Vec	No	38.5 ± 2.5	37.7 ± 2.6	38.3 ± 2.5	0.08 ± 0.04
		Yes	**39.9 ± 1.2**	**39.2 ± 1.4**	**39.7 ± 1.2**	**0.10 ± 0.02**
	GloVe	No	36.6 ± 1.7	36.3 ± 1.9	36.5 ± 1.8	0.05 ± 0.03
		Yes	36.9 ± 1.9	36.6 ± 2.1	36.8 ± 2.0	0.05 ± 0.03

Table 6.28. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from YouTube Personality considering a LSTM architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	33	63	4	73	15	12	60	33	7	42	42	16	37	40	23
	MP	26	68	6	63	22	15	55	38	7	33	47	20	31	52	17
	HP	21	73	6	60	21	19	62	28	10	26	48	26	34	46	20
W2v-F	LP	37	54	9	**59**	23	18	55	28	17	**57**	28	15	**50**	28	22
	MP	28	62	10	49	**27**	24	55	31	14	36	**34**	30	36	**41**	23
	HP	25	60	15	40	20	**40**	46	24	30	26	36	**38**	39	34	**27**
Glv-F	LP	41	46	13	58	23	19	**47**	36	17	54	31	15	39	33	28
	MP	32	51	17	46	27	27	43	**42**	15	37	41	22	30	43	26
	HP	30	51	19	41	26	33	40	30	**30**	33	37	30	37	36	27
W2v-NF	LP	36	56	7	66	20	14	54	31	15	50	33	17	47	32	21
	MP	26	63	11	59	23	18	52	34	14	37	39	24	36	44	20
	HP	22	65	13	50	24	26	50	24	26	33	37	30	39	37	24
Glv-NF	LP	**43**	43	14	60	20	20	46	36	18	52	33	15	36	35	29
	MP	33	**52**	15	48	27	25	42	42	16	36	41	23	28	44	28
	HP	30	49	**21**	44	24	32	43	29	28	33	40	27	34	37	29

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

6.2.2 Results with the Spanish dataset from YouTube Personality

Personality trait estimation

The results of personality trait estimation (regression experiments) for the YouTube transliterations in Spanish language taking into account deep learning methods such as the CNN and LSTM architectures can be found in Table 6.29 and Table 6.30. As can be seen, the best results were obtained when considering the pre-trained word embeddings of Word2Vec and GloVe and freezing the embedding layer, since for the CNN architecture the best result

was achieved in all the five traits considering this configuration and for the LSTM architecture this was achieved in four of the five traits. Similarly, comparing the regression performance of the CNN architecture with the LSTM architecture, we can observe that the performance of the CNN architecture is superior to the LSTM architecture for conscientiousness, extraversion and agreeableness traits, while for openness to experience and emotional stability traits, the LSTM architecture performs better. Now, compared to the SVR results with the Spanish transliterations (see Table 6.7), the performance of the openness to experience trait was improved with the LSTM architecture as follows: r, ρ by 0.1 and MAE, RMSE by 0.01. The best result between the two architectures for this regression experiment was given for conscientiousness trait with $r = 0.25$, $\rho = 0.27$, MAE $= 0.61$ and RMSE $= 0.78$.

Table 6.29. Results for personality trait estimation with the Spanish dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	No	0.08 ± 0.03	0.07 ± 0.04	0.69 ± 0.01	0.87 ± 0.02
	Word2Vec	No	0.07 ± 0.04	0.06 ± 0.05	0.62 ± 0.01	0.77 ± 0.01
		Yes	0.08 ± 0.03	0.07 ± 0.03	0.62 ± 0.01	0.77 ± 0.01
	GloVe	No	0.08 ± 0.02	0.09 ± 0.02	0.62 ± 0.01	0.78 ± 0.01
		Yes	**0.10 ± 0.02**	**0.10 ± 0.02**	**0.62 ± 0.01**	**0.78 ± 0.01**
Cons	Keras	No	0.12 ± 0.04	0.11 ± 0.04	0.72 ± 0.02	0.89 ± 0.02
	Word2Vec	No	0.21 ± 0.02	0.20 ± 0.03	0.62 ± 0.01	0.79 ± 0.01
		Yes	0.22 ± 0.03	0.21 ± 0.04	0.61 ± 0.01	0.78 ± 0.01
	GloVe	No	0.24 ± 0.02	0.25 ± 0.02	0.61 ± 0.01	0.78 ± 0.01
		Yes	**0.25 ± 0.03**	**0.27 ± 0.04**	**0.61 ± 0.01**	**0.78 ± 0.01**
Extr	Keras	No	0.15 ± 0.03	0.17 ± 0.03	0.86 ± 0.01	1.06 ± 0.01
	Word2Vec	No	0.20 ± 0.02	0.20 ± 0.02	0.82 ± 0.01	0.99 ± 0.01
		Yes	**0.23 ± 0.02**	**0.22 ± 0.03**	**0.80 ± 0.01**	**0.97 ± 0.01**
	GloVe	No	0.18 ± 0.03	0.19 ± 0.03	0.82 ± 0.01	0.99 ± 0.01
		Yes	0.19 ± 0.02	0.20 ± 0.02	0.81 ± 0.01	0.99 ± 0.01
Agr	Keras	No	0.16 ± 0.02	0.15 ± 0.02	0.78 ± 0.01	0.97 ± 0.01
	Word2Vec	No	0.17 ± 0.05	0.15 ± 0.05	0.73 ± 0.02	0.90 ± 0.02
		Yes	**0.24 ± 0.04**	**0.22 ± 0.04**	**0.71 ± 0.01**	**0.88 ± 0.01**
	GloVe	No	0.14 ± 0.03	0.14 ± 0.03	0.74 ± 0.01	0.92 ± 0.01
		Yes	0.18 ± 0.03	0.18 ± 0.03	0.73 ± 0.01	0.91 ± 0.01
Emot	Keras	No	0.12 ± 0.03	0.11 ± 0.04	0.73 ± 0.01	0.91 ± 0.01
	Word2Vec	No	0.15 ± 0.04	0.14 ± 0.04	0.64 ± 0.01	0.81 ± 0.01
		Yes	**0.17 ± 0.03**	**0.15 ± 0.03**	**0.64 ± 0.01**	**0.81 ± 0.01**
	GloVe	No	0.15 ± 0.03	0.14 ± 0.03	0.67 ± 0.01	0.83 ± 0.01
		Yes	0.16 ± 0.02	0.15 ± 0.03	0.66 ± 0.01	0.82 ± 0.01

Table 6.30. Results for personality trait estimation with the Spanish dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	No	0.05 ± 0.05	0.05 ± 0.05	0.58 ± 0.00	0.72 ± 0.01
	Word2Vec	No	0.09 ± 0.04	0.10 ± 0.04	0.58 ± 0.00	0.72 ± 0.00
		Yes	0.18 ± 0.02	0.16 ± 0.03	0.57 ± 0.00	0.71 ± 0.00
	GloVe	No	0.13 ± 0.04	0.12 ± 0.04	0.57 ± 0.00	0.71 ± 0.00
		Yes	**0.23 ± 0.02**	**0.22 ± 0.02**	**0.56 ± 0.00**	**0.70 ± 0.00**
Cons	Keras	No	0.14 ± 0.04	0.12 ± 0.04	0.60 ± 0.00	0.76 ± 0.01
	Word2Vec	No	0.14 ± 0.02	0.11 ± 0.02	0.61 ± 0.01	0.77 ± 0.01
		Yes	0.15 ± 0.04	0.14 ± 0.04	0.60 ± 0.01	0.76 ± 0.01
	GloVe	No	0.12 ± 0.05	0.11 ± 0.06	0.60 ± 0.01	0.77 ± 0.00
		Yes	**0.16 ± 0.05**	**0.15 ± 0.04**	**0.60 ± 0.01**	**0.76 ± 0.01**
Extr	Keras	No	0.13 ± 0.04	0.05 ± 0.05	0.81 ± 0.01	0.97 ± 0.01
	Word2Vec	No	0.14 ± 0.06	0.11 ± 0.05	0.81 ± 0.01	0.97 ± 0.01
		Yes	**0.17 ± 0.03**	**0.15 ± 0.04**	**0.80 ± 0.01**	**0.97 ± 0.01**
	GloVe	No	0.13 ± 0.03	0.07 ± 0.04	0.80 ± 0.00	0.97 ± 0.01
		Yes	0.14 ± 0.03	0.11 ± 0.04	0.80 ± 0.01	0.97 ± 0.01
Agr	Keras	No	0.22 ± 0.03	0.19 ± 0.02	0.68 ± 0.01	0.86 ± 0.01
	Word2Vec	**No**	**0.22 ± 0.02**	**0.21 ± 0.02**	**0.68 ± 0.00**	**0.86 ± 0.00**
		Yes	0.21 ± 0.03	0.19 ± 0.03	0.68 ± 0.01	0.86 ± 0.01
	GloVe	No	0.15 ± 0.04	0.15 ± 0.03	0.68 ± 0.01	0.87 ± 0.00
		Yes	0.14 ± 0.03	0.13 ± 0.04	0.68 ± 0.01	0.87 ± 0.00
Emot	Keras	No	0.21 ± 0.02	0.17 ± 0.02	0.60 ± 0.00	0.76 ± 0.00
	Word2Vec	No	0.18 ± 0.02	0.15 ± 0.03	0.60 ± 0.00	0.77 ± 0.00
		Yes	**0.23 ± 0.02**	**0.18 ± 0.02**	**0.60 ± 0.01**	**0.76 ± 0.00**
	GloVe	No	0.17 ± 0.05	0.14 ± 0.05	0.60 ± 0.01	0.77 ± 0.01
		Yes	0.15 ± 0.02	0.14 ± 0.02	0.60 ± 0.00	0.77 ± 0.00

Weak vs. strong presence of each trait

The results of the binary classification between weak presence and strong presence of personality traits considering CNNs and LSTMs are found in Table 6.31 and Table 6.32 respectively. Similar to the regression case, it is

appropriate to use the pre-trained word embeddings from either Word2Vec or GloVe and freezing the embedding layer, as most of the best results were obtained when the neural network is set up in this way: in 3 out of 5 traits for the CNN architecture and in 4 out of 5 traits for the LSTM architecture. Better results were achieved with the CNN architecture, since only for the openness to experience trait the best performance was achieved with the LSTM architecture. The best performance for this experiment (between LSTMs and CNNs) was achieved for the agreeableness trait with accuracy = 60%, AUC = 0.62. Comparing these results with those obtained by the SVM (see Table 6.8), only for openness to experience trait the performance was improved by 5.7% in terms of accuracy.

Table 6.31. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	51.1 ± 1.8	45.1 ± 8.1	57.1 ± 8.9	50.6 ± 1.8	0.52 ± 0.02
	Word2Vec	No	53.2 ± 2.5	50.4 ±3.9	56.0 ± 3.3	53.1 ± 2.5	0.55 ± 0.03
		Yes	**54.0 ± 2.2**	**51.5 ± 4.9**	**56.5 ± 3.6**	**53.9 ± 2.2**	**0.55 ± 0.02**
	GloVe	No	52.6 ± 2.0	50.5 ± 4.4	54.7 ± 3.5	52.5 ± 2.0	0.55 ± 0.01
		Yes	53.4 ± 2.0	49.5 ± 4.3	57.2 ± 2.9	53.2 ± 2.1	0.55 ± 0.02
Cons	Keras	No	57.2 ± 1.5	44.8 ± 3.6	68.8 ± 3.9	56.5 ± 1.5	0.61 ± 0.02
	Word2Vec	No	58.5 ± 1.5	54.0 ± 3.3	62.6 ± 2.2	58.4 ± 1.6	0.62 ± 0.02
		Yes	58.5 ± 2.1	52.8 ± 4.0	63.8 ± 1.9	58.4 ± 2.2	0.62 ± 0.01
	GloVe	No	58.2 ± 2.0	54.4 ± 4.3	61.7 ± 2.0	58.1 ± 2.1	0.62 ± 0.02
		Yes	**59.4 ± 2.0**	**56.1 ± 2.8**	**62.6 ± 3.3**	**59.4 ± 2.0**	**0.64 ± 0.02**
Extr	Keras	No	56.8 ± 1.5	42.9 ± 4.2	69.7 ± 2.6	55.9 ± 1.7	0.60 ± 0.02
	Word2Vec	No	57.8 ± 1.4	52.5 ± 3.7	62.8 ± 3.5	57.7 ± 1.4	0.62 ± 0.02
		Yes	**58.2 ± 1.3**	**50.8 ± 2.8**	**65.1 ± 2.3**	**58.0 ± 1.4**	**0.62 ± 0.02**
	GloVe	No	57.2 ± 2.4	49.4 ± 4.7	64.5 ± 4.3	56.9 ± 2.4	0.61 ± 0.02
		Yes	56.1 ± 1.9	45.6 ± 2.9	65.8 ± 2.9	55.6 ± 1.9	0.59 ± 0.02
Agr	Keras	No	58.9 ± 1.0	35.5 ± 3.5	78.8 ± 3.1	56.7 ± 1.2	0.63 ± 0.02
	Word2Vec	No	59.1 ± 1.9	42.4 ± 3.4	73.4 ± 3.1	58.1 ± 1.9	0.63 ± 0.02
		Yes	59.1 ± 2.4	41.2 ± 2.9	74.3 ± 3.4	57.8 ± 2.4	0.64 ± 0.03
	GloVe	**No**	**60.0 ± 1.4**	**40.0 ± 3.7**	**77.2 ± 1.2**	**58.5 ± 1.8**	**0.62 ± 0.02**
		Yes	58.6 ± 1.9	35.6 ± 4.5	78.1 ± 3.3	56.5 ± 2.2	0.63 ± 0.02
Emot	Keras	No	56.7 ± 1.1	56.6 ± 6.5	56.7 ± 7.0	56.5 ± 1.3	0.60 ± 0.01
	Word2Vec	No	54.6 ± 2.3	54.9 ± 4.4	54.2 ± 4.5	54.5 ± 2.3	0.58 ± 0.03
		Yes	55.7 ± 1.9	56.3 ± 3.3	55.1 ± 2.3	55.7 ± 1.9	0.59 ± 0.02
	GloVe	**No**	**57.1 ± 1.1**	**55.5 ± 3.2**	**58.8 ± 2.9**	**57.1 ± 1.1**	**0.61 ± 0.01**
		Yes	55.5 ± 2.5	54.2 ± 3.7	56.8 ± 5.9	55.4 ± 2.4	0.59 ± 0.02

