



# Automatic Personality Evaluation from Transliterations of YouTube Vlogs Using Classical and State-of-the-Art Word Embeddings

## Evaluación automática de la personalidad a partir de transliteraciones de vlogs de YouTube mediante el uso de incrustaciones de palabras clásicas y del estado del arte

Felipe O. López-Pabón <sup>1</sup>, Juan R. Orozco-Arroyave <sup>2</sup>

### ABSTRACT

The study of automatic personality recognition has gained attention in the last decade thanks to a variety of applications deriving from this field. The Big Five model (also known as OCEAN) constitutes a well-known method to label different personality traits. This work considered transliterations of video recordings collected from YouTube (originally provided by the Idiap research institute) and automatically generated scores for the Big Five personality traits, which were also in the database. The transliterations were modeled with three different word embedding approaches (Word2Vec, GloVe, and BERT) and three different levels of analysis, namely a regression to predict the score of each personality trait, a binary classification between the strong vs. weak presence of each trait, and a tri-class classification according to three different levels of manifestations in each trait (low, medium, and high). According to our findings, the proposed approach provides similar results to others reported in the specialized literature. We believe that further research is required to find better results. Our results, as well as others reported in the literature, suggest that there is a big gap in the study of personality traits based on linguistic patterns, which highlights the need to work on collecting and labeling data considering the knowledge of expert psychologists and psycholinguists.

**Keywords:** personality, word embeddings, YouTube, regression, classification

### RESUMEN

El estudio del reconocimiento automático de la personalidad ha ganado atención en la última década gracias a las diversas aplicaciones que se derivan de este campo. El modelo de los cinco grandes (también conocido como OCEAN) constituye un método ampliamente conocido para etiquetar diferentes rasgos de personalidad. Este trabajo consideró transliteraciones de grabaciones de vídeo recogidas de YouTube (proporcionadas originalmente por el instituto de investigación Idiap) y puntuaciones generadas automáticamente para los cinco grandes rasgos de personalidad, las cuales también se encontraban en la base de datos. Las transliteraciones se modelaron con tres enfoques diferentes de incrustación de palabras (Word2Vec, GloVe y BERT) y se incluyeron tres niveles diferentes de análisis, a saber: regresión para predecir la puntuación de cada rasgo de personalidad, clasificación binaria entre presencia fuerte y débil de cada rasgo, y una clasificación tri-clase según tres niveles diferentes de manifestaciones en cada rasgo (bajo, medio y alto). Según nuestros resultados, el enfoque propuesto proporciona resultados similares a otros reportados en la literatura especializada. Creemos que es necesario seguir investigando para encontrar mejores resultados. Nuestros resultados, así como otros reportados en la literatura, sugieren que existe un gran vacío en el estudio de los rasgos de personalidad basados en patrones lingüísticos, lo cual resalta la necesidad de trabajar en la recolección y etiquetado de datos considerando el conocimiento de psicólogos y psicolingüistas expertos.

**Palabras clave:** personalidad, incrustaciones de palabras, YouTube, regresión, clasificación

**Received:** February 21th, 2021

**Accepted:** August 18th, 2021

### Introduction

Text analysis and natural language processing have emerged as a very useful sub-area of artificial intelligence, which allows extracting valuable information from text and performing a specific task. Some applications of these areas include web page classification (Onan, 2015), text document classification (Onan *et al.*, 2016a; Onan, 2017a), text genre classification (Onan, 2016), text document clustering (Onan, 2017b), and, more recently, topic extraction modeling (Onan, 2019a) and opinion mining (Onan, 2019b). Similarly, one of the most frequent tasks in text analysis is sentiment classification (Onan and Korukoğlu, 2015; Onan *et al.*, 2016b; Onan *et al.*, 2016c; Onan, 2018; Onan, 2020). Regarding the most used types of features in text analysis, we found: Bag of Words (BoW) (Onan and Korukoğlu, 2015;

Korukoğlu and Bulut, 2016c), latent topics obtained with Latent Dirichlet Allocation (LDA) (Onan *et al.*, 2016b), five

<sup>1</sup>Electronics Engineer, Universidad de Antioquia, Colombia. Affiliation: Master's student in Telecommunications Engineering and teaching assistant, Universidad de Antioquia, Colombia. E-mail: forlando.lopez@udea.edu.co

<sup>2</sup>Electronics Engineer, Universidad de Antioquia, Colombia. M.Sc Telecommunications Engineering, Universidad de Antioquia, Colombia. Ph.D. in Computer Science, Friedrich-Alexander-Universität, Erlangen, Germany. Affiliation: Associate Professor, Universidad de Antioquia, Colombia and Adjunct Researcher the the Pattern Recognition Lab, Friedrich-Alexander-Universität, Erlangen, Germany. E-mail: rafael.orozco@udea.edu.co

**How to cite:** 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(2), e93803. <https://doi.org/10.15446/ing.investig.93803>



Attribution 4.0 International (CC BY 4.0) Share - Adapt

main categories from Linguistic Inquiry and Word Count (LIWC) (Onan, 2018), and, more recently, word embeddings such as Word2Vec, FastText, GloVe, LDA2vec, DOC2vec, and Term Frequency - Inverse Document Frequency (TF-IDF) weighted Global Vectors for word representation (GloVe) (Onan, 2020). Regarding the use of machine learning methods, the following have been widely used in the literature: i) classical learning methods such as Naive Bayes (NB) (Onan, 2015; Onan *et al.*, 2016a), K-Nearest Neighbors (KNN) (Onan, 2016), Support Vector Machines (SVMs) (Onan, 2017a, 2017b), and Logistic Regression (LR) (Onan, 2019b); and ii) deep learning methods such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Bidirectional RNNs with attention mechanisms (RNNAM), Gated Recurrent Units (GRUs), Long short-term memory (LSTMs) (Onan, 2019b), and a combination of CNNs and LSTMs (Onan, 2020). Ensemble learning methods such as bagging, adaptive boosting, and random subspace have also been used (Onan *et al.*, 2016b; Onan *et al.*, 2016c; Onan, 2018).

Another important task related to text analysis which has emerged in the last decade is the study of personality, which plays an important role in human interaction and is defined as the combination of the various behavioral characteristics, emotions, motivations, and thinking patterns of an individual (Allport, 1937). 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 (White *et al.*, 2004; Vinciarelli and Mohammadi, 2014; Xue *et al.*, 2017). Nowadays, the increasing amount of information that users publish online allow the study of different personality traits with considerable volumes of data. One of the specific sources of information is the text that people publish in their status updates, tweets, blogs, vlogs (video-blogs), and reviews (Mohammad and Kiritchenko, 2013). From the psychological point of view, one suitable and accepted way of assessing personality traits is the Big Five Factor Model of personality dimensions. This is a well-known scale that evaluates the presence of five personality traits (John *et al.*, 2008). The list below shows the five basic traits included in it and their corresponding social aspects (John *et al.*, 1991; Celli *et al.*, 2014). Note that the resulting acronym is OCEAN, which gives the other widely used name of the model.

Openness to experience: intellectual vs. unimaginative.

Conscientiousness: self-disciplined vs. careless.

Extraversion: sociable vs. shy.

Agreeableness: friendly vs. uncooperative.

Neuroticism (the inverse of emotional stability): neurotic vs. calm.

The automatic classification of personality traits has a wide variety of applications ranging from cognitive-based market segmentation to human health evaluation (Cambria *et al.*, 2017). Different methods to model personality traits from texts directly generated by the person (*i.e.*, social media posts) or from transliterations generated from audio or video

recordings have been proposed in the literature. Celli (2012) classified the five personality traits of the OCEAN model considering linguistic features extracted from status updates of a social network called FriendFeed. The author found an average accuracy of 61,3% for the five traits. Similarly, in Hassanein *et al.* (2018), the authors classified the presence vs. absence of the same five personality traits in posts belonging to the myPersonality dataset (Kosinski *et al.*, 2015). The authors extracted semantic and morphological features and reported an average accuracy of 64%. Different machine learning methods were used in Pratama and Sarno (2015) to classify different personalities from different datasets. The authors created models based on TF-IDF and reported accuracies of up to 60% with data from Facebook posts and 65% with Twitter posts (Tweets). Later, also working with the myPersonality dataset, Mao *et al.* (2018) used different classifiers, including KNN, NB, and Decision Trees (DT), in order to classify the personality traits. The features considered by the authors include temporal measures (*e.g.*, frequency of status updates per day), social network measures (*e.g.*, network size), morphological features (*e.g.*, frequency of adjectives), and TF-IDF-based features. According to their findings, TF-IDF features are suitable for classifying personality traits and the best reported F-measure was 79% for the 'openness to experience' trait.

In the same year, da Silva and Paraboni (2018) presented a study where different features were extracted, including BoW, psycholinguistics, Word2Vec (600-dimensional Continuous BoW and Skipgram embedding models), Doc2Vec, and LSTM (600-dimensional Keras-based embedding layer) to recognize the personality of approximately one thousand Facebook Brazilian users. The highest F1-score reported by the authors was 61% for the 'extraversion' trait. According to them, there is no single model capable of providing the best results for all five traits, which may suggest that not all personality traits are equally modeled from text. They also noticed that word embeddings seem to outperform other models based on lexical resources.

Among the works about language model features, Mehta *et al.* (2020a) used language model embeddings obtained from Bidirectional Encoder Representations from Transformers (BERT) models, as well as psycholinguistic features obtained with Mairesse, SenticNet, NRC Emotion Lexicon, and other methods to predict the personality traits of the Big Five model in the Essays dataset (Pennebaker and King, 1999). In their fine-tuned setup, they experimented with LR, SVM, and Multilayer Perceptron (MLP). Their results showed that language modeling features (based on BERT embeddings) consistently outperformed conventional psycholinguistic features for personality prediction. In another work with the same dataset (Kazameini *et al.*, 2020), the authors used the BERT linguistic model to extract contextualized word embeddings from textual data and psycholinguistic features obtained with Mairesse for automatic author personality detection. Their extensive experiments led them to develop a new model that feeds contextualized embeddings along with psycholinguistic features to a bagged-SVM classifier for personality trait prediction. Their model outperformed previous results in the state of the art by 1,04% while being significantly more efficient. Similarly, Jiang *et al.* (2020) analyzed the Essays dataset using the pre-trained

contextual embeddings obtained from BERT and RoBERTa while also using different linguistic features obtained with LIWC. The authors tested several types of neural networks: HCNN (Hierarchical CNN model), ABCNN, and ABLSTM, which represent CNN and Bidirectional LSTM models with attention mechanism and HAN (Hierarchical Attention Network). They concluded that, in comparison with LIWC-based models and different Neural Networks (HCNN, ABCNN, ABLSTM), their model improved the performance by approximately 2,5% for the five traits on average when 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, the authors achieved the best accuracy in four out of the five traits: 59,7% for 'agreeableness', 60,6% for 'extraversion', 65,9% for 'openness to experience', 61,1% for 'neuroticism', and 60,1% for 'conscientiousness'.

In addition to the studies mentioned above, there are works where transliterations obtained from YouTube videos are considered as the input to the model in order to evaluate different personality traits. Our work is focused on the automatic evaluation of personality traits based on the transliterations provided in [Biel et al. \(2013\)](#) (see the *Data* subsection for further details). The authors that originally introduced the dataset extracted features from the texts using LIWC and TF-IDF features, which were extracted from bi-grams and tri-grams. The authors proposed an approach based on a Random Forest (RF) regressor to predict the label of each trait. The authors report coefficients of determination of 0,04, 0,18, 0,13, 0,31, and 0,17 for each trait in the OCEAN model, respectively. Later, also working with the same dataset, [Sarkar et al. \(2014\)](#) considered each trait as a separate bi-class problem (i.e., they performed the automatic classification of presence vs. absence of each trait). Their model was based on uni-gram BoW and TF-IDF features, and the classification was performed with a 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 [Alam and Riccardi \(2014\)](#), where the best result was obtained with part-of-speech (POS) tagging features, and the classification was performed using an SVM for each separate trait. 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'.

[Das-Kumar and Das-Dipankar \(2017\)](#) considered transliterations from the same dataset and used 69-dimensional LIWC vectors to represent the texts. The authors reported accuracies of up to 62,3% to classify different traits of the OCEAN model. [Sun et al. \(2019\)](#) 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 the OCEAN traits, respectively. More recently, also working with unsupervised methods based on the skip-gram algorithm, [Guan et al. \(2020\)](#) 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, [Salminen et al. \(2020\)](#) considered 300-dimensional embedding vectors obtained from the Google-News Word2Vec pre-trained model. The authors created a neural network architecture that combined convolutional and recurrent layers to perform the classification of the traits.

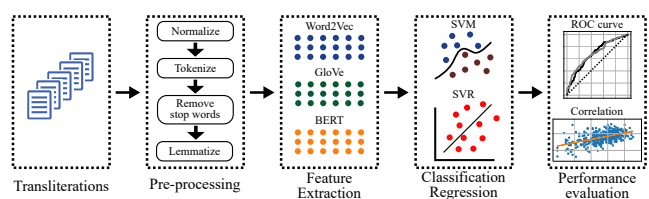
The reported average F1-score value for the OCEAN traits was 54,74%.

Other works have addressed the study of personality considering different biosignals and also merging information in a multimodal approach. For instance, [Mehta et al. \(2020b\)](#) reviewed different machine learning methods according to the input modality, including text, audio, video, and multimodal. For the specific case of modeling personality through text, the authors mentioned that the labels are usually created through questionnaires and surveys. The authors also highlight the importance of data pre-processing to find better models that are more robust and stable. The authors also mention that open vocabulary methods (e.g., Word2Vec, GloVe, BERT) are more robust and have a better generalization capability than others based on a prior judgment of words or categories such as those that rely on lexicons or dictionaries (e.g., LIWC, SenticNet, or NRC Emotion Lexicon). Finally, the authors criticize the use of machine learning methods as the only way to model personality through text, given that those methods highly depend on the data used to train them.

Aiming to make progress in the field of automatic recognition of personality traits, in this study, we focused on extracting information from transliterations of the YouTube database presented in [Biel et al. \(2013\)](#). The proposed approach considers two classical word embeddings, namely Word2Vec and GloVe and state-of-the-art word embeddings obtained from the BERT-base and BERT-large language models. Machine learning systems based on support vector regression and support vector machines are used to estimate and classify the personality traits. Different performance metrics are included in the results. The rest of this paper is organized as follows: first, the methodology and database used in the paper are described; later, the experiments and the discussion of the results are presented; and, in the final section, conclusions and future work are presented.

## Methodology

Figure 1 illustrates the main components of the implemented methodology. Details are included in the next subsections.

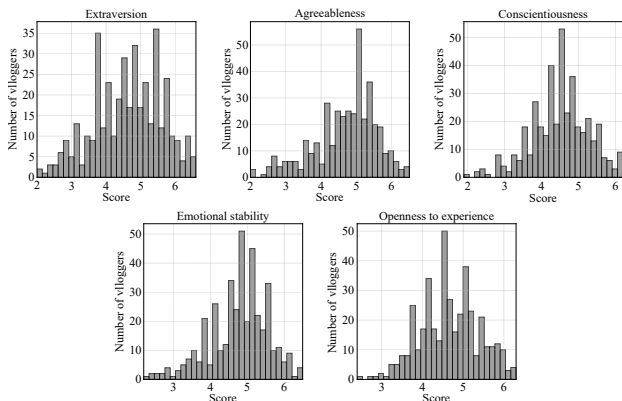


**Figure 1.** Block diagram of the methodology proposed in this study  
Source: Authors

## Data

The 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 are no content-related restrictions in the videos, and the language is natural, diverse, and informal. The transliterations contain approximately 10

000 unique words and 240 000 word tokens. The data is gender-balanced (52% female). The transliterations were originally produced in the 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 (Buhrmester et al., 2016) and the Ten-Item Personality Inventory (Gosling et al., 2003). Figure 2 shows histograms with the scores assigned to each trait. Some statistical information on the scores is also provided in Table 1.