Table 6.32. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	54.0 ± 1.7	41.7 ± 4.3	66.1 ± 3.9	53.2 ± 1.9	0.58 ± 0.01
	Word2Vec	No	56.5 ± 1.4	53.0 ± 5.7	59.9 ± 5.9	56.3 ± 1.4	0.59 ± 0.01
		Yes	57.7 ± 2.3	55.9 ± 7.1	59.5 ± 5.5	57.5 ± 2.4	0.61 ± 0.02
	GloVe	No	56.7 ± 1.2	48.7 ± 7.1	64.5 ± 5.9	56.2 ± 1.6	0.60 ± 0.01
		Yes	**58.0 ± 1.8**	**56.0 ± 6.7**	**60.0 ± 7.9**	**57.8 ± 1.7**	**0.62 ± 0.02**
Cons	Keras	No	53.4 ± 1.7	34.5 ± 7.4	71.1 ± 5.8	51.5 ± 2.7	0.56 ± 0.02
	Word2Vec	No	51.1 ± 2.0	35.7 ± 9.4	65.6 ± 7.3	49.6 ± 2.8	0.52 ± 0.03
		Yes	52.4 ± 2.1	37.6 ± 9.0	66.2 ± 8.6	51.0 ± 2.2	0.54 ± 0.02
	GloVe	No	54.6 ± 1.8	42.1 ± 7.3	66.2 ± 5.0	53.7 ± 2.3	0.57 ± 0.02
		Yes	**55.0 ± 1.5**	**41.1 ± 3.5**	**67.8 ± 3.0**	**54.1 ± 1.6**	**0.58 ± 0.02**
Extr	Keras	No	55.3 ± 1.0	35.1 ± 5.0	74.2 ± 4.2	53.4 ± 1.5	0.58 ± 0.01
	Word2Vec	No	57.5 ± 1.2	47.4 ± 8.2	67.0 ± 8.0	56.8 ± 1.3	0.61 ± 0.02
		Yes	**58.1 ± 1.1**	**46.8 ± 4.4**	**68.6 ± 3.4**	**57.5 ± 1.3**	**0.62 ± 0.01**
	GloVe	No	55.2 ± 1.8	39.9 ± 10.1	69.5 ± 8.8	53.7 ± 2.7	0.58 ± 0.03
		Yes	57.4 ± 1.4	48.3 ± 8.4	66.0 ± 6.5	56.8 ± 1.9	0.60 ± 0.01
Agr	Keras	No	56.2 ± 1.4	27.9 ± 8.8	80.4 ± 6.3	52.5 ± 3.1	0.57 ± 0.02
	Word2Vec	No	54.7 ± 1.7	29.4 ± 6.9	76.3 ± 4.9	51.8 ± 2.7	0.55 ± 0.03
		Yes	**56.6 ± 2.6**	**34.2 ± 6.6**	**75.7 ± 4.5**	**54.5 ± 3.1**	**0.58 ± 0.03**
	GloVe	No	54.5 ± 1.0	24.9 ± 6.1	79.7 ± 4.7	50.5 ± 2.4	0.54 ± 0.03
		Yes	55.9 ± 2.1	24.9 ± 4.5	82.3 ± 4.5	51.7 ± 2.4	0.56 ± 0.02
Emot	Keras	**No**	**54.1 ± 2.1**	**53.2 ± 4.5**	**54.9 ± 3.2**	**54.0 ± 2.1**	**0.57 ± 0.02**
	Word2Vec	No	51.7 ± 1.9	53.2 ± 9.4	50.2 ± 9.8	51.3 ± 2.0	0.54 ± 0.02
		Yes	51.0 ± 1.3	47.5 ± 12.1	54.4 ± 13.6	50.1 ± 1.2	0.51 ± 0.02
	GloVe	No	51.9 ± 1.4	43.3 ± 10.2	60.4 ± 11.8	51.0 ± 1.3	0.53 ± 0.03
		Yes	52.4 ± 1.6	45.2 ± 10.4	59.5 ± 8.7	51.7 ± 2.1	0.54 ± 0.02

The ROC curves of this bi-class experiment can be found in Figure 6.6, where the behavior of CNN and LSTM architectures can be observed. In general, the performance is better when CNN-based architectures are taken into account, since for 4 of the 5 traits (except for openness to experience) better results were obtained in terms of AUC values. Now, regarding the type of embedding used, as previously mentioned, better results are obtained when

considering pre-trained word embeddings, and in this case, specifically those based on GloVe, since with these word embeddings better AUC values were obtained (see green and gray curves in the graphs for the different traits).

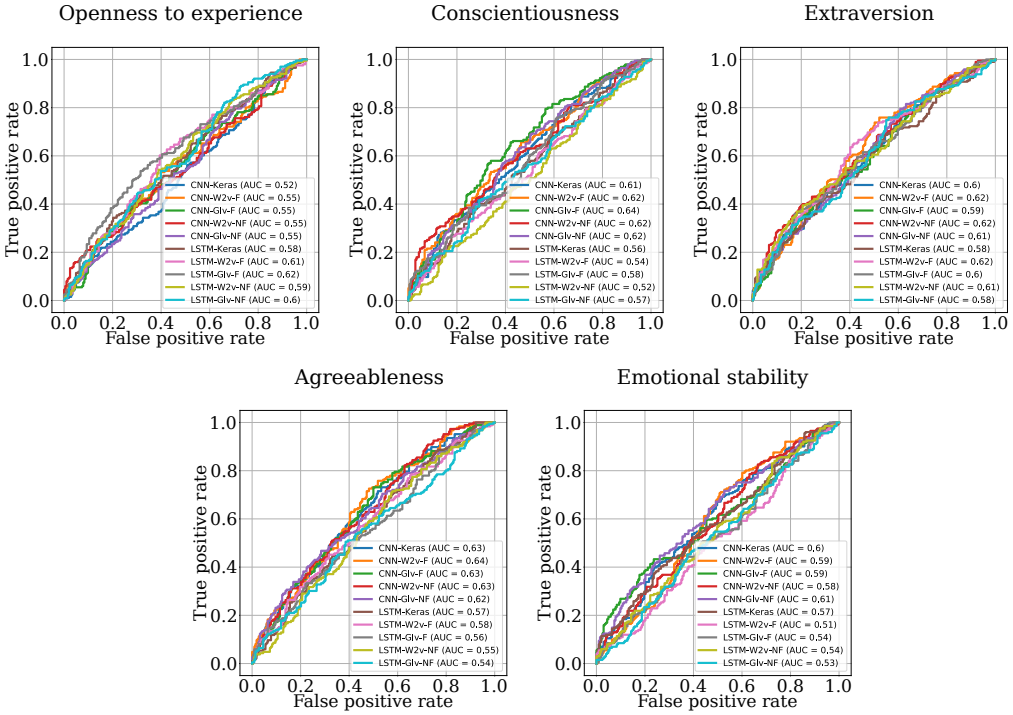

Figure 6.6. ROC curves obtained for Spanish dataset from YouTube Personality considering CNN and LSTM architectures. Keras: Keras embedding layer trained from scratch, W2v: pre-trained Word2Vec embeddings, Glv: pre-trained GloVe embeddings, F: Freezing the embedding layer, NF: No freezing the embedding layer.

Classification of personality traits into 3 levels

The results of the tri-class classification experiments taking into account Spanish transliterations and architectures based on CNNs and LSTMs can be found in Table 6.33 and Table 6.35. As we can observe from the results, for both architectures better results are achieved when pre-trained embeddings are considered, but for the case of the architecture based on CNNs, the embedding layer should not be frozen; while for the architecture with LSTMs it is better if the embedding layer is frozen in order to obtain better results. Continuing in these line, if we compare the results obtained between CNNs

and LSTMs, the CNN wins as it provided the best results in three of five traits: conscientiousness, extraversion and agreeableness. Now, if we compare these results with those obtained with the classical method (see Table 6.9), we realize that only the performance in terms of F1-score was improved for the openness to experience trait (by 0.6% with the CNN architecture and by 2.6% with the LSTM architecture) and also for the emotional stability trait by 0.9% considering a neural network with LSTMs. The confusion matrices can be found in Table 6.34 and Table 6.36. In this case, the algorithms with CNNs and the LSTMs were able to better discriminate the subjects belonging to the 'low presence' class than the subjects of the 'high presence' class, since in most cases and features, the percentage of classification of the LP class is higher than the percentage of classification of the HP class.

Table 6.33. Tri-class classification results with the Spanish dataset from YouTube Personality considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	37.3 ± 1.7	31.9 ± 2.8	35.2 ± 1.8	0.03 ± 0.03
	Word2Vec	**No**	**38.5 ± 2.0**	**36.6 ± 2.0**	**37.3 ± 1.9**	**0.06 ± 0.03**
		Yes	38.0 ± 1.1	35.3 ± 1.7	36.5 ± 1.1	0.05 ± 0.02
	GloVe	No	38.4 ± 1.0	36.3 ± 1.8	37.2 ± 1.1	0.06 ± 0.02
		Yes	38.4 ± 0.9	35.2 ± 1.6	36.7 ± 1.0	0.05 ± 0.01
Cons	Keras	No	40.0 ± 2.3	37.8 ± 2.8	38.9 ± 2.4	0.09 ± 0.04
	Word2Vec	No	41.1 ± 2.5	40.5 ± 2.6	40.6 ± 2.5	0.11 ± 0.04
		Yes	41.2 ± 2.4	40.2 ± 2.5	40.8 ± 2.3	0.11 ± 0.04
	GloVe	No	41.4 ± 2.3	40.2 ± 2.3	40.7 ± 2.3	0.11 ± 0.03
		Yes	**42.2 ± 1.7**	**41.2 ± 1.6**	**41.6 ± 1.7**	**0.13 ± 0.03**
Extr	Keras	No	37.8 ± 0.9	35.6 ± 1.38	36.8 ± 0.9	0.05 ± 0.01
	Word2Vec	**No**	**40.5 ± 1.7**	**39.9 ± 1.7**	**40.1 ± 1.6**	**0.10 ± 0.02**
		Yes	40.3 ± 1.9	39.8 ± 1.9	39.8 ± 1.8	0.10 ± 0.02
	GloVe	No	38.1 ± 1.5	36.7 ± 2.1	37.4 ± 1.6	0.06 ± 0.02
		Yes	38.0 ± 1.6	37.3 ± 1.7	37.5 ± 1.6	0.06 ± 0.02
Agr	Keras	No	42.4 ± 1.3	41.4 ± 1.4	42.6 ± 1.3	0.14 ± 0.01
	Word2Vec	No	41.5 ± 1.2	41.3 ± 1.1	41.3 ± 1.1	0.12 ± 0.02
		Yes	41.0 ± 2.1	40.5 ± 2.2	40.8 ± 2.2	0.11 ± 0.03
	GloVe	**No**	**42.8 ± 1.4**	**42.0 ± 1.8**	**42.6 ± 1.4**	**0.14 ± 0.02**
		Yes	41.9 ± 1.6	41.1 ± 1.6	41.7 ± 1.6	0.13 ± 0.02
Emot	Keras	No	37.2 ± 1.8	36.7 ± 2.0	37.1 ± 1.7	0.06 ± 0.03
	Word2Vec	**No**	**37.3 ± 2.0**	**36.9 ± 2.2**	**37.2 ± 2.0**	**0.06 ± 0.03**
		Yes	36.7 ± 1.9	36.5 ± 1.9	36.6 ± 1.9	0.05 ± 0.03
	GloVe	No	36.6 ± 2.2	36.1 ± 2.3	36.6 ± 2.2	0.05 ± 0.03
		Yes	37.0 ± 2.0	36.4 ± 2.1	36.9 ± 2.0	0.05 ± 0.03

Table 6.34. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from YouTube Personality considering a CNN architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	24	70	6	65	21	14	60	30	10	50	32	18	40	34	26
	MP	23	72	5	55	24	21	56	31	13	27	48	25	28	41	31
	HP	26	64	9	47	26	27	52	28	20	24	46	30	29	41	30
W2v-F	LP	29	58	13	53	26	21	51	32	17	47	34	19	42	32	26
	MP	24	63	13	44	26	30	42	37	21	30	46	24	29	37	34
	HP	24	59	17	32	25	43	39	29	32	30	40	30	31	38	31
Glv-F	LP	29	60	11	**57**	23	20	50	33	17	53	28	19	39	36	25
	MP	23	66	11	42	**29**	29	45	35	20	36	43	21	31	40	29
	HP	25	59	16	34	27	**39**	41	32	27	35	36	29	31	37	32
W2v-NF	LP	**36**	49	15	54	28	18	**50**	31	19	46	33	21	**41**	35	24
	MP	29	**57**	14	45	31	24	41	**39**	20	28	43	29	28	**41**	31
	HP	30	50	**20**	33	30	37	39	30	**31**	24	42	34	28	42	**30**
Glv-NF	LP	36	51	13	59	23	18	49	37	14	**56**	25	19	42	32	26
	MP	32	57	11	43	27	30	45	41	14	35	**43**	22	34	34	32
	HP	33	49	18	37	26	37	41	36	23	34	36	**30**	32	34	34