**Figure 2.** Histogram for the score in the five traits  
**Source:** Authors

**Table 1.** Statistical information of the personality scores

Trait	Min.	T1	Med.	T2	Max.	Var.
Extraversion	2	4,2	4,7	5,2	6,6	0,95
Agreeableness	2	4,4	4,9	5,1	6,5	0,77
Conscientiousness	1,9	4,2	4,5	4,8	6,2	0,59
Emotional stability	2,2	4,5	4,8	5,1	6,5	0,61
Openness to experience	2,4	4,4	4,7	5,0	6,3	0,51

Med.: Median; Min.: Minimum; Max.: Maximum; T1: 1st tertile; T2: 2nd tertile; Var.: Variance.

**Source:** Authors

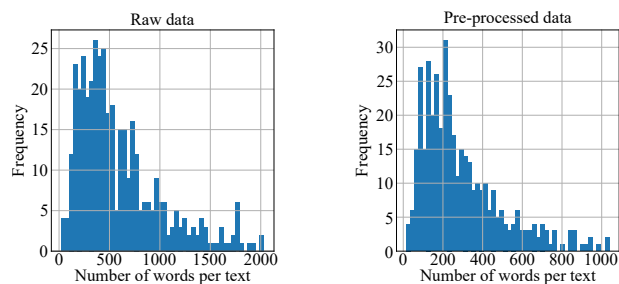
### Pre-processing

Before feature extraction, the data need to be cleaned and standardized in order to remove the ‘noise’ and prepare them for analysis. The steps followed during the text pre-processing include: i) removal of non-content words like ‘xxx’, ‘um’, ‘uh’, and others; ii) conversion of all texts to lower case, removal of punctuation, numbers, and stop-words; and iii) lemmatization, which is applied to transform words into their root form. Figure 3 shows the number of words per text before and after pre-processing.

### Feature extraction

One of the goals of Natural Language Processing (NLP) is to mathematically represent words of a text in a vector space. This vector representation is such that similar words are represented by nearby points. In this work, we consider three different techniques to create said vectors: Word2Vec, Global Vectors (GloVe), and BERT. Details of each model are presented below.

**Word2Vec:** Word2Vec considers information from nearby words to represent target words with a shallow



**Figure 3.** Number of words per text before and after pre-processing  
**Source:** Authors

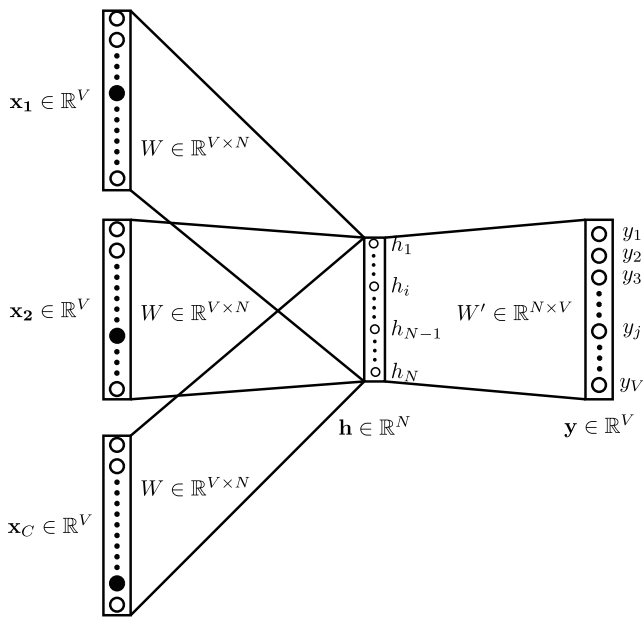
neural network whose hidden layer encodes the word’s representation (Mikolov et al., 2013). The network contains one hidden layer, whose dimension is equal to the embedding size, which is smaller than the one at the input/output vector. A softmax activation function is applied at the output layer, so that each element of the output vector represents the probability of a given word appearing in the context. Word embeddings are obtained from the hidden layer following the Continuous Bag of Words (CBow) method, which considers information from the neighboring words to model the probability of the target word appearing in a given context. In this case, the input is the words around in the given context and the output is the target word. For example, assuming a context of 2 and ‘drink’ as the target word, in the sentence ‘I will drink orange juice’, the input would be ‘I’, ‘will’, ‘orange’, and ‘juice’, while the output would be ‘drink’.

All inputs and output data have the same dimension and are coded as *one-hot* vectors. The length of the vector is equal to the size of the vocabulary in the corpus, which considers unique words. Typically, these unique words are coded in alphabetical order, that is, *one-hot* vectors for words beginning with ‘a’ are expected to have the target ‘1’ in a lower index, while words beginning with ‘z’ are expected to have the target ‘1’ in a higher index. Figure 4 shows the structure of the network for the CBow method, where  $W \in \mathbb{R}^{V \times N}$  refers to the weight matrix that maps inputs  $x_i$  to the hidden layer, and  $W' \in \mathbb{R}^{N \times V}$  is the weight matrix that maps outputs of the hidden layer to the final output layer. The neurons in the hidden layer copy the weighted sum of inputs to the next layer (*i.e.*, linear activation function).

**GloVe:** The Global Vectors for Word Representation (GloVe) model creates word vectors by examining the co-occurrences of words within a corpus (Pennington et al., 2014). Before the model is trained, an  $X$  co-occurrence matrix is created, where each element  $X_{ij}$  represents the frequency for the  $i$ -th word to appear in the context of the  $j$ -th word. The corpus is traversed only once to create the  $X$  matrix, and these co-occurrence data are then used instead of the corpus. 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  $(w_i, w_j)$ :

$$\vec{w}_i^T \vec{w}_j + b_i + b_j = \log(X_{ij}) \tag{1}$$

where  $w_i$  and  $w_j$  are word vectors and  $b_i$  and  $b_j$  are scalar biases associated with the  $i$ -th and  $j$ -th word, respectively.



**Figure 4.** CBoW topology used in Word2Vec.  $x$ : *one-hot* vectors;  $V$ : size of the vocabulary;  $C$ : number of context words;  $h$ : hidden layer with  $N$  neurons, where  $N$  is also the number of dimensions to represent the word;  $y$ : *one-hot* vector for target word. **Source:** Modified from Bellei (2018)

This model creates word vectors with relevant information about how each pair of words coexist. The objective function of the optimization problem is called  $J$ , and it evaluates the mean square error of Equation 1, weighted by the function  $f$ :

$$J = \sum_{i=1}^V \sum_{j=1}^V f(X_{ij})(\vec{w}_i^T \vec{w}_j + b_i + b_j - \log(X_{ij}))^2 \quad (2)$$

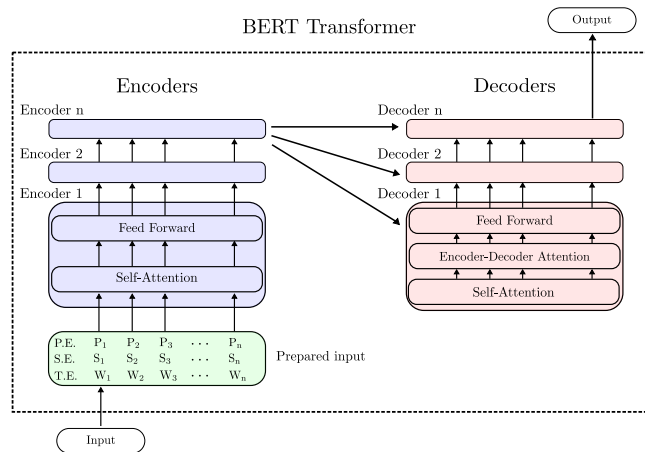
From Equation 2,  $f(X_{ij})$  is chosen in such a way that common word pairs are not considered (those with large  $X_{ij}$  values) because they would deviate too much from regular words:

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_{\max}}\right)^\alpha & \text{if } X_{ij} < x_{\max} \\ 1 & \text{in other case.} \end{cases} \quad (3)$$

In Equation 3,  $x_{\max}$  refers to the maximum co-occurrence value between the  $i$ -th and  $j$ -th word. When common word pairs are found (such that  $X_{ij} > x_{\max}$ ), the function limits its output to 1. For all other word pairs, a weight in the range  $[0 - 1]$  is computed. The distribution of weights depends on  $\alpha$ , which is a hyper-parameter that controls the sensitivity of the weights to increased co-occurrence counts.