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

Table 6.35. Tri-class classification results with the Spanish dataset from YouTube Personality considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	37.1 ± 1.4	30.8 ± 1.2	34.9 ± 1.2	0.02 ± 0.02
	Word2Vec	No	37.3 ± 1.5	33.4 ± 2.5	35.5 ± 1.6	0.03 ± 0.03
		Yes	39.1 ± 1.9	35.6 ± 2.4	37.4 ± 1.9	0.06 ± 0.03
	GloVe	No	38.3 ± 2.1	33.8 ± 2.4	36.5 ± 2.1	0.05 ± 0.03
		Yes	**40.0 ± 1.5**	**37.5 ± 2.3**	**38.6 ± 1.7**	**0.08 ± 0.03**
Cons	Keras	No	38.4 ± 1.9	33.7 ± 3.4	36.8 ± 2.0	0.05 ± 0.03
	Word2Vec	No	37.9 ± 1.8	36.0 ± 2.2	36.9 ± 1.9	0.06 ± 0.03
		Yes	38.5 ± 1.5	36.3 ± 1.9	37.4 ± 1.6	0.06 ± 0.02
	GloVe	No	39.7 ± 2.4	37.6 ± 2.8	38.6 ± 2.4	0.08 ± 0.04
		Yes	**40.5 ± 1.7**	**38.4 ± 2.3**	**39.4 ± 1.8**	**0.09 ± 0.03**
Extr	Keras	No	38.2 ± 2.3	35.2 ± 3.8	37.1 ± 2.4	0.06 ± 0.04
	Word2Vec	**No**	**38.7 ± 1.3**	**37.8 ± 1.3**	**38.1 ± 1.3**	**0.07 ± 0.02**
		Yes	37.9 ± 1.8	37.5 ± 1.9	37.6 ± 1.8	0.06 ± 0.03
	GloVe	No	37.2 ± 1.9	35.6 ± 2.6	36.4 ± 2.0	0.05 ± 0.03
		Yes	37.5 ± 1.7	36.1 ± 2.2	36.8 ± 1.8	0.05 ± 0.03
Agr	Keras	No	38.3 ± 1.6	37.1 ± 1.9	38.0 ± 1.6	0.07 ± 0.02
	Word2Vec	No	39.6 ± 2.0	38.5 ± 2.1	39.3 ± 2.0	0.09 ± 0.03
		Yes	**40.0 ± 1.6**	**38.8 ± 1.8**	**39.7 ± 1.6**	**0.10 ± 0.02**
	GloVe	No	38.8 ± 1.9	38.0 ± 1.7	38.6 ± 1.8	0.08 ± 0.03
		Yes	38.3 ± 1.5	37.3 ± 1.2	38.0 ± 1.4	0.07 ± 0.02
Emot	Keras	No	36.0 ± 1.0	35.0 ± 1.7	35.9 ± 1.1	0.04 ± 0.02
	Word2Vec	No	38.0 ± 1.6	37.7 ± 1.5	38.0 ± 1.6	0.07 ± 0.02
		Yes	**38.1 ± 1.5**	**37.8 ± 1.5**	**38.1 ± 1.5**	**0.07 ± 0.02**
	GloVe	No	35.2 ± 1.7	34.2 ± 1.6	35.1 ± 1.6	0.03 ± 0.02
		Yes	35.1 ± 1.3	34.5 ± 1.1	35.0 ± 1.2	0.02 ± 0.02

Table 6.36. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from YouTube Personality considering a LSTM architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	30	66	4	73	20	7	64	26	10	42	41	17	32	50	18
	MP	26	71	3	67	25	8	61	30	9	30	50	20	27	50	23
	HP	26	71	3	67	21	13	61	22	17	28	51	21	24	50	26
W2v-F	LP	38	56	6	63	23	14	45	33	22	**54**	28	18	**40**	32	28
	MP	28	62	10	54	27	19	43	33	24	37	**43**	20	31	**41**	28
	HP	26	62	12	52	25	23	37	28	35	41	36	**23**	34	33	**33**
Glv-F	LP	**37**	56	7	**64**	19	17	53	33	14	47	34	19	39	38	24
	MP	27	**62**	11	51	**27**	22	48	34	18	36	44	20	34	40	26
	HP	24	59	**17**	49	25	**26**	43	33	24	39	37	24	31	42	27
W2v-NF	LP	39	53	8	60	23	17	**51**	32	17	52	31	17	40	35	25
	MP	32	60	8	54	26	20	45	**36**	19	36	44	20	31	42	27
	HP	32	60	8	50	25	25	40	32	**28**	39	39	22	32	37	31
Glv-NF	LP	39	57	4	64	22	14	52	33	15	48	31	21	40	37	23
	MP	31	64	5	55	28	17	48	37	15	37	41	22	34	39	27
	HP	28	65	7	49	27	24	44	36	20	36	38	26	33	40	27

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

6.2.3 Results with the English dataset from PAN-AP-2015

In this case, we took into account the database per Tweet (14152 samples for training and 13178 for testing), in order to have more samples when training the architectures. The predictions of the neural networks are given per Tweet, but the reported metrics correspond to the performance on the test set taking into account the database per subject (152 samples for training and 142 for testing). We went from the predictions per Tweet to the predictions and metrics per subject as follows: for the regression case we considered the average of each subject's predictions, for the bi-class and tri-class classi-

fication case we considered the mode of the subject's predictions. Thus, for example, for the case of bi-class classification, if 70 out of 100 predictions (mode) for a subject say that the subject belongs to class 0 (weak presence of the trait), then the subject is predicted as belonging to class 0. The regression case works in a similar way, where the average is taken instead of the mode, since the predictions are real values and not binary values as in the case of bi-class classification.

Personality trait estimation

The results taking into account deep learning methods and using English Tweets can be found in Table 6.37 and Table 6.38. As can be seen, similar to the results with the YouTube database, better results are obtained when considering pre-trained embeddings, but with the difference that it is better not to freeze the embedding layer, since for example for the CNN architecture, in the 5 traits of the OCEAN model the best results were achieved taking into account this configuration and for the LSTM architecture the best result was achieved for two traits: extraversion and emotional stability. Comparing the performance of CNNs with the performance of LSTMs, it is observed that architectures based on CNNs are better for this regression task for the five traits since they obtain better results than LSTMs, and comparing the results obtained with these deep learning methods with respect to those obtained with SVR (see Table 6.11), we can observe that the classical methods achieved better results in general, but, with the CNN architecture some metrics were improved: Pearson's correlation coefficient, r, by 0.047 for the conscientiousness trait and by 0.131 for agreeableness trait and the Spearman correlation coefficient, ρ, by 0.072 for agreeableness trait.

Table 6.37. Results for personality trait estimation with the English dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	No	0.531	0.550	0.126	0.147
	Word2Vec	**No**	**0.545**	**0.625**	**0.122**	**0.141**
		Yes	0.515	0.518	0.124	0.142
	GloVe	No	0.564	0.598	0.123	0.141
		Yes	0.533	0.532	0.127	0.143
Cons	Keras	No	0.632	0.674	0.111	0.136
	Word2Vec	No	0.604	0.617	0.111	0.135
		Yes	0.523	0.520	0.114	0.138
	GloVe	**No**	**0.648**	**0.671**	**0.109**	**0.134**
		Yes	0.554	0.565	0.112	0.137
Extr	Keras	No	0.368	0.413	0.120	0.152
	Word2Vec	**No**	**0.400**	**0.382**	**0.119**	**0.151**
		Yes	0.269	0.282	0.123	0.156
	GloVe	No	0.371	0.355	0.122	0.153
		Yes	0.426	0.433	0.120	0.154
Agr	Keras	No	0.495	0.397	0.110	0.143
	Word2Vec	No	0.470	0.372	0.109	0.144
		Yes	0.436	0.307	0.111	0.145
	GloVe	**No**	**0.580**	**0.447**	**0.109**	**0.142**
		Yes	0.506	0.391	0.111	0.145
Emot	Keras	No	0.586	0.590	0.171	0.204
	Word2Vec	**No**	**0.622**	**0.619**	**0.173**	**0.204**
		Yes	0.583	0.597	0.174	0.206
	GloVe	No	0.590	0.590	0.173	0.206
		Yes	0.537	0.507	0.176	0.209

Table 6.38. Results for personality trait estimation with the English dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	**No**	**0.445**	**0.465**	**0.127**	**0.145**
	Word2Vec	No	0.452	0.459	0.129	0.149
		Yes	0.432	0.431	0.131	0.149
	GloVe	No	0.495	0.506	0.127	0.147
		Yes	0.412	0.387	0.131	0.150
Cons	Keras	**No**	**0.552**	**0.615**	**0.114**	**0.138**
	Word2Vec	No	0.475	0.474	0.116	0.141
		Yes	0.428	0.397	0.119	0.144
	GloVe	No	0.591	0.604	0.112	0.136
		Yes	0.512	0.512	0.118	0.142
Extr	Keras	No	0.178	0.146	0.124	0.157
	Word2Vec	No	0.310	0.192	0.124	0.156
		Yes	0.334	0.245	0.124	0.156
	GloVe	**No**	**0.253**	**0.174**	**0.122**	**0.154**
		Yes	0.286	0.165	0.123	0.154
Agr	Keras	No	0.342	0.288	0.114	0.149
	Word2Vec	No	0.320	0.326	0.113	0.148
		Yes	0.344	0.200	0.116	0.151
	GloVe	No	0.275	0.260	0.115	0.151
		Yes	**0.367**	**0.347**	**0.111**	**0.147**
Emot	Keras	No	0.515	0.573	0.180	0.212
	Word2Vec	**No**	**0.521**	**0.522**	**0.175**	**0.207**
		Yes	0.517	0.500	0.180	0.212
	GloVe	No	0.498	0.528	0.180	0.214
		Yes	0.503	0.528	0.178	0.210

Weak vs. strong presence of each trait

The bi-class classification results between weak presence and strong presence considering architectures with CNN and LSTM can be found in Table 6.39 and Table 6.40 respectively. Comparing the results between the two architectures we observed that the CNN architecture gave better results for openness to experience, extraversion and agreeableness traits; while for conscientiousness and emotional stability traits, better results were achieved with the LSTM architecture. The best result between the two architectures was achieved for openness to experience trait with accuracy = 72.5% and AUC = 0.75, followed by the emotional stability trait with accuracy = 70.4% and AUC = 0.75; while the lowest result was obtained for the agreeableness trait with accuracy = 55.6% and AUC = 0.52. Compared to the SVM results (see Table 6.12), no improvement was achieved, however the results are similar, especially for example for emotional stability trait (0.6% difference in accuracy). Apart from the above, it is worth noting that some of the low results in terms of the specificity metric may be due to the fact that the neural networks is overfitted towards the 'strong presence' class, as there is a considerably higher number of subjects belonging to this class compared to the number of subjects in the 'weak presence' class for some traits (see Table 5.2).

Table 6.39. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	71.8	95.8	23.4	66.6	0.79
	Word2Vec	No	69.7	97.9	12.8	61.6	0.79
		Yes	68.3	97.9	8.5	58.9	0.73
	GloVe	**No**	**72.5**	**97.9**	**21.3**	**66.5**	**0.75**
		Yes	68.3	98.9	6.4	57.9	0.71
Cons	Keras	No	64.8	94.5	33.3	61.0	0.81
	Word2Vec	No	66.2	58.9	73.9	66.0	0.76
		Yes	**67.6**	**61.6**	**73.9**	**67.5**	**0.75**
	GloVe	No	56.3	17.8	97.1	48.4	0.80
		Yes	64.1	42.5	86.9	62.3	0.75
Extr	Keras	No	56.3	69.9	42.0	55.5	0.62
	Word2Vec	No	59.9	68.5	50.7	59.5	0.66
		Yes	57.0	71.2	42.0	56.1	0.59
	GloVe	No	57.0	95.9	15.9	48.7	0.60
		Yes	**60.6**	**87.7**	**31.9**	**57.1**	**0.59**
Agr	Keras	No	51.4	100.0	0.0	34.9	0.53
	Word2Vec	No	53.5	100.0	4.3	39.5	0.52
		Yes	**55.6**	**100.0**	**8.7**	**43.7**	**0.52**
	GloVe	No	51.4	100.0	0.0	34.9	0.52
		Yes	52.1	100.0	1.4	36.5	0.52
Emot	Keras	**No**	**70.4**	**85.7**	**43.1**	**68.9**	**0.75**
	Word2Vec	No	67.6	84.6	37.3	65.6	0.73
		Yes	62.0	92.3	7.8	53.1	0.71
	GloVe	No	64.1	93.4	11.8	56.1	0.64
		Yes	68.3	84.6	39.2	66.5	0.71

Table 6.40. Results for bi-class system: weak presence vs strong presence of the trait with the English dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	71.1	98.9	15.0	63.3	0.77
	Word2Vec	No	68.3	97.9	8.5	58.9	0.73
		Yes	67.6	97.9	6.4	57.5	0.70
	GloVe	**No**	**71.8**	**98.9**	**17.0**	**64.6**	**0.70**
		Yes	69.0	98.9	8.5	59.3	0.71
Cons	Keras	**No**	**70.4**	**49.3**	**92.8**	**69.0**	**0.83**
	Word2Vec	No	67.6	52.1	84.1	66.8	0.81
		Yes	64.8	58.9	71.0	64.7	0.69
	GloVe	No	52.1	9.6	97.1	41.0	0.74
		Yes	58.7	26.0	91.3	52.9	0.71
Extr	Keras	No	58.5	61.6	55.1	58.4	0.64
	Word2Vec	**No**	**59.2**	**83.6**	**33.3**	**56.3**	**0.61**
		Yes	58.5	21.9	97.1	51.8	0.65
	GloVe	No	56.3	23.3	91.3	50.8	0.65
		Yes	53.5	31.5	76.8	51.1	0.63
Agr	Keras	**No**	**52.1**	**100.0**	**1.5**	**36.5**	**0.52**
	Word2Vec	No	51.4	98.6	1.5	36.1	0.50
		Yes	51.4	100.0	0.0	34.9	0.49
	GloVe	No	50.7	98.6	0.0	34.6	0.54
		Yes	51.4	100.0	0.0	34.9	0.48
Emot	Keras	No	66.9	76.9	49.0	66.5	0.75
	Word2Vec	**No**	**70.4**	**72.5**	**66.7**	**70.8**	**0.75**
		Yes	64.1	100.0	0.0	50.1	0.59
	GloVe	No	61.3	47.3	86.3	61.2	0.77
		Yes	63.4	98.9	0.0	49.7	0.59

The ROC curves of this bi-class experiment between weak presence and strong presence of personality traits are presented in Figure 6.7. As we can observe, in general, in terms of the AUC metric, the values obtained with the CNN were superior to those obtained with the LSTM for openness to

experience, extraversion and emotional stability traits; while the LSTM was superior to the CNN for conscientiousness and agreeableness traits. We can also observe that in terms of AUC, for the embedding layer, the Keras word embeddings trained from scratch were useful, since they are in the highest AUC results for the 5 traits (see dark blue and brown curves).