**BERT** makes use of transformers, which are attention mechanisms that learn contextual relations between words (or sub-words) in a text (Devlin et al., 2018). 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 influence many language representations that include word predictions (Devlin et al., 2018). To effectively train a bidirectional transformer, BERT uses two techniques called Masked Language Model (MLM) and Next

Sentence Prediction (NSP). The transformer architecture comprises 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, in addition to an encoder-decoder attention layer between the self-attention layer and the feed-forward layer. Figure 5 shows the architecture of the transformer in BERT.



**Figure 5.** Architecture of the transformer used in BERT. **T.E.** text embedding; **S.E.** segment embedding; **P.E.** positional embedding. **Source:** Modified from Alammari (2018)

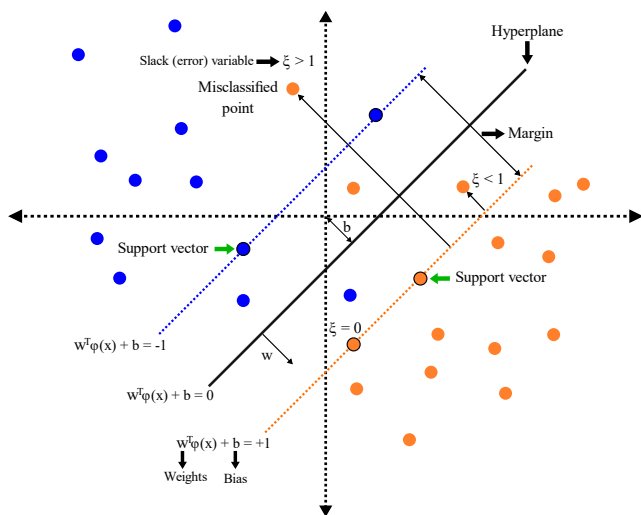
Pre-trained models were used for Word2Vec, GloVe, and BERT. For Word2Vec and GloVe, we considered the Python gensim module (Rehurek and Sojka, 2010). In the case of Word2Vec, the word2vec-google-news-300 model was employed, which was trained with Google News, with a corpus of around 100 billion words and a vector dimension of 300 (Mikolov, 2015). For GloVe, the model glove-wiki-gigaword-300 was employed, which was trained with the Wikipedia 2014 + Gigaword 5 corpus. In this case, there are 5,6 Billion tokens, a vocabulary of 400 000 words, and a vector dimension of 300 as well (Pennington, 2014). To obtain word embeddings based on BERT, we used the WEBERT Toolkit (Pérez, 2020), which is a Python tool typically used to obtain pre-trained BERT embeddings in English and Spanish. To train the BERT word embeddings in the English language, the Multi-Genre Natural Language Inference corpus was used. We considered two different pre-trained BERT models: (i) BERT-base and (ii) BERT-large, where the last layer (768 units for BERT-base and 1 024 units for BERT-large) was taken as the word-embedding representation.

Vector representations for each transcript are created per word, so that  $x \in \mathbb{R}^{300}$  for Word2Vec and GloVe,  $x \in \mathbb{R}^{768}$  for Bert-base, and  $x \in \mathbb{R}^{1024}$  for Bert-large. Since texts correspond to spontaneous speech, they have a different number of words per individual. To obtain vectors of a fixed dimension per text, six statistics are computed along the vectors: mean, standard deviation, skewness, kurtosis, minimum, and maximum. The resulting feature matrix is given by  $X \in \mathbb{R}^{N \times 1800}$  for Word2Vec and Glove,  $X \in \mathbb{R}^{N \times 4608}$  for BERT-base, and  $X \in \mathbb{R}^{N \times 6144}$  for BERT-large. Where  $N$  is the number of transliterations ( $N = 404$ ).

## Classification and regression models

Since Support Vector Machines (SVM) are one of the most used classification methods in the state of the art, and considering their robustness regarding high-dimensional representation spaces (Schölkopf et al., 2002), we decided to adopt this as our main framework for the classification and regression experiments. The following subsections provide some details of the mathematical background of the methods. However, we recommend that the reader refer to Schölkopf et al. (2002) for a more comprehensive description of the methods.

**Bi-class classification:** In this case, the goal is to discriminate data samples by finding a separating hyperplane that maximizes the margin between classes. Soft-margin SVMs allow errors in the process of finding the optimal hyper-plane. These allowed errors are data samples located on the wrong side of the hyper-plane but within the optimal margin. An example of a soft-margin SVM is shown in Figure 6.



**Figure 6.** Soft-margin SVM  
Source: Modified from Dey (2018)

The decision function of a soft-margin SVM is expressed according to Equation 4, where  $\xi_n$  is a slack variable that penalizes the number of errors allowed in the optimization process;  $y_n \in \{-1, +1\}$  are class labels;  $\phi(\mathbf{x}_n)$  is a kernel function that transforms the feature space  $\mathbf{x}$  into a higher dimensional space, where a linear solution to the problem can be found; and the weight vector  $\mathbf{w}$  and the bias  $b$  define the separating hyperplane.

$$y_n \cdot (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n, \quad n = 1, 2, 3, \dots, N \quad (4)$$

The optimization problem is defined in Equation 5, where the hyper-parameter  $C$  controls the offset between  $\xi_n$  and the margin width. Samples  $\mathbf{x}_n$  that satisfy the condition of equality in Equation 4 are called 'support vectors' ( $\mathbf{x}_m$ ).

$$\begin{aligned} & \underset{\mathbf{w}, b}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \\ & \text{subject to} && y \cdot (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n \\ & && \xi_n \geq 0 \end{aligned} \quad (5)$$

**Tri-class classification:** We adopted the One vs. All approach (OVA, also called One vs. Rest, OvR). This method consists of building one SVM per class, which is trained to distinguish the samples of one class from the samples of all the other classes. The decision is made according to the maximum output among all SVMs (Milgram et al., 2006). This approach requires that each model predict a probability score per class. The max argument of these scores (class index with the highest score) is then used to predict a class.

**Regression:** We implemented a Support Vector Regressor (SVR) to predict the value of each label assigned to the personality traits. An SVR is an extension of an SVM, where, instead of integer-valued labels, real-valued labels are predicted. Particularly, an  $\varepsilon$ -SVR aims to find a linear function  $f(\mathbf{x})$  where only samples outside the  $\varepsilon$ -radius 'tube' are penalized (Smola and Schölkopf, 2004) (Figure 7). The linear regression function  $f(\mathbf{x})$  is represented in Equation 6 as:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (6)$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product and  $b \in \mathbb{R}$ . The resulting optimization problem is written as follows:

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} && y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \varepsilon \\ & && \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \varepsilon \end{aligned} \quad (7)$$

The assumption in Equation 7 is that the function  $f(\mathbf{x})$  exists and that 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  $\xi_i$  and  $\xi_i^*$  to cope with otherwise infeasible constraints of the optimization problem in Equation 7. The resulting optimization problem is as follows (Vapnik, 1995):

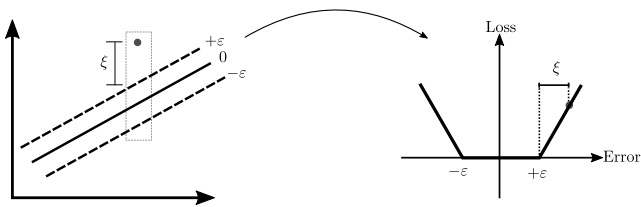
$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ & \text{subject to} && y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \varepsilon + \xi_i \\ & && \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ & && \xi_i, \xi_i^* \geq 0 \end{aligned} \quad (8)$$

In Equation 8, the constant  $C > 0$  determines the trade-off between the flatness of  $f$  and the maximum allowed deviation  $\varepsilon$ . This corresponds to the so called  $\varepsilon$ -insensitive loss function  $|\xi|_\varepsilon$ , which is described in Equation 9:

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \quad (9)$$

Figure 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 SVM to nonlinear functions (Smola and Schölkopf, 2004; Ranković et al., 2014).

**Parameter optimization:** For both SVM and SVR, Gaussian kernel and linear kernel were considered in our preliminary experiments. However since the results with the first one were better in most of the cases, only results with the Gaussian kernel are reported (see Appendix with results for the linear kernel). The hyper-parameters  $C$ ,  $\gamma$ , and  $\varepsilon$  were optimized through a grid-search up to powers of ten between  $1 \times 10^{-4}$  and  $1 \times 10^4$ . A subject-independent



**Figure 7.** Linear SVR with  $\varepsilon$ -insensitive loss function  
**Source:** Modified from Smola and Schölkopf (2004)

10-fold Cross-Validation (CV) strategy was followed in the training process, *i.e.*, the data were divided into 10 groups (randomly chosen, but data from the same subject were never included in the train and test fold simultaneously). The CV strategy was repeated 10 times to evaluate the generalization capability of the model (Kohavi, 1995). The reported metrics correspond to the average and standard deviation of the ten repetitions.

### Performance evaluation

The classification systems used in this work were evaluated with typical performance metrics including the confusion matrix (Powers, 2020). For the particular case of a two-class classification problem, the matrix is as shown in Table 2.

**Table 2.** Confusion matrix

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

**Source:** Authors

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

- **True positive (TP)** refers to the number of samples in class 0 that are correctly classified as class 0.
- **False negative (FN)** corresponds to the number of samples in class 0 that are incorrectly classified as class 1.
- **False positive (FP)** is the number of samples in class 1 that are incorrectly classified as class 0.
- **True negative (TN)** is the number of samples in class 1 that are correctly classified as class 1.

Derived from the aforementioned terms, different performance measures are defined and taken into account, including accuracy (ACC), sensitivity (SEN), specificity (SPE), and the F1-score (F1). Apart from the aforementioned measures, the receiver operating characteristic curves (ROC) were used as a graphical representation that summarizes the performance of binary-classification systems. The performance of the multi-class classification systems was evaluated with the Unweighted Average Recall (UAR) and Cohen's kappa coefficient ( $\kappa$ ).

The regression systems of this study were evaluated according to Spearman's correlation coefficient ( $\rho$ ).

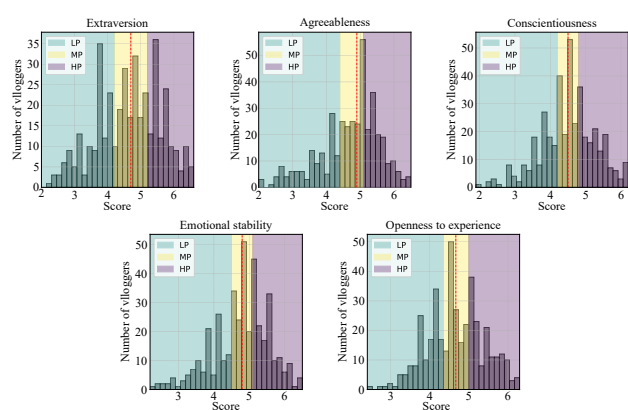
Additionally, to allow comparisons with different works in the literature that report results with different measures, we decided to include other metrics such as the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), and the coefficient of determination,  $R^2$ .

## Experiments, results, and discussion

This section presents the three main experiments performed in this work: i) personality trait estimation according to the scores of the OCEAN model; ii) classification between weak presence vs. strong presence of each personality trait; and iii) classification of three levels in the manifestation of each personality trait (low, medium, and high). Before providing details of each experiment, the next subsection presents the details of the strategy followed to distribute the data, together with statistical analyses that evaluate the suitability of the proposed approach prior to training the classifiers.

### Data distribution and statistical analyses

The personality trait estimation considers the values assigned to each sample for each trait. Their distribution per trait is presented in Figure 2, and the corresponding statistics are presented in Table 1. 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 weak, while those above are labeled as strong. This distribution criterion allows for a balanced number of samples per trait. The median threshold is shown in Figure 8 as a red dotted line. The distribution of the data for the tri-class classification problem is made according to the tertiles of the distribution of the scores per trait. This strategy guarantees balance among the three resulting subgroups. The distribution of these three subgroups is shown in Figure 8 as the three shadowed regions. The number of samples per class and subgroup (two for the bi-class problem and three for the tri-class problem) is summarized in Table 3.



**Figure 8.** Score thresholds for the bi-class and tri-class classification problems. LP: low presence; MP: medium presence; HP: high presence.

**Source:** Authors

Two statistical tests were performed. The first one was the Kruskal-Wallis test regarding the feature matrices extracted per sample and trait. This test was performed for the two scenarios: weak vs. strong presence of each trait, and the three levels of manifestation of the traits. In all

**Table 3.** Number of subjects for the bi-class and tri-class classification problem

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

Source: Authors

cases, the null hypothesis was rejected with  $p \ll 0,01$ . The second statistical test aimed to evaluate whether the gender of subjects biased the distribution of the extracted features. We performed  $\chi^2$  tests for both the bi-class and the tri-class scenarios. A possible bias regarding gender was discarded for the 'extraversion', 'conscientiousness', 'emotional stability', and 'openness to experience' traits, whereas a possible bias was found for the 'agreeableness' trait.

*Experiment 1: personality trait estimation*

This experiment was mainly based on SVR systems with Gaussian kernel. Experiments with linear kernel were also performed, and the results can be found in the Appendix. As mentioned before, to allow comparisons with respect to other works in the literature that use the same corpus as this study, the results are reported in Table 4 regarding the four metrics mentioned in the performance evaluation subsection, where the results in bold letters correspond to the best for each trait. Note that, in three out of the five traits, the best result was obtained when merging the Word2Vec and GloVe embeddings, except for the 'extraversion' trait, whose best result was obtained with the BERT-base embeddings, and 'openness to experience' whose best result was obtained with GloVe embeddings. When observing  $R^2$  and  $\rho$ , the best result was obtained for the 'agreeableness' trait ( $R^2 = 0,24$  and  $\rho = 0,43$ ), followed by 'conscientiousness' ( $R^2 = 0,16$  and  $\rho = 0,41$ ). The worst result was for the 'openness to experience' trait, with  $R^2 = 0,05$  and  $\rho = 0,21$ . The lowest MAE = 0,55 was obtained with 'conscientiousness', and the lowest RMSE value was obtained with 'openness to experience', with a value of RMSE = 0,70.