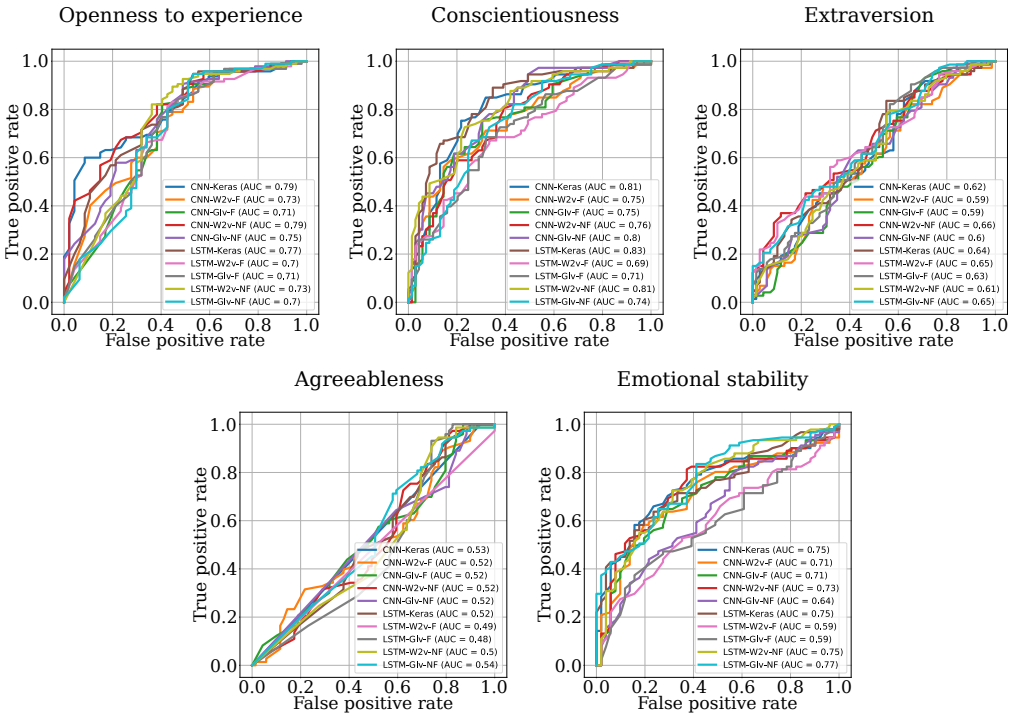

Figure 6.7. ROC curves obtained for English dataset from PAN-AP-2015 considering CNN and LSTM architectures. Keras: Keras embedding layer trained from scratch, W2v: pre-trained Word2Vec embeddings, Glv: pre-trained GloVe embeddings, F: Freezing the embedding layer, NF: No freezing the embedding layer.

Classification of personality traits into 3 levels

The tri-class classification results considering English Twitter data and architectures based on CNNs and LSTMs can be found in Table 6.41 and Table 6.43 respectively. We observed a general behavior for the results with LSTMs: for the five traits of the OCEAN model, the best results were obtained considering pre-trained embeddings and without freezing the embedding layer (3 traits with Word2Vec pre-trained embeddings and 2 traits with

GloVe pre-trained embeddings). For the case of the architecture with CNNs, the best results were obtained with different configurations: for 3 traits the best performance was obtained with the GloVe pre-trained embeddings and for the other 2 traits the best performance was obtained taking into account the Keras embeddings trained from scratch. Now, if we directly compare the results obtained with CNNs and the results obtained with LSTMS, we can observe that CNNs obtained better results for conscientiousness, extraversion and agreeableness traits; while for openness to experience and emotional stability traits, better results were achieved considering LSTMs. The confusion matrices can be found in Table 6.42 and Table 6.44. Directly comparing the performance of CNNs and LSTMs in terms of the confusion matrices, we can observe that LSTMs perform better in 3 of the 5 traits: openness to experience, conscientiousness and emotional stability; while CNNs performed better in extraversion and agreeableness traits. We can also observe that for the extraversion and emotional stability traits, there is an overfitting of the classifiers towards the 'high presence' class, because they classify this class very well but cannot discriminate the subjects of the 'low presence' or 'medium presence' classes, perhaps because the number of subjects of the 'high presence' class to train the systems is considerably higher than the number of samples of the other two classes (see Table 5.2).

Table 6.41. Tri-class classification results with the English dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	47.9	39.6	47.0	0.21
	Word2Vec	No	48.6	41.9	47.8	0.22
		Yes	50.0	42.3	49.1	0.24
	GloVe	No	45.8	37.4	44.8	0.18
		Yes	**52.8**	**45.6**	**52.0**	**0.28**
Cons	Keras	**No**	**45.8**	**44.1**	**43.4**	**0.14**
	Word2Vec	No	40.8	38.8	42.7	0.12
		Yes	33.8	28.7	32.8	-0.02
	GloVe	No	35.2	28.1	38.2	0.07
		Yes	40.1	38.2	39.3	0.07
Extr	Keras	**No**	**52.8**	**37.9**	**35.0**	**0.04**
	Word2Vec	No	51.4	34.9	33.3	0.00
		Yes	50.7	34.6	32.9	-0.01
	GloVe	No	51.4	34.9	33.3	0.00
		Yes	51.4	34.9	33.3	0.00
Agr	Keras	No	53.5	41.5	37.2	0.07
	Word2Vec	No	60.6	54.6	47.9	0.25
		Yes	57.0	48.6	42.6	0.16
	GloVe	**No**	**64.8**	**60.9**	**54.1**	**0.35**
		Yes	54.2	44.6	39.2	0.10
Emot	Keras	No	36.6	19.6	33.3	0.00
	Word2Vec	No	36.6	19.6	33.3	0.00
		Yes	36.6	19.6	33.3	0.00
	GloVe	No	36.6	20.8	33.5	0.00
		Yes	**36.6**	**20.9**	**33.5**	**0.00**

Table 6.42. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from PAN-AP-2015 considering a CNN architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	51	0	49	**14**	52	34	**0**	0	100	6	0	94	0	0	100
	MP	17	2	80	0	**73**	28	0	**5**	95	0	9	91	0	0	100
	HP	12	0	88	0	56	**44**	0	0	**100**	2	2	96	0	0	100
W2v-F	LP	55	0	45	0	62	38	0	0	100	12	0	88	0	0	100
	MP	20	4	76	0	73	27	0	0	100	0	20	80	0	0	100
	HP	12	0	88	0	74	26	1	0	99	1	3	96	0	0	100
Glv-F	LP	**62**	0	38	14	45	41	0	0	100	12	0	88	**3**	0	97
	MP	22	**6**	72	0	70	30	0	0	100	0	11	89	4	**0**	96
	HP	12	0	**88**	0	66	34	0	0	100	1	4	95	2	0	**98**
W2v-NF	LP	53	0	47	21	62	17	0	0	100	18	0	82	0	0	100
	MP	20	6	74	0	80	20	0	0	100	0	31	69	0	0	100
	HP	16	0	84	3	70	27	0	0	100	1	4	95	0	0	100
Glv-NF	LP	43	0	57	7	90	3	0	0	100	**24**	0	76	3	0	97
	MP	15	2	83	0	93	7	0	0	100	0	**46**	54	0	0	100
	HP	10	0	90	0	85	15	0	0	100	3	4	**93**	2	0	98

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

Table 6.43. Tri-class classification results with the English dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	49.4	48.1	49.4	0.24
	Word2Vec	**No**	**57.0**	**56.9**	**57.3**	**0.36**
		Yes	43.7	34.3	42.6	0.14
	GloVe	No	42.3	32.5	41.2	0.12
		Yes	49.4	47.5	49.4	0.24
Cons	Keras	No	36.6	32.2	39.3	0.07
	Word2Vec	**No**	**38.7**	**34.2**	**42.1**	**0.11**
		Yes	42.3	40.9	39.2	0.06
	GloVe	No	43.0	41.3	40.0	0.07
		Yes	31.0	19.0	34.8	0.02
Extr	Keras	No	50.7	34.6	32.9	-0.01
	Word2Vec	**No**	**52.1**	**36.4**	**34.1**	**0.02**
		Yes	51.4	34.9	33.3	0.00
	GloVe	No	51.4	34.9	33.3	0.00
		Yes	51.4	34.9	33.3	0.00
Agr	Keras	No	56.3	46.7	41.1	0.14
	Word2Vec	No	57.0	47.2	41.6	0.15
		Yes	52.1	36.4	34.3	0.02
	GloVe	**No**	**55.6**	**51.5**	**46.6**	**0.21**
		Yes	51.4	37.2	34.3	0.02
Emot	Keras	No	37.3	21.1	34.2	0.01
	Word2Vec	No	36.6	20.9	33.5	0.00
		Yes	36.6	19.6	33.3	0.00
	GloVe	**No**	**45.8**	**34.2**	**44.9**	**0.17**
		Yes	36.6	19.6	33.3	0.00

Table 6.44. Confusion matrix for the classification of personality traits into 3 levels with the English dataset from PAN-AP-2015 considering a LSTM architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	43	4	53	14	72	14	0	0	100	12	0	88	3	0	97
	MP	13	26	61	0	85	15	0	0	100	0	14	86	0	0	100
	HP	8	12	80	0	81	19	1	0	99	1	1	98	0	0	100
W2v-F	LP	30	0	70	14	34	52	0	0	100	0	0	100	0	0	100
	MP	9	2	89	0	60	40	0	0	100	0	3	97	0	0	100
	HP	4	0	96	0	56	44	0	0	100	0	0	100	0	0	100
Glv-F	LP	34	15	51	0	97	3	0	0	100	0	0	100	0	0	100
	MP	15	31	54	0	98	2	0	0	100	0	6	94	0	0	100
	HP	6	10	84	0	93	7	0	0	100	1	1	98	0	0	100
W2v-NF	LP	**68**	32	0	**17**	69	14	**0**	0	100	12	0	88	3	0	97
	MP	28	**61**	11	0	**90**	10	0	**2**	98	0	14	86	2	0	98
	HP	26	31	**43**	0	81	**19**	0	0	**100**	1	0	99	2	0	98
Glv-NF	LP	26	0	74	14	27	59	0	0	100	**15**	9	76	**38**	0	62
	MP	7	2	91	0	63	37	0	0	100	0	**46**	54	37	**0**	63
	HP	4	0	96	0	56	44	0	0	100	1	19	**80**	4	0	**96**

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

6.2.4 Results with the Spanish dataset from PAN-AP-2015

Similar to the case of the English database, in this case of Spanish language, the predictions of the neural networks are given per Tweet, but the metrics reported correspond to the performance on the test set taking into account the database per subject (9132 samples for training and 7729 for testing). See subsection 6.2.3 for more details on how to switch from prediction per Tweet to prediction per subject.

Personality trait estimation

The results of the regression experiment considering Spanish Tweets and considering CNN and LSTM architectures can be found in Table 6.45 and Table 6.46 respectively. As can be seen, it is useful if pre-trained word embeddings are considered (especially those based on GloVe) and the embedding layer is frozen, since with this configuration the best results were achieved in the 5 traits of the OCEAN model for the CNN architecture and in 3 of the 5 traits for the LSTM architecture. If we directly compare the results obtained with the CNN and those obtained with the LSTM, we can observe that the CNN obtained the best results for the 5 traits; now, if we compare the results of these two deep learning methods (CNN and LSTM architectures) for this regression task with the results of the classic method (SVR), it is observed that in general the SVR performs better (see Table 6.16), but with certain exceptions such as: i) taking into account a CNN-based architecture, it was possible to improve the r by 0.018 for the openness to experience trait, also the r, ρ was improved by 0.061 and 0.056 respectively for agreeableness trait and likewise the r, ρ was improved by 0.039 and 0.076 respectively for the emotional stability trait; and ii) taking into account an architecture based on LSTM, an improvement was achieved for the trait emotional stability by 0.048 and 0.046 respectively for r and ρ respectively.