*Experiment 2: weak vs. strong presence of each trait*

In this case, SVM classifiers with Gaussian kernel were used, and the results are reported in Table 5, where the results in bold letters correspond to the best for each trait. Note that four out of the five traits (all except 'openness to experience') obtained the best result when considering word embeddings based on BERT, three of them with the base model ('extraversion', 'agreeableness', and 'emotional stability') and the remaining one ('conscientiousness') with the large model. Furthermore, note that three out of the five traits exhibit accuracies above 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 seen a few lines below, these results are similar to most of the results reported in

**Table 4.** Results for personality trait estimation

Trait	Feature	$C, \gamma, \epsilon$	$R^2$	$\rho$	MAE	RMSE
Extr	Word2Vec	1e1, 1e-3, 1e-4	0,10 ± 0,00	0,30 ± 0,01	0,77 ± 0,00	0,93 ± 0,00
	GloVe	1, 1e-4, 1e-4	0,10 ± 0,00	0,31 ± 0,02	0,77 ± 0,01	0,93 ± 0,01
	Fusion	1, 1e-4, 1e-4	0,12 ± 0,00	0,35 ± 0,01	0,75 ± 0,01	0,92 ± 0,01
	<b>BERT-b</b>	<b>1, 1e-4, 1e-1</b>	<b>0,14 ± 0,00</b>	<b>0,37 ± 0,02</b>	<b>0,74 ± 0,00</b>	<b>0,91 ± 0,01</b>
	BERT-l	1, 1e-4, 1e-1	0,13 ± 0,00	0,36 ± 0,01	0,74 ± 0,01	0,91 ± 0,01
Agr	Word2Vec	1, 1e-4, 1e-1	0,24 ± 0,00	0,43 ± 0,01	0,61 ± 0,00	0,79 ± 0,00
	GloVe	1, 1e-4, 1e-1	0,16 ± 0,00	0,38 ± 0,02	0,63 ± 0,01	0,81 ± 0,01
	<b>Fusion</b>	<b>1, 1e-4, 1e-1</b>	<b>0,24 ± 0,00</b>	<b>0,43 ± 0,02</b>	<b>0,60 ± 0,00</b>	<b>0,77 ± 0,01</b>
	BERT-b	1e1, 1e-4, 1e-1	0,22 ± 0,00	0,44 ± 0,01	0,62 ± 0,00	0,78 ± 0,01
	BERT-l	1e1, 1e-4, 1e-1	0,19 ± 0,00	0,39 ± 0,01	0,63 ± 0,01	0,79 ± 0,01
Cons	Word2Vec	1, 1e-4, 1e-1	0,16 ± 0,00	0,40 ± 0,01	0,55 ± 0,00	0,71 ± 0,00
	GloVe	1, 1e-4, 1e-1	0,13 ± 0,00	0,37 ± 0,02	0,56 ± 0,01	0,72 ± 0,01
	<b>Fusion</b>	<b>1, 1e-4, 1e-1</b>	<b>0,16 ± 0,00</b>	<b>0,41 ± 0,01</b>	<b>0,55 ± 0,00</b>	<b>0,71 ± 0,00</b>
	BERT-b	1, 1e-4, 1e-1	0,15 ± 0,00	0,40 ± 0,01	0,55 ± 0,00	0,71 ± 0,00
	BERT-l	1, 1e-4, 1e-1	0,15 ± 0,00	0,40 ± 0,01	0,55 ± 0,00	0,71 ± 0,01
Emot	Word2Vec	1, 1e-4, 1e-1	0,05 ± 0,00	0,18 ± 0,02	0,60 ± 0,01	0,76 ± 0,01
	GloVe	1e1, 1e-3, 1e-4	0,07 ± 0,00	0,22 ± 0,02	0,59 ± 0,00	0,75 ± 0,00
	<b>Fusion</b>	<b>1, 1e-4, 1e-1</b>	<b>0,08 ± 0,00</b>	<b>0,24 ± 0,02</b>	<b>0,59 ± 0,01</b>	<b>0,75 ± 0,01</b>
	BERT-b	1e1, 1e-4, 1e-1	0,06 ± 0,00	0,21 ± 0,02	0,59 ± 0,01	0,76 ± 0,01
	BERT-l	1e1, 1e-4, 1e-1	0,06 ± 0,00	0,22 ± 0,02	0,60 ± 0,03	0,76 ± 0,00
Open	Word2Vec	1, 1e-3, 1e-4	0,04 ± 0,00	0,18 ± 0,02	0,57 ± 0,00	0,70 ± 0,00
	<b>GloVe</b>	<b>1, 1e-3, 1e-4</b>	<b>0,05 ± 0,00</b>	<b>0,21 ± 0,02</b>	<b>0,56 ± 0,00</b>	<b>0,70 ± 0,00</b>
	Fusion	1, 1e-4, 1e-1	0,02 ± 0,00	0,16 ± 0,02	0,57 ± 0,00	0,71 ± 0,00
	BERT-b	1, 1e-4, 1e-4	0,00 ± 0,00	0,06 ± 0,04	0,58 ± 0,01	0,72 ± 0,01
	BERT-l	1, 1e-4, 1e-4	0,00 ± 0,00	0,04 ± 0,02	0,58 ± 0,00	0,72 ± 0,00

Fusion: Word2Vec + GloVe; BERT-b: BERT base; BERT-l: BERT large.

Source: Authors

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 be done 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 both humans and machines.

**Table 5.** Results for bi-class system: weak presence vs. strong presence of the trait

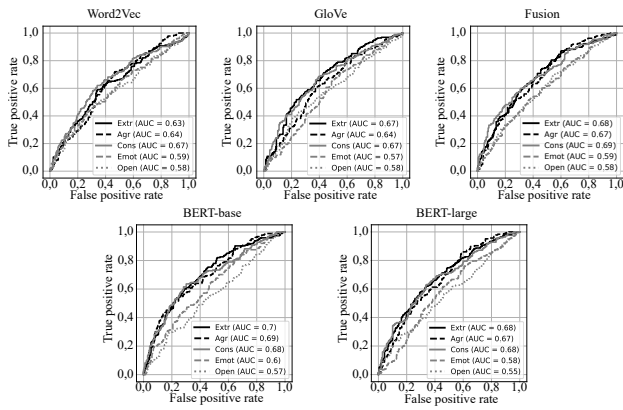
Trait	Feature	$C, \gamma$	Accuracy	Sensitivity	Specificity	F1-score	AUC
Extr	Word2Vec	1e1, 1e-3	60,9 ± 0,8	53,2 ± 1,7	68,1 ± 1,6	60,7 ± 0,9	0,63 ± 0,01
	GloVe	1e1, 1e-3	63,8 ± 1,2	54,7 ± 1,0	72,3 ± 2,1	63,5 ± 1,2	0,67 ± 0,01
	Fusion	1e1, 1e-4	63,4 ± 1,1	62,9 ± 1,6	64,7 ± 1,8	63,4 ± 1,1	0,68 ± 0,01
	<b>BERT-b</b>	<b>1e1, 1e-4</b>	<b>64,7 ± 0,6</b>	<b>63,5 ± 0,9</b>	<b>65,8 ± 1,6</b>	<b>64,7 ± 0,6</b>	<b>0,70 ± 0,01</b>
	BERT-l	1e1, 1e-4	63,4 ± 1,0	62,3 ± 1,3	64,4 ± 1,8	63,4 ± 1,0	0,68 ± 0,01
Agr	Word2Vec	1e1, 1e-4	59,8 ± 1,4	53,3 ± 3,3	65,2 ± 1,9	59,6 ± 1,4	0,64 ± 0,01
	GloVe	1e1, 1e-4	60,3 ± 1,5	52,2 ± 2,3	67,2 ± 2,8	60,0 ± 1,5	0,64 ± 0,01
	Fusion	1e1, 1e-4	60,9 ± 1,6	56,7 ± 2,7	64,5 ± 2,4	60,8 ± 1,6	0,67 ± 0,02
	<b>BERT-b</b>	<b>1e1, 1e-4</b>	<b>64,3 ± 0,8</b>	<b>59,4 ± 1,7</b>	<b>68,5 ± 1,5</b>	<b>64,2 ± 0,8</b>	<b>0,69 ± 0,08</b>
	BERT-l	1e1, 1e-4	61,7 ± 1,2	57,3 ± 2,2	65,4 ± 1,9	61,6 ± 1,2	0,67 ± 0,01
Cons	Word2Vec	1e1, 1e-3	62,5 ± 0,8	53,6 ± 1,2	70,8 ± 1,6	62,2 ± 0,8	0,67 ± 0,01
	GloVe	1e1, 1e-3	63,4 ± 0,7	57,9 ± 1,2	68,6 ± 1,3	63,3 ± 0,7	0,67 ± 0,01
	Fusion	1, 1e-4	63,0 ± 1,1	66,5 ± 1,7	59,8 ± 1,4	62,9 ± 1,1	0,69 ± 0,01
	BERT-b	1, 1e-4	63,6 ± 1,7	64,4 ± 1,8	62,9 ± 1,8	63,6 ± 1,7	0,68 ± 0,01
	<b>BERT-l</b>	<b>1, 1e-4</b>	<b>63,9 ± 1,0</b>	<b>63,9 ± 1,4</b>	<b>63,8 ± 1,6</b>	<b>63,9 ± 1,0</b>	<b>0,68 ± 0,01</b>
Emot	Word2Vec	1, 1e-4	56,7 ± 1,9	52,4 ± 2,9	60,9 ± 3,5	56,6 ± 1,9	0,59 ± 0,02
	GloVe	1, 1e-3	55,5 ± 1,2	53,8 ± 1,2	57,1 ± 2,1	55,5 ± 1,2	0,57 ± 0,02
	Fusion	1, 1e-4	55,9 ± 1,1	54,2 ± 2,1	57,6 ± 1,8	55,9 ± 1,1	0,59 ± 0,02
	<b>BERT-b</b>	<b>1e1, 1e-4</b>	<b>56,8 ± 1,0</b>	<b>54,0 ± 1,4</b>	<b>59,6 ± 1,6</b>	<b>56,8 ± 1,0</b>	<b>0,60 ± 0,01</b>
	BERT-l	1e1, 1e-4	56,5 ± 1,3	54,5 ± 2,7	58,4 ± 2,9	56,4 ± 1,3	0,58 ± 0,01
Open	Word2Vec	1, 1e-3	56,4 ± 1,9	51,7 ± 2,9	60,9 ± 1,9	56,3 ± 1,9	0,58 ± 0,02
	GloVe	1e1, 1e-3	56,4 ± 1,2	52,8 ± 2,3	59,9 ± 2,1	56,3 ± 1,3	0,58 ± 0,02
	<b>Fusion</b>	<b>1e1, 1e-4</b>	<b>56,5 ± 1,5</b>	<b>49,9 ± 2,5</b>	<b>63,0 ± 3,3</b>	<b>56,3 ± 1,5</b>	<b>0,58 ± 0,02</b>
	BERT-b	1e1, 1e-4	55,2 ± 1,2	48,5 ± 2,4	61,9 ± 3,8	55,0 ± 1,2	0,57 ± 0,01
	BERT-l	1, 1e-4	54,0 ± 1,6	52,2 ± 4,2	55,9 ± 2,1	54,0 ± 1,6	0,55 ± 0,01