Table 6.45. Results for personality trait estimation with the Spanish dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	No	0.521	0.462	0.102	0.126
	Word2Vec	No	0.533	0.478	0.101	0.126
		Yes	0.487	0.455	0.104	0.131
	GloVe	**No**	**0.560**	**0.482**	**0.101**	**0.126**
		Yes	0.220	0.167	0.106	0.134
Cons	Keras	No	0.675	0.716	0.121	0.149
	Word2Vec	No	0.632	0.698	0.122	0.151
		Yes	0.682	0.725	0.142	0.172
	GloVe	**No**	**0.665**	**0.697**	**0.118**	**0.147**
		Yes	0.519	0.624	0.133	0.161
Extr	Keras	No	0.665	0.658	0.141	0.181
	Word2Vec	No	0.664	0.668	0.140	0.180
		Yes	0.684	0.706	0.143	0.183
	GloVe	**No**	**0.701**	**0.703**	**0.140**	**0.180**
		Yes	0.629	0.663	0.145	0.186
Agr	Keras	No	0.699	0.726	0.137	0.168
	Word2Vec	**No**	**0.743**	**0.749**	**0.134**	**0.163**
		Yes	0.654	0.677	0.136	0.168
	GloVe	No	0.712	0.714	0.135	0.166
		Yes	0.572	0.585	0.142	0.173
Emot	Keras	No	0.606	0.581	0.166	0.190
	Word2Vec	**No**	**0.618**	**0.637**	**0.163**	**0.187**
		Yes	0.592	0.636	0.167	0.192
	GloVe	No	0.559	0.561	0.165	0.190
		Yes	0.592	0.530	0.169	0.196

Table 6.46. Results for personality trait estimation with the Spanish dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	r	ρ	MAE	RMSE
Open	Keras	**No**	**0.424**	**0.407**	**0.105**	**0.132**
	Word2Vec	No	0.242	0.214	0.107	0.133
		Yes	0.392	0.367	0.106	0.132
	GloVe	No	0.323	0.371	0.106	0.136
		Yes	0.119	0.084	0.108	0.135
Cons	Keras	No	0.640	0.691	0.126	0.155
	Word2Vec	No	0.574	0.634	0.140	0.171
		Yes	0.610	0.662	0.133	0.160
	GloVe	**No**	**0.545**	**0.582**	**0.125**	**0.154**
		Yes	0.522	0.579	0.136	0.164
Extr	Keras	No	0.611	0.674	0.147	0.189
	Word2Vec	No	0.545	0.614	0.146	0.187
		Yes	**0.639**	**0.684**	**0.146**	**0.186**
	GloVe	No	0.532	0.591	0.145	0.188
		Yes	0.546	0.597	0.149	0.190
Agr	Keras	No	0.659	0.679	0.140	0.173
	Word2Vec	No	0.510	0.557	0.141	0.174
		Yes	0.471	0.508	0.144	0.176
	GloVe	**No**	**0.592**	**0.600**	**0.138**	**0.171**
		Yes	0.456	0.511	0.146	0.177
Emot	Keras	No	0.600	0.565	0.170	0.197
	Word2Vec	**No**	**0.627**	**0.601**	**0.172**	**0.197**
		Yes	0.548	0.558	0.174	0.200
	GloVe	No	0.563	0.524	0.175	0.201
		Yes	0.588	0.587	0.173	0.199

Weak vs. strong presence of each trait

The bi-class classification results between weak presence and strong presence of personality traits are found in Table 6.47 and Table 6.48. For both architectures (based on CNN and based on LSTM) it was useful to use pre-trained word embeddings (especially those based on Word2Vec) to achieve better results; since for the CNN architecture, for 3 of the 5 traits the best results were achieved taking into account the pre-trained word embeddings and freezing the embedding layer; while in the case of the LSTM architecture, also for 3 of the 5 traits of the OCEAN model the best results were achieved taking into account the pre-trained word embeddings but without freezing the embedding layer. If we directly compare the performance of the architecture based on CNNs with the performance of the architecture based on LSTM, we observe that they have a similar performance but the LSTM architecture presented better performance in 3 of the 5 traits: conscientiousness, extraversion and emotional stability. Now, if we compare the performance of the CNNs and LSTMs with respect to the performance of the SVM (see Table 6.17), we notice that in general, the results with the SVM are superior, except for the conscientiousness trait, where we were able to improve the accuracy percentage by 5% considering an architecture based on LSTM. We can also observe that some specificity values are very low and the sensitivity values are very high in some cases (especially for openness to experience and agreeableness traits), this may be due to the fact that the classifiers present a type of overfitting towards the samples of the 'strong presence' class, and this is because in the training of the models, there was a considerably larger quantity of the samples of the 'strong presence' class compared to the quantity of samples of the 'weak presence' class (see Table 5.3).

Table 6.47. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	50.6	100.0	0.0	34.0	0.50
	Word2Vec	No	49.4	97.5	0.0	33.5	0.58
		Yes	49.4	97.5	0.0	33.5	0.51
	GloVe	No	49.4	97.5	0.0	33.5	0.60
		Yes	**50.6**	**100.0**	**0.0**	**34.0**	**0.63**
Cons	Keras	No	55.7	41.1	91.3	56.1	0.74
	Word2Vec	No	45.6	30.3	82.6	45.0	0.70
		Yes	**65.8**	**60.7**	**78.3**	**67.4**	**0.71**
	GloVe	No	63.3	58.9	73.9	65.0	0.70
		Yes	62.0	55.3	78.3	63.7	0.73
Extr	Keras	No	51.9	100.0	0.0	35.5	0.88
	Word2Vec	No	59.5	97.6	18.4	51.7	0.86
		Yes	**68.4**	**95.1**	**39.5**	**65.5**	**0.91**
	GloVe	No	52.0	100.0	0.0	35.5	0.83
		Yes	58.2	100.0	13.2	48.2	0.87
Agr	Keras	No	58.2	100.0	0.0	42.9	0.49
	Word2Vec	No	72.2	97.8	36.4	68.6	0.85
		Yes	62.0	100.0	9.1	51.0	0.85
	GloVe	**No**	**74.7**	**95.7**	**45.5**	**72.5**	**0.86**
		Yes	62.0	97.8	12.1	52.5	0.75
Emot	Keras	**No**	**72.2**	**57.1**	**89.2**	**71.5**	**0.83**
	Word2Vec	No	57.0	100.0	8.1	44.9	0.83
		Yes	60.8	90.5	27.0	56.1	0.82
	GloVe	No	67.1	92.9	37.8	64.2	0.86
		Yes	67.1	100.0	29.7	62.1	0.82

Table 6.48. Results for bi-class system: weak presence vs strong presence of the trait with the Spanish dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	Sensitivity	Specificity	F1-score	AUC
Open	Keras	No	50.6	100.0	0.0	34.0	0.50
	Word2Vec	**No**	**50.6**	**100.0**	**0.0**	**34.0**	**0.51**
		Yes	50.6	100.0	0.0	34.0	0.46
	GloVe	No	50.6	100.0	0.0	34.0	0.47
		Yes	50.6	100.0	0.0	34.0	0.50
Cons	Keras	No	58.2	25.0	91.3	41.8	0.71
	Word2Vec	**No**	**65.8**	**62.5**	**73.9**	**67.4**	**0.71**
		Yes	40.5	19.6	91.3	36.3	0.68
	GloVe	No	38.0	16.1	91.3	32.5	0.68
		Yes	58.2	48.2	82.6	59.6	0.71
Extr	Keras	No	57.0	100.0	10.5	45.8	0.92
	Word2Vec	No	65.8	95.1	34.2	65.2	0.89
		Yes	62.0	97.6	23.7	55.8	0.89
	GloVe	No	57.0	97.6	13.2	47.4	0.88
		Yes	**82.3**	**80.5**	**84.2**	**82.3**	**0.88**
Agr	Keras	No	58.2	100.0	0.0	42.9	0.81
	Word2Vec	No	59.5	100.0	3.0	45.7	0.81
		Yes	60.8	100.0	6.1	48.3	0.77
	GloVe	No	59.5	100.0	3.0	45.7	0.79
		Yes	**62.0**	**100.0**	**9.1**	**50.9**	**0.74**
Emot	Keras	No	69.6	50.0	91.9	68.4	0.87
	Word2Vec	No	50.6	100.0	0.0	34.0	0.83
		Yes	54.4	100.0	2.7	39.7	0.77
	GloVe	**No**	**77.2**	**92.9**	**59.5**	**76.4**	**0.87**
		Yes	64.6	100.0	24.3	58.2	0.82

In Figure 6.8 are the ROC curves. In terms of the AUC metric, CNN-based architectures generally perform better for openness to experience, conscientiousness, and agreeableness traits compared to LSTM-based architectures. Similarly, the word embeddings that gave the best performance (also

in terms of AUC) are the GloVe-based word embeddings, since in general, the best performance was obtained where these word embeddings are took into account (with and without freezing the embedding layer), see for example curves in dark green and purple for the five traits.

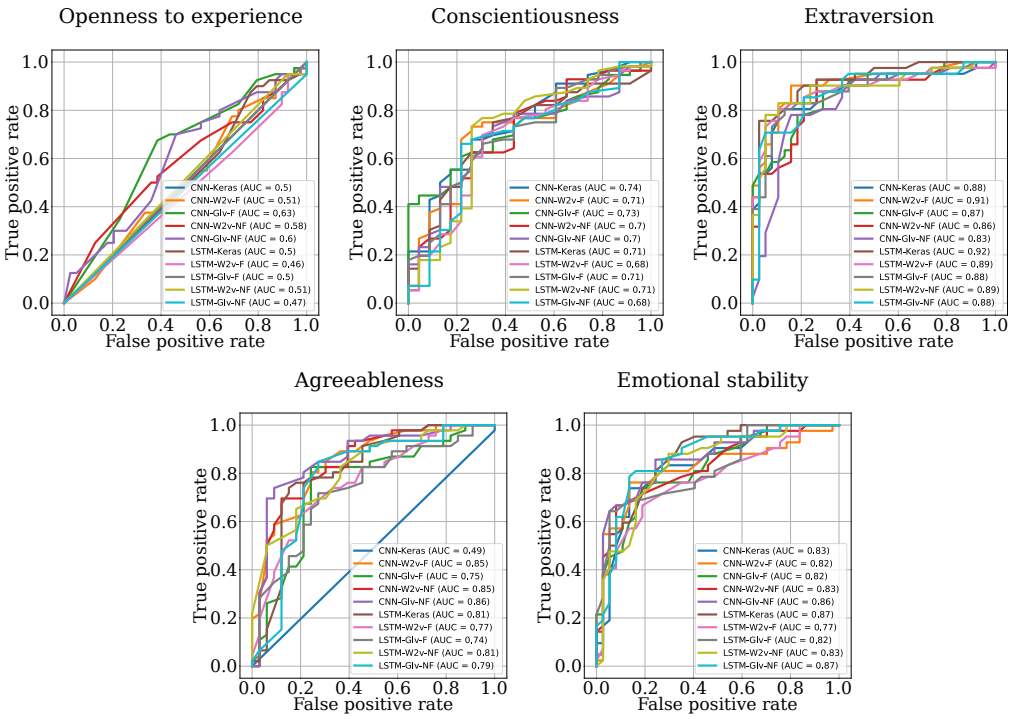

Figure 6.8. ROC curves obtained for Spanish dataset from PAN-AP-2015 considering CNN and LSTM architectures. Keras: Keras embedding layer trained from scratch, W2v: pre-trained Word2Vec embeddings, Glv: pre-trained GloVe embeddings, F: Freezing the embedding layer, NF: No freezing the embedding layer.

Classification of personality traits into 3 levels

The metrics of the results of the tri-class classification experiment between low presence, medium presence and high presence are in Table 6.49 and Table 6.51. As we can see, it was very useful for both architectures (especially the based on CNNs) to use the pre-trained word embeddings of Word2Vec and GloVe without freezing the embedding layer since, for example, for the CNN architecture the best results were achieved with this settings for 4 of the 5 traits (except for emotional stability trait). For the LSTM architecture,

something was introduced that had not happened in previous experiments: openness to experience, extraversion and emotional stability traits perform better when considering Keras embeddings trained from scratch. Also, directly comparing CNN with LSTM, we can see that CNN performance is higher for 3 of the 5 traits of the OCEAN model: openness to experience, extraversion and agreeableness. The confusion matrices for this tri-class experiment are in Table 6.50 and Table 6.52. Comparing the two tables, we can see that the architecture based on CNN allows to obtain better results compared to the architecture based on LSTM, since for 4 traits (except for the conscientiousness trait) better performance is obtained in terms of confusion matrix. We can also observe that the classifiers (both with CNNs and LSTMs architectures) have an overfitting towards the class 'low presence' for openness to experience, conscientiousness and extraversion traits; since it classified the samples belonging to this class very well, but it is not able to distinguish the samples of the classes 'medium presence' or 'high presence'; this may be due to the fact that the number of samples (when the neural networks were trained) for the class 'low presence' for these three traits is considerably higher than the number of samples for the other two classes (see Table 5.3).

Table 6.49. Tri-class classification results with the Spanish dataset from PAN-AP-2015 considering a CNN architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	No	49.4	32.6	33.3	0.00
	Word2Vec	No	49.4	32.6	33.3	0.00
		Yes	49.4	32.6	33.3	0.00
	GloVe	**No**	**50.6**	**35.3**	**35.1**	**0.03**
		Yes	49.4	32.6	33.0	0.00
Cons	Keras	No	58.2	44.5	32.6	0.01
	Word2Vec	No	58.2	44.9	32.6	0.03
		Yes	59.5	46.9	36.3	0.07
	GloVe	**No**	**57.0**	**48.1**	**34.9**	**0.13**
		Yes	57.0	45.4	34.9	0.04
Extr	Keras	No	67.1	64.6	59.0	0.44
	Word2Vec	**No**	**68.4**	**68.8**	**65.9**	**0.50**
		Yes	57.0	47.9	43.9	0.20
	GloVe	No	62.0	54.5	49.7	0.32
		Yes	53.2	41.8	39.5	0.13
Agr	Keras	No	44.3	38.1	39.4	0.11
	Word2Vec	**No**	**51.9**	**45.3**	**47.1**	**0.25**
		Yes	44.3	36.7	41.9	0.15
	GloVe	No	44.3	37.6	38.6	0.10
		Yes	43.0	31.9	35.4	0.04
Emot	Keras	**No**	**57.0**	**51.1**	**50.8**	**0.31**
	Word2Vec	No	45.6	32.1	36.9	0.07
		Yes	44.3	35.7	36.6	0.06
	GloVe	No	54.4	45.7	45.2	0.23
		Yes	45.6	33.6	37.1	0.07

Table 6.50. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from PAN-AP-2015 considering a CNN architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	100	0	0	98	2	0	95	5	0	61	39	0	**89**	11	0
	MP	100	0	0	100	0	0	55	39	6	42	58	0	42	**58**	0
	HP	100	0	0	91	9	0	43	14	43	45	55	0	61	33	**6**
W2v-F	LP	100	0	0	98	2	0	100	0	0	33	67	0	25	75	0
	MP	100	0	0	89	11	0	94	6	0	8	92	0	15	85	0
	HP	100	0	0	87	13	0	70	4	26	10	90	0	6	94	0
Glv-F	LP	100	0	0	94	6	0	100	0	0	91	9	0	14	86	0
	MP	100	0	0	89	11	0	94	6	0	85	15	0	3	97	0
	HP	100	0	0	87	13	0	78	9	13	90	10	0	0	100	0
W2v-NF	LP	100	0	0	98	2	0	**79**	18	3	**61**	39	0	11	89	0
	MP	100	0	0	100	0	0	22	**67**	11	19	**81**	0	0	100	0
	HP	100	0	0	87	13	0	17	31	**52**	15	85	**0**	0	100	0
Glv-NF	LP	**100**	0	0	**94**	6	0	100	0	0	70	30	0	36	64	0
	MP	100	**0**	0	89	**11**	0	78	5	17	54	46	0	0	100	0
	HP	95	0	**5**	52	48	**0**	48	9	43	55	45	0	0	100	0

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

Table 6.51. Tri-class classification results with the Spanish dataset from PAN-AP-2015 considering a LSTM architecture.

Trait	Embedding	Freezing of the Embedding Layer	Accuracy	F1-score	UAR	κ
Open	Keras	**No**	**50.6**	**35.3**	**34.9**	**0.03**
	Word2Vec	No	49.4	32.6	33.3	0.00
		Yes	49.4	32.6	33.3	0.00
	GloVe	No	49.4	32.6	33.3	0.00
		Yes	49.4	32.6	33.3	0.00
Cons	Keras	No	58.2	44.5	32.6	0.01
	Word2Vec	**No**	**59.5**	**47.1**	**36.3**	**0.08**
		Yes	59.5	45.1	33.3	0.03
	GloVe	No	58.2	44.5	32.6	0.01
		Yes	58.2	44.5	32.6	0.01
Extr	Keras	**No**	**68.4**	**65.6**	**59.9**	**0.47**
	Word2Vec	No	65.8	61.3	55.2	0.39
		Yes	57.0	46.6	43.5	0.20
	GloVe	No	63.3	56.7	51.5	0.34
		Yes	55.7	46.3	42.8	0.18
Agr	Keras	No	39.2	28.0	38.4	0.09
	Word2Vec	No	50.6	43.2	44.5	0.20
		Yes	41.8	33.4	39.9	0.11
	GloVe	**No**	**51.9**	**44.9**	**46.3**	**0.24**
		Yes	40.5	33.6	38.0	0.08
Emot	Keras	**No**	**59.5**	**51.8**	**51.6**	**0.34**
	Word2Vec	No	58.2	53.8	51.6	0.32
		Yes	51.9	46.7	45.0	0.21
	GloVe	No	40.5	28.2	37.4	0.07
		Yes	43.0	27.3	34.5	0.02

Table 6.52. Confusion matrix for the classification of personality traits into 3 levels with the Spanish dataset from PAN-AP-2015 considering a LSTM architecture (results in %).

		Open			**Cons**			**Extr**			**Agr**			**Emot**		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Keras	LP	**100**	0	0	98	2	0	**97**	3	0	15	85	0	**82**	18	0
	MP	95	**5**	0	100	0	0	44	**39**	17	0	100	0	27	**73**	0
	HP	100	0	**0**	91	9	0	35	22	**43**	5	95	0	33	67	**0**
W2v-F	LP	100	0	0	100	0	0	100	0	0	27	73	0	54	46	0
	MP	100	0	0	100	0	0	89	0	11	8	92	0	24	76	0
	HP	100	0	0	91	9	0	70	0	30	15	85	0	17	78	5
Glv-F	LP	100	0	0	98	2	0	100	0	0	33	67	0	4	96	0
	MP	100	0	0	100	0	0	89	11	0	19	81	0	0	100	0
	HP	100	0	0	91	9	0	74	9	17	20	80	0	0	100	0
W2v-NF	LP	100	0	0	**98**	1	0	100	0	0	**76**	24	0	68	32	0
	MP	100	0	0	89	**11**	0	72	22	6	42	**58**	0	24	76	0
	HP	100	0	0	83	17	**0**	57	0	43	50	50	**0**	33	56	11
Glv-NF	LP	100	0	0	98	1	0	100	0	0	70	30	0	100	0	0
	MP	100	0	0	100	0	0	78	11	11	31	69	0	88	12	0
	HP	100	0	0	91	9	0	57	0	43	30	70	0	83	17	0

LP: Low presence, **MP:** Medium presence, **HP:** High presence. **Keras:** Keras embedding layer trained from scratch. **W2v:** pre-trained Word2Vec embeddings. **Glv:** pre-trained GloVe embeddings. **F:** Freezing the embedding layer. **NF:** No freezing the embedding layer.

6.3 Graphical summary of the best results

The best results for the YouTube Personality dataset and PAN-AP-2015 dataset along the regression and classification experiments obtained in sections 6.1 and 6.2 are summarized in Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12. The first column of sub-figures shows the regression results. It can be observed that the regressors did a good job: with respect to the YouTube Personality dataset, in conscientiousness, extraversion and agreeableness traits, a Spearman's correlation above 0.35 was obtained. For the case of PAN-AP-2015 dataset, the Spearman's correlation values are above

0.50 for openness to experience, conscientiousness and emotional stability traits.

In the second and third columns the resulting representation spaces from the bi-class and tri-class scenarios are shown, respectively. Note that in the bi-class scenario the figures illustrate the result of applying a dimensionality reduction based on Principal Component Analysis (PCA). Notice the high dispersion of the samples along the representation space. This is one of the reasons for the low accuracies found in the classification experiments. For this case of bi-class classification and considering the YouTube Personality dataset, it is again presented that the conscientiousness, extraversion and agreeableness traits obtained the best performance in terms of percentage accuracy, with performances $\geq 60\%$. In the case of the PAN-AP-2015 dataset, the extraversion and emotional stability traits have accuracy percentages $\geq 70\%$.

Finally, the tri-class scenario is shown in the third column of sub-figures, where three different colors are used to represent the three classes: LP, MP, and HP. These are the representations resulting from a dimensionality reduction technique based on the Linear Discriminant Analysis (LDA) algorithm, which represents the projection of the feature space to a two-dimensional space that contributes the most to the separation of the classes. Even though the results appear to be low, the representation spaces show that the three sub-groups are found both for YouTube Personality and PAN-AP-2015 dataset. When considering the YouTube data, better results are obtained for agreeableness and conscientiousness traits, with F1-score $\geq 44\%$. While with Twitter data, better results are obtained for extraversion and agreeableness traits with F1-score $\geq 65\%$.

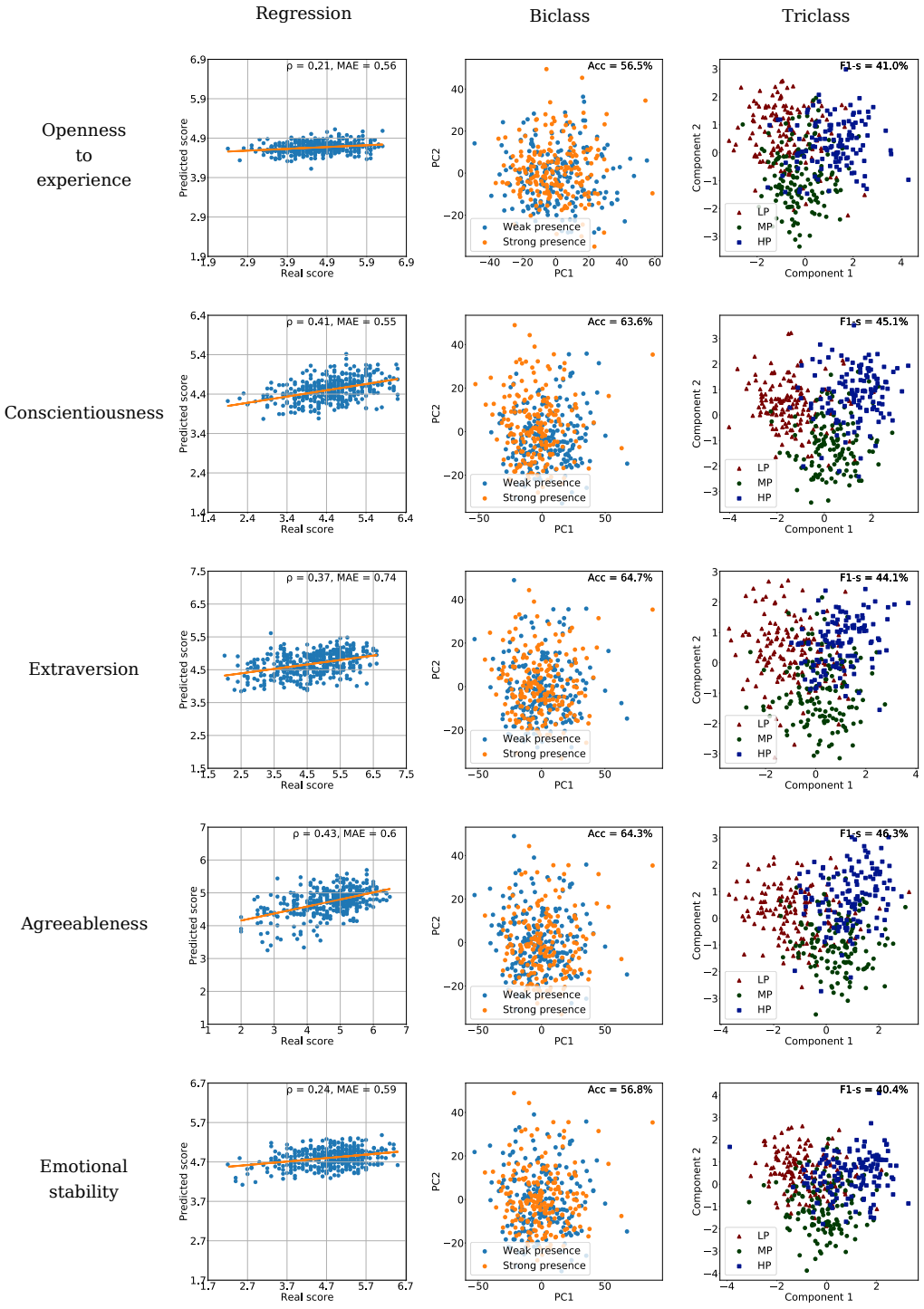

Figure 6.9. Graphical summary of the best results with the English dataset from YouTube Personality.