Fusion: Word2Vec + GloVe; BERT-b: BERT base; BERT-l: BERT large.

Source: Authors

Results are shown more compactly in Figure 9, where the ROC curves resulting from the bi-class experiments are included. Each panel in the figure includes the results obtained with the three feature extraction approaches. The AUC values show that, in the majority of the cases, better results are obtained for 'extraversion' and 'conscientiousness', and also that the best AUC values are obtained by the Fusion of Word2Vec and GloVe embeddings, as well as by embeddings based on BERT.





**Figure 9.** ROC curves obtained with Word2Vec, GloVe, Fusion (Word2Vec + GloVe), BERT-base and BERT-large embeddings  
**Source:** Authors

*Experiment 3: classification of personality traits into three levels*

Three groups were created according to the scores in the personality traits, as it was explained in the data distribution and statistical analyses subsection: LP (low presence), MP (medium presence), and HP (high presence). The results of the tri-class classification are presented in Table 6 in terms of accuracy, F1-score, UAR, and  $\kappa$ , where the results in bold letters correspond to the best for each trait. Note that, in four out of the five traits, the best results were obtained with the Fusion of Word2Vec and GloVe embeddings. Only in the case of ‘conscientiousness’ was the best result obtained with Word2Vec. It can also be observed that the models trained with BERT embeddings did not improve the performance of the models in comparison with the classical word embeddings (Word2Vec and GloVe) in any of the OCEAN traits. However, with 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%.

To facilitate a detailed look at the results, confusion matrices are presented in Table 7, where, again, the results in bold letters correspond to the best for each trait. The results do not show a clear pattern when comparing the three trait levels. 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. Even though the models presented in this study could be improved, while also acknowledging that the addressed problem is very challenging, the behavior of the observed results may reflect possible labeling problems. We believe that there is a big gap in the study of personality traits based on linguistic patterns, which makes it necessary to work on collecting and labeling data considering the knowledge of expert psychologists and psycholinguists (Sun *et al.*, 2019; Guan *et al.*, 2020).

The best results obtained throughout the regression and classification experiments are summarized in Figure 10. The first column of sub-figures shows the regression results. It can be observed that the regressor is doing a good job, especially in the first three traits, where Spearman’s correlation is above 0,35. In the second and third columns,

**Table 6.** Tri-class classification results

Trait	Feature	Accuracy	F1-score	UAR	$\kappa$
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
	<b>Fusion</b>	<b>44,3 ± 1,2</b>	<b>44,1 ± 1,2</b>	<b>44,7 ± 1,3</b>	<b>0,17 ± 0,02</b>
	BERT-b	41,8 ± 1,1	41,7 ± 1,1	41,8 ± 1,1	0,12 ± 0,02
	BERT-l	41,9 ± 1,0	41,8 ± 1,0	41,8 ± 1,0	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
	<b>Fusion</b>	<b>46,2 ± 1,3</b>	<b>46,3 ± 1,3</b>	<b>46,2 ± 1,3</b>	<b>0,19 ± 0,02</b>
	BERT-b	45,9 ± 0,9	45,8 ± 0,82	45,9 ± 0,8	0,19 ± 0,01
	BERT-l	44,7 ± 1,3	44,5 ± 1,2	44,6 ± 1,3	0,17 ± 0,02
Cons	<b>Word2Vec</b>	<b>46,6 ± 1,2</b>	<b>45,1 ± 1,2</b>	<b>45,8 ± 1,2</b>	<b>0,19 ± 0,02</b>
	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-b	45,5 ± 1,5	45,3 ± 1,5	45,1 ± 1,5	0,18 ± 0,02
	BERT-l	43,6 ± 0,8	43,5 ± 0,8	43,3 ± 0,8	0,15 ± 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
	<b>Fusion</b>	<b>40,4 ± 1,3</b>	<b>40,4 ± 1,3</b>	<b>40,5 ± 1,3</b>	<b>0,11 ± 0,02</b>
	BERT-b	38,3 ± 0,7	38,0 ± 0,6	38,2 ± 0,7	0,07 ± 0,01
	BERT-l	34,7 ± 1,6	34,4 ± 1,5	34,6 ± 1,5	0,02 ± 0,02
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
	<b>Fusion</b>	<b>41,2 ± 0,8</b>	<b>41,0 ± 0,9</b>	<b>40,8 ± 0,8</b>	<b>0,11 ± 0,01</b>
	BERT-b	35,0 ± 2,1	34,2 ± 2,1	34,3 ± 2,0	0,01 ± 0,03
	BERT-l	34,6 ± 1,4	33,9 ± 1,5	34,0 ± 1,4	0,01 ± 0,02

**Fusion:** Word2Vec + GloVe; **BERT-b:** BERT base; **BERT-l:** BERT large.

**Source:** Authors

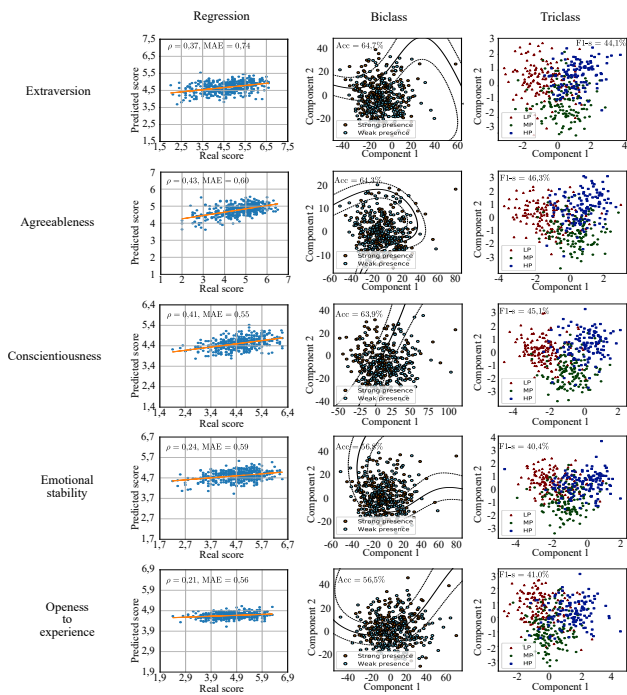
**Table 7.** Confusion matrix for the classification of personality traits into three levels (results in %)

		Extr			Agr			Cons			Emot			Open		
		LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP	LP	MP	HP
Word2Vec	LP	58	31	11	56	26	18	<b>66</b>	21	13	47	32	21	36	45	19
	MP	46	32	22	25	45	30	51	<b>26</b>	23	38	37	25	35	47	18
	HP	36	28	36	23	40	37	34	20	<b>46</b>	32	34	34	32	38	30
GloVe	LP	60	28	12	53	25	22	70	18	12	42	32	26	44	37	19
	MP	47	33	20	27	44	29	53	24	23	36	34	30	31	48	21
	HP	34	28	38	23	36	41	31	25	44	29	30	41	28	45	27
Fusion	LP	<b>45</b>	33	22	<b>51</b>	26	23	49	35	16	<b>41</b>	33	26	<b>42</b>	36	22
	MP	31	<b>35</b>	34	25	<b>43</b>	32	31	41	28	35	<b>37</b>	28	29	<b>46</b>	25
	HP	19	26	<b>55</b>	19	37	<b>44</b>	18	35	47	27	29	<b>44</b>	30	36	<b>34</b>
BERT-b	LP	49	36	15	55	28	17	56	29	15	47	27	26	41	39	20
	MP	44	32	24	32	38	30	33	34	33	37	34	29	40	41	19
	HP	28	28	44	23	33	44	24	31	45	32	34	34	36	37	27
BERT-l	LP	48	34	18	53	24	23	53	33	14	43	34	23	41	40	19
	MP	39	34	27	33	39	28	36	31	33	36	34	30	42	39	19
	HP	27	30	43	22	36	42	23	31	46	31	42	27	41	37	22

**Fusion:** Word2Vec + GloVe; **BERT-b:** BERT base; **BERT-l:** BERT large.  
 LP: Low presence; MP: Medium presence; HP: High presence.

**Source:** Authors

the resulting representation spaces from the bi-class and tri-class scenarios are shown, respectively. Note that, in these two cases, the Figures illustrate the result of applying a dimensionality reduction based on Principal Component Analysis (PCA). In the bi-class scenario, the hyper-planes shown in the Figure correspond to those found with the optimal parameters of the SVM, *i.e.*, using the parameters reported in Table 5. 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. 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 also representations resulting from the PCA projection of the feature space. Even though the results appear to be low, the representation spaces show that the three sub-groups are found. These results, as well as the summarizing Figures, motivate us to continue working on this topic considering other approaches from fields such as language features and deep learning.



**Figure 10.** Graphical summary of the best results  
**Source:** Authors

**Comparison with recent works**

The results reported in this study were 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 regression and the the bi-class results.

The summary of our regression results and those reported by others in the literature are included in Table 8. According to the average results reported in the last row of the table, our approach shows similar performance to others reported in the literature. Our results are in fact better with regard to the MAE value in about 0,02. Now, in the case of RMSE, our results are 0,03 below (worst) and 0,04 below (worst) with respect to the previous result reported for  $R^2$ . Although other works in the literature did not report results for Spearman’s correlation, we decided to examine it because this measure is more intuitive, especially when the ‘actual vs. predicted’ plot is shown (Figure 10).

**Table 8.** Comparison of our regression model with recent works

Trait	Our approach				Biel <i>et al.</i> (2013)	Sun <i>et al.</i> (2019)	Guan <i>et al.</i> (2020)
	$R^2$	$\rho$	RMSE	MAE	$R^2$	RMSE	MAE
Extr	0,14	0,37	0,91	0,74	0,13	0,89	0,72
Agre	0,24	0,43	0,77	0,60	0,31	0,77	0,67
Cons	0,16	0,41	0,71	0,55	0,18	0,69	0,57
Emot	0,08	0,24	0,75	0,59	0,17	0,69	0,60
Open	0,05	0,21	0,70	0,56	0,04	0,68	0,58
<b>Average</b>	0,13	0,33	0,77	0,61	0,17	0,74	0,63

**Source:** Authors

The comparison regarding the bi-class classification scenario is reported in Table 9. Note that, on average, our results are slightly better than the other works, except for those obtained by Das-Kumar and Das-Dipankar (2017). Unfortunately, the authors of that work only report the average accuracy along with the five traits, which does

not allow for direct comparisons in specific traits. If we consider the average performance in terms of the F1-score compared to the performance reported by Salminen *et al.* (2020), we were able to improve the performance by 6,5%, which is relevant considering that we did not use neural networks. This gives us an idea that, for certain traits (‘agreeableness’, ‘conscientiousness’, and ‘emotional stability’), classical methods such as those used in this paper (SVM with Gaussian kernel) yield better results than those found with other methods like the one reported by Salminen *et al.* (2020).

**Table 9.** Comparison of our bi-class classification model with recent works

Trait	Sarkar <i>et al.</i> (2014)		Alam and Riccardi (2014)		Das-Kumar and Das-Dipankar (2017)		Salminen <i>et al.</i> (2020)	
	Acc	F1-score	F1-score	F1-score	Acc	F1-score	F1-score	
Extr	64,7	64,7	60,5	57,8	-	-	71,9	
Agre	64,3	64,2	65,7	69,6	-	-	44,4	
Cons	63,9	63,9	65,8	54,3	-	-	48,5	
Emot	56,8	56,8	47,7	61,9	-	-	40,3	
Open	56,5	56,3	60,8	57,3	-	-	68,6	
<b>Average</b>	61,2	61,2	60,1	60,2	62,3	-	54,7	

**Source:** Authors

**Conclusions and future work**

This work considered the transliterations of multimedia recordings obtained from YouTube and explored the use of three word embedding methods (Word2Vec, GloVe, and BERT) to model the five different personality traits included in the OCEAN model. Standard regression and classification methods were considered to facilitate the analysis regarding the embedding methods. Three different evaluation scenarios were presented in this work: i) estimation of the personality scores by creating a regression system, ii) automatic classification of the strong vs. weak presence of each personality trait, and iii) the classification of three levels of personality traits. According to our findings in the regression experiments, Spearman’s correlation coefficients ranging from 0,21 to 0,43 were obtained between the predicted personality scores and the ones assigned by the labeling system. Other performance measures such as MAE, RMSE, and  $R^2$  were also reported to allow comparisons with respect to other studies in the state of the art. The classification between strong vs. weak presence of the traits shows that the accuracy and F1-score of the proposed approach are in the same range as those reported in the literature, and, in some traits (‘extraversion’, ‘conscientiousness’, and ‘emotional stability’), our results outperform those of previous works in terms of the F1-score metric. Finally, the classification of three levels of personality traits (low, medium, and high) shows accuracies between 40,4 and 46,6%.

In general terms, our findings suggest that Word2Vec and GloVe embedding methods may be combined to obtain better results, and that the addition of the BERT-base and BERT-large models did not improve the performance in regression or in tri-class classification experiments. However, they considerably improved the performance in the two-class classification experiments with respect to the performance of the models based on Word2Vec, GloVe, or a fusion of both. It can also be concluded from the results that models based on word embeddings obtained with BERT-base generally outperform models based on word embeddings with BERT-large, which is in line with the work

by Mehta *et al.* (2020a), who found that the use of a larger linguistic model does not always result in a better performance. Further research is required to increase the results. Additionally, although the use of the data and labels considered in this work is relatively standard, we believe that there is also a big gap that needs to be filled in the labeling process. We are aware of the fact that these processes are expensive, time-consuming, and require sophisticated knowledge (especially