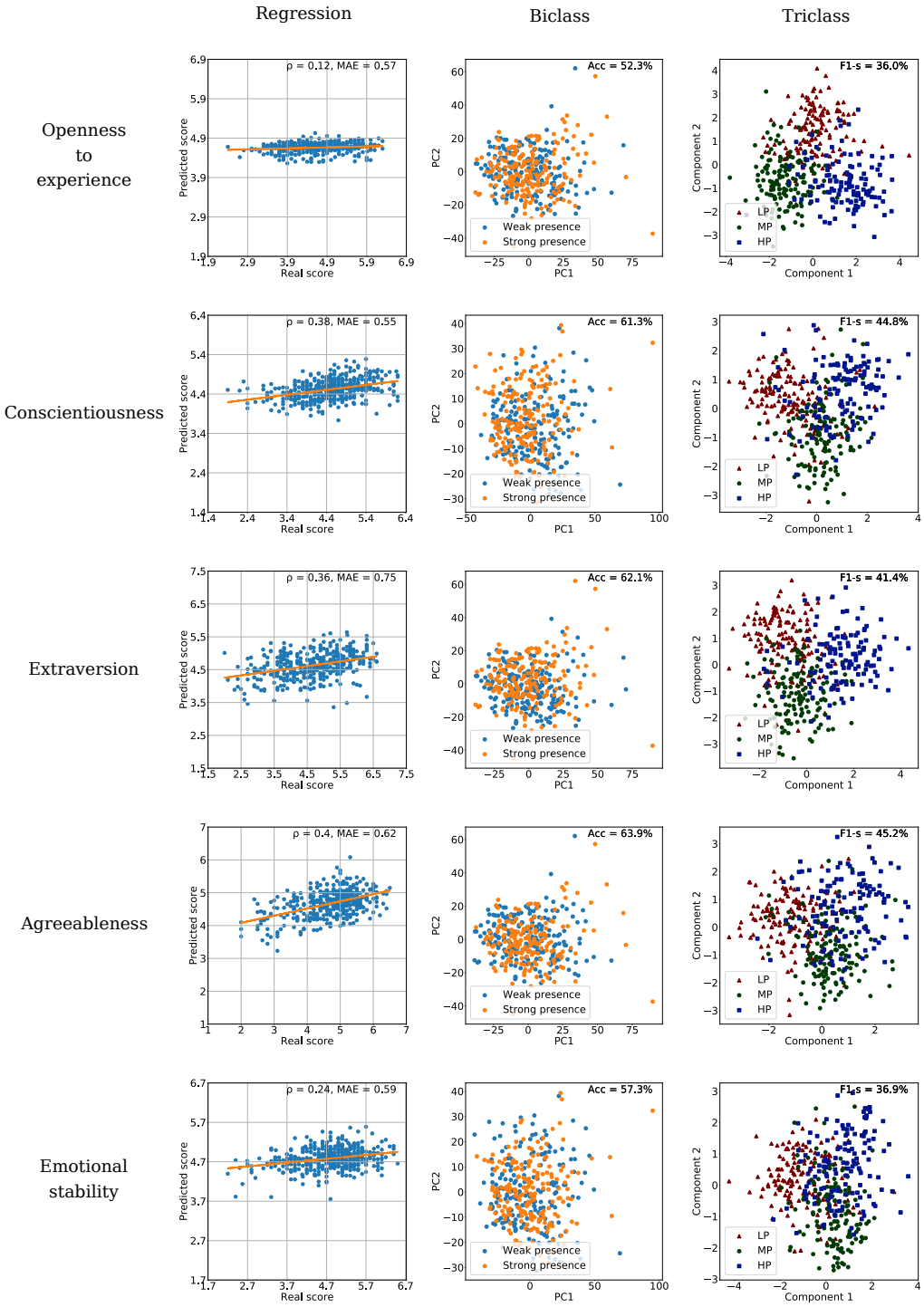

Figure 6.10. Graphical summary of the best results with the Spanish dataset from YouTube Personality.

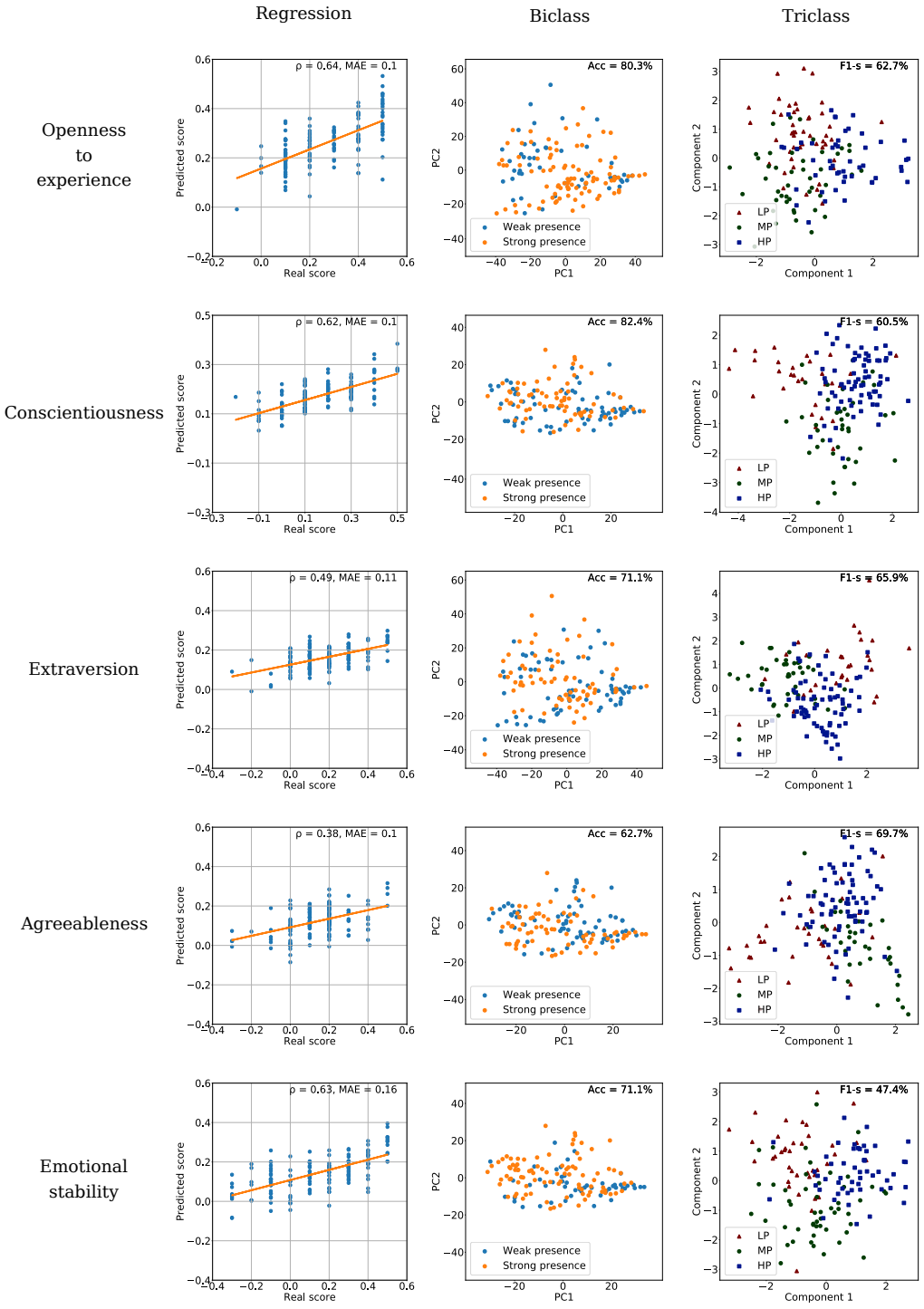

Figure 6.11. Graphical summary of the best results with the English dataset from PAN-AP-2015.

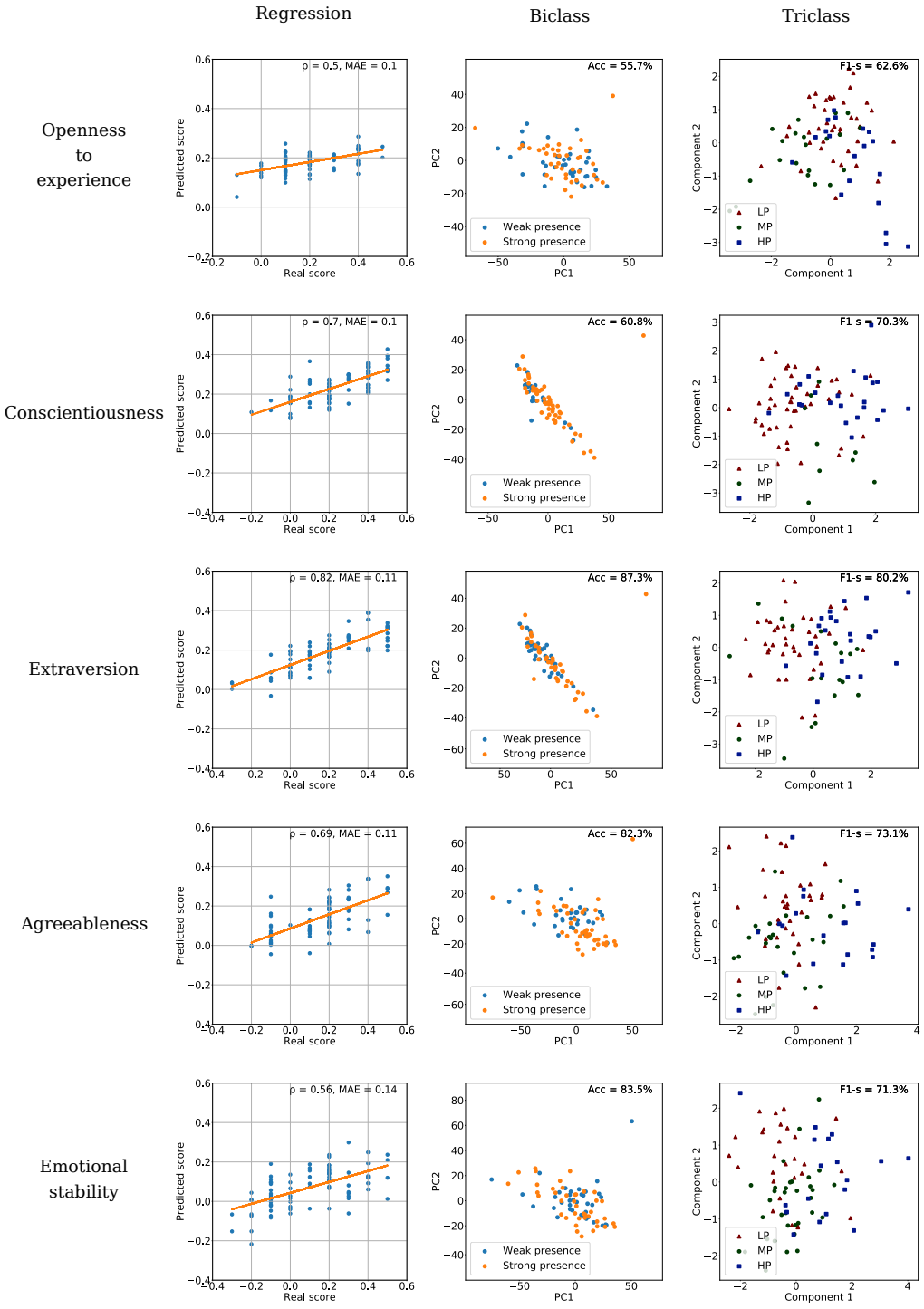

Figure 6.12. Graphical summary of the best results with the Spanish dataset from PAN-AP-2015.

Chapter 7

Conclusions and future work

7.1 Conclusions about classical machine learning methods

For the transliterations of YouTube Personality dataset in English language the ideal is to work with the fusion of the word embeddings obtained with Word2Vec and GloVe, since the best results were obtained with them as features from the text. Only for the case of bi-class classification it was useful to make use of the word embeddings obtained from BERT, since in the case of regression and tri-class classification, the features from the method based on transformers did not show improvements compared to the classical word embeddings (Word2Vec or GloVe). However, for the case of the Spanish transliterations of the YouTube Personality dataset, it was very useful to consider the embeddings obtained with transformer-based models (i.e. BERT and BETO), since in most cases, better results were achieved for the five traits of the OCEAN model compared to the results taking into account the classical embeddings (Word2Vec, GloVe or their fusion). Now, regarding the most difficult personality trait to estimate/predict in the YouTube Personality dataset, we have the openness to experience trait, followed by the emotional stability trait, both for English and Spanish transliterations.

For the case of English Tweets from PAN-AP-2015 dataset, in general, the embeddings obtained with the GloVe method were more useful compared with the embeddings based on Word2Vec or BERT for estimate/predict the OCEAN personality traits. For the case of the PAN-AP-2015 dataset in Spanish language, in order to obtain better results, it was better to use the word embeddings obtained with models based on transformers (BERT or

BETO) or with the fusion of the word embeddings obtained with Word2Vec and GloVe. Now, regarding the most difficult trait to estimate/predict in the PAN-AP-2015 dataset, we have the agreeableness trait when considering Tweets in English language and openness to experience trait when considering Tweets in Spanish language.

7.2 Conclusions about deep learning methods

According to the results with CNNs and LSTMs considering the English YouTube Personality dataset, it is recommended to make use of the word embeddings obtained with Word2Vec and to freeze the embedding layer in order to obtain better results in the three main experiments of this work: estimation of the values of the personality traits (regression experiments), classification between weak presence and strong presence (bi-class classification experiments) and classification between low presence, medium presence and high presence (tri-class classification experiments). For the Spanish YouTube Personality dataset, it is recommended to make use of the pre-trained word embeddings of Word2Vec and GloVe and to freeze the embedding layer, except for the case of tri-class classification experiments considering CNNs architectures, where it is better not to freeze the embedding layer. Now, considering the deep learning architectures we considered in this work (based on CNNs and LSTMs), the most difficult trait to predict/estimate for the YouTube Personality dataset in both languages (English and Spanish) is emotional stability trait, followed by openness to experience trait.

When considering the Tweets from PAN-AP-2015 dataset in English language and considering deep learning techniques, we can conclude that for LSTM architectures (and also for the specific case of regression experiment with CNN architecture), it was useful to take into account the pre-trained word embeddings but without freezing the embedding layer, i.e. retraining the weights of this layer. For the case of PAN-AP-2015 in Spanish language, we can conclude according to the results in the regression, bi-class classification and tri-class classification experiments that it is recommended for both architectures based in CNNs and LSTMs, to take into account the pre-trained word embeddings of Word2Vec and GloVe as weights of the embedding layer and without doing freezing of the embedding layer. Also, we can conclude that when considering deep learning methods, the most difficult trait to predict considering the English PAN-AP-2015 dataset is the extraversion trait,

while the openness to experience trait is the most difficult one for the Spanish PAN-AP-2015 dataset.

7.3 General conclusions and future line of work

Comparing the results of the classical machine learning methods (SVM) with the results of the deep learning methods (CNNs and LSTMs) we can observe that for this work, where we estimated the five personality traits of the OCEAN model defined in psychology and considering YouTube transliterations and Twitter data, the classical learning methods allow in a general way to obtain better results for the 3 main experiments of this work: regression, bi-class classification and tri-class classification. This may be due, among other things, to the fact that the number of samples available for both databases are not large enough to correctly train the architectures based on CNNs and LSTMs, and also to the fact that in some cases, there was a large imbalance in the number of samples of one class compared to the number of the other classes (for the classification cases), which may negatively affect the training of neural networks. Now, we can also conclude that for the YouTube Personality dataset, it is more difficult to estimate/classify the 5 personality traits in Spanish language compared to the English language; while for the PAN-AP-2015 dataset, for 3 of the 5 traits of the OCEAN model (extraversion, agreeableness and emotional stability) it is easier to estimate/classify the personality traits in Spanish language compared to the English language.

As future work, we plan to use data augmentation techniques to obtain a larger number of samples or get larger databases in order to improve the performance of deep learning methods, as these have worked very well with problems similar to the ones we work on here. Similarly, we also plan to work with other types of features that can be extracted from the text, such as Part-of-Speech (PoS) tagging features, other types of word embeddings that for example consider information from the topics that are developed in the texts such as LDA2VEC, BERTopic; and also DOC2VEC, which is other type of embedding that allows obtaining fixed-dimension embeddings for a whole document [80]–[83].

Appendix A

Publications emerging from the development of this master's thesis

A.1 Journals

✓ **López-Pabón, F. O.**, & Orozco-Arroyave, J. R. (2022). Automatic personality evaluation from transliterations of YouTube vlogs using classical and state of the art word embeddings. *Ingeniería e Investigación*, 42(3). In press.

A.2 Book Chapters

✓ **López-Pabón, F. O.**, & Orozco-Arroyave, J. R. (2021, October). Evaluation of Different Word Embeddings to Create Personality Models in Spanish. In *Workshop on Engineering Applications* (pp. 121-132). Springer, Cham.

List of Figures

Bibliography

[1] G. W. Allport, "Personality: A psychological interpretation.," 1937.

[2] R. P. Tett, D. N. Jackson, and M. Rothstein, "Personality measures as predictors of job performance: A meta-analytic review," *Personnel psychology*, vol. 44, no. 4, pp. 703–742, 1991.

[3] J. K. White, S. S. Hendrick, and C. Hendrick, "Big five personality variables and relationship constructs," *Personality and individual differences*, vol. 37, no. 7, pp. 1519–1530, 2004.

[4] A. Vinciarelli and G. Mohammadi, "A survey of personality computing," *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 273–291, 2014.

[5] O. P. John, R. W. Robins, and L. A. Pervin, *Handbook of personality: Theory and research*. Guilford Press, 2010.

[6] R. D. Oliveira, M. Cherubini, and N. Oliver, "Influence of personality on satisfaction with mobile phone services," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 20, no. 2, pp. 1–23, 2013.

[7] A. Laleh and R. Shahram, "Analyzing facebook activities for personality recognition," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017, pp. 960–964.

[8] F. Rangel, P. Rosso, M. Potthast, B. Stein, and W. Daelemans, "Overview of the 3rd author profiling task at pan 2015," in *CLEF*, sn, 2015, p. 2015.

[9] E. L. Kelly and J. J. Conley, "Personality and compatibility: A prospective analysis of marital stability and marital satisfaction.," *Journal of personality and social psychology*, vol. 52, no. 1, p. 27, 1987.

[10] L. R. Goldberg, "An alternative "description of personality": The big-five factor structure.," *Journal of personality and social psychology*, vol. 59, no. 6, p. 1216, 1990.

[11] Y. Mao, D. Zhang, C. Wu, K. Zheng, and X. Wang, "Feature analysis and optimisation for computational personality recognition," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, IEEE, 2018, pp. 2410–2414.

[12] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, "Recent trends in deep learning based personality detection," *Artificial Intelligence Review*, pp. 1–27, 2019.

[13] B. B. C. da Silva and I. Paraboni, "Personality recognition from facebook text," in *International Conference on Computational Processing of the Portuguese Language*, Springer, 2018, pp. 107–114.

[14] F. Celli, "Unsupervised personality recognition for social network sites," *Proc. of ICDS. Valencia*, 2012.

[15] B. Verhoeven and W. Daelemans, "Clips stylometry investigation (csi) corpus: A dutch corpus for the detection of age, gender, personality, sentiment and deception in text.," in *LREC*, 2014, pp. 3081–3085.

[16] K. Luyckx and W. Daelemans, "Using syntactic features to predict author personality from text," *Proceedings of digital humanities*, vol. 2008, pp. 146–9, 2008.

[17] M. Kocher and J. Savoy, "Unine at clef 2015: Author identification," *Working notes papers of the CLEF*, 2015.

[18] I. Pervaz, I. Ameer, A. Sittar, and R. M. A. Nawab, "Identification of author personality traits using stylistic features: Notebook for pan at clef 2015.," in *CLEF (Working Notes)*, Citeseer, 2015.

[19] M. Arroju, A. Hassan, and G. Farnadi, "Age, gender and personality recognition using tweets in a multilingual setting," in *6th Conference and Labs of the Evaluation Forum (CLEF 2015): Experimental IR meets multilinguality, multimodality, and interaction*, 2015, pp. 23–31.

[20] B. Verhoeven, B. Plank, and W. Daelemans, "Multilingual personality profiling on twitter," in *To be presented at DHBenelux 2016*, 2016.

[21] B. Verhoeven, W. Daelemans, and B. Plank, "Twisty: A multilingual twitter stylometry corpus for gender and personality profiling," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 1632–1637.

[22] J. W. Pennebaker and L. A. King, "Linguistic styles: Language use as an individual difference.," *Journal of personality and social psychology*, vol. 77, no. 6, p. 1296, 1999.

[23] D. Stillwell and M. Kosinski, *Mypersonality project website*, 2015.

[24] B. Plank and D. Hovy, "Personality traits on twitter—or—how to get 1,500 personality tests in a week," in *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2015, pp. 92–98.

[25] J.-I. Biel, V. Tsiminaki, J. Dines, and D. Gatica-Perez, "Hi youtube! personality impressions and verbal content in social video," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, 2013, pp. 119–126.

[26] M. Hassanein, W. Hussein, S. Rady, and T. F. Gharib, "Predicting personality traits from social media using text semantics," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, IEEE, 2018, pp. 184–189.

[27] D. Quercia, M. Kosinski, D. Stillwell, and J. Crowcroft, "Our twitter profiles, our selves: Predicting personality with twitter," in *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, IEEE, 2011, pp. 180–185.

[28] B. Y. Pratama and R. Sarno, "Personality classification based on twitter text using naive bayes, knn and svm," in *2015 International Conference on Data and Software Engineering (ICoDSE)*, IEEE, 2015, pp. 170–174.

[29] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.

[30] D. Xue, L. Wu, Z. Hong, *et al.*, "Deep learning-based personality recognition from text posts of online social networks," *Applied Intelligence*, vol. 48, no. 11, pp. 4232–4246, 2018.

[31] H. Jiang, X. Zhang, and J. D. Choi, "Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 13 821–13 822.

[32] C. Sarkar, S. Bhatia, A. Agarwal, and J. Li, "Feature analysis for computational personality recognition using youtube personality data set," in *Proceedings of the 2014 ACM multi media on workshop on computational personality recognition*, 2014, pp. 11–14.

[33] F. Alam and G. Riccardi, "Predicting personality traits using multimodal information," in *Proceedings of the 2014 ACM multi media on workshop on computational personality recognition*, 2014, pp. 15–18.

[34] K. G. Das and D. Das, "Developing lexicon and classifier for personality identification in texts," in *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, 2017, pp. 362–372.

[35] X. Sun, B. Liu, Q. Meng, J. Cao, J. Luo, and H. Yin, "Group-level personality detection based on text generated networks," *World Wide Web*, pp. 1–20, 2019.

[36] Z. Guan, B. Wu, B. Wang, and H. Liu, "Personality2vec: Network representation learning for personality," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2020, pp. 30–37.

[37] J. Salminen, R. G. Rao, S.-g. Jung, S. A. Chowdhury, and B. J. Jansen, "Enriching social media personas with personality traits: A deep learning approach using the big five classes," in *International Conference on Human-Computer Interaction*, Springer, 2020, pp. 101–120.

[38] M. A. Alvarez-Carmona, A. P. López-Monroy, M. Montes-y-Gómez, L. Villasenor-Pineda, and H. Jair-Escalante, "Inaoe's participation at pan'15: Author profiling task," *Working Notes Papers of the CLEF*, 2015.

[39] M. Giménez, R. Paredes, and P. Rosso, "Personality recognition using convolutional neural networks," in *International Conference on Computational Linguistics and Intelligent Text Processing*, Springer, 2017, pp. 313–323.

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

[41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013.

[42] X. Rong, "Word2vec parameter learning explained," *arXiv preprint arXiv:1411.2738*, 2014.

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

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

[45] Rani Horev. "Bert explained: State of the art language model for nlp." [Online; accessed 25-March-2021]. (2018), [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[46] Jay Alammar. "The illustrated transformer." [Online; accessed 28-March-2021]. (2018), [Online]. Available: http://jalammar.github.io/illustrated-transformer/.

[47] Jason Brownlee. "How to use word embedding layers for deep learning with keras." [Online; accessed 12-April-2021]. (2017), [Online]. Available: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/.

[48] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Citeseer, 2010.

[49] Google, *Tool for computing continuous distributed representations of words*, [Online; accessed 28-March-2021], 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/.

[50] Radim Rehurek, *How to download pre-trained models and corpora*, [Online; accessed 28-March-2021], 2019. [Online]. Available: https://radimrehurek.com/gensim/auto_examples/howtos/run_downloader_api.html.

[51] Jeffrey Pennington, Richard Socher, Christopher D. Manning, *Glove: Global vectors for word representation*, [Online; accessed 20-March-2021], 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/.

[52] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*, Citeseer, 2010.

[53] Wikimedia, *Data downloads*, [Online; accessed 20-March-2021], 2018. [Online]. Available: https://dumps.wikimedia.org/eswiki/latest/.

[54] Jay Alammar, *The illustrated word2vec*, [Online; accessed 25-March-2021], 2019. [Online]. Available: http://jalammar.github.io/illustrated-word2vec/.

[55] P. A. Perez-Toro, *PauPerezT/WEBERT: Word Embeddings using BERT*, urlhttps://doi.org/10.5281/zenodo.3964244, version V0.0.1, Jul. 2020. DOI: 10.5281/zenodo.3964244.

[56] J. Canete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish pre-trained bert model and evaluation data," *PML4DC at ICLR*, vol. 2020, 2020.

[57] Sandipan Dey. "Implementing a soft-margin kernelized support vector machine binary classifier with quadratic programming in r and python." [Online; accessed 22-Feb-2020]. (2018), [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/implementing-a-soft-margin-kernelized-support-vector-machine.

[58] B. Schölkopf, A. J. Smola, F. Bach, *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[59] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[60] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[61] V. Ranković, N. Grujović, D. Divac, and N. Milivojević, "Development of support vector regression identification model for prediction of dam structural behaviour," *Structural Safety*, vol. 48, pp. 33–39, 2014.

[62] Daphne Cornelisse. "An intuitive guide to convolutional neural networks." [Online; accessed 10-April-2021]. (2018), [Online]. Available: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/.

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

[64] S. Gehrmann, F. Dernoncourt, Y. Li, *et al.*, "Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives," *PloS one*, vol. 13, no. 2, 2018.

[65] Pedro Torres Perez. "Deep learning: Recurrent neural networks." [Online; accessed 10-May-2021]. (2018), [Online]. Available: https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010.

[66] A. A. Ismail, T. Wood, and H. C. Bravo, "Improving long-horizon forecasts with expectation-biased lstm networks," *arXiv preprint arXiv*, 2018.

[67] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality data?," 2016.

[68] S. D. Gosling, P. J. Rentfrow, and W. B. Swann Jr, "A very brief measure of the big-five personality domains," *Journal of Research in personality*, vol. 37, no. 6, pp. 504–528, 2003.

[69] S. Loria, "Textblob documentation," *Release 0.15*, vol. 2, 2018.

[70] B. Rammstedt and O. P. John, "Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german," *Journal of research in Personality*, vol. 41, no. 1, pp. 203–212, 2007.

[71] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, Montreal, Canada, vol. 14, 1995, pp. 1137–1145.

[72] Ajitesh Kumar. "Hold-out method for training machine learning models." [Online; accessed 14-April-2021]. (2020), [Online]. Available: https://vitalflux.com/hold-out-method-for-training-machine-learning-model/.

[73] Lund Research Ltd. "Pearson product-moment correlation." [Online; accessed 10-March-2021]. (2018), [Online]. Available: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php.

[74] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[75] M. R. Junge and J. R. Dettori, "Roc solid: Receiver operator characteristic (roc) curves as a foundation for better diagnostic tests," *Global spine journal*, vol. 8, no. 4, pp. 424–429, 2018.

[76] M. L. McHugh, "Interrater reliability: The kappa statistic," *Biochemia medica: Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[77] O.-M. Sulea and D. Dichiu, "Automatic profiling of twitter users based on their tweets: Notebook for pan at clef 2015.," in *CLEF (Working Notes)*, 2015.

[78] L. Miculicich, "Statistical learning methods for profiling analysis," in *Proceedings of CLEF*, Citeseer, 2015.

[79] C. E. González-Gallardo, A. Montes, G. Sierra, J. A. Núnez-Juárez, A. J. Salinas-López, and J. Ek, "Tweets classification using corpus dependent tags, character and pos n-grams.," in *CLEF (working notes)*, 2015.

[80] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, PMLR, 2014, pp. 1188–1196.

[81] C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," *arXiv preprint arXiv:1605.02019*, 2016.

[82] M. Grootendorst, *Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics.* Version v0.9.4, 2020. DOI: 10.5281/zenodo.4381785. [Online]. Available: https://doi.org/10.5281/zenodo.4381785.

[83] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 23, e5909, 2021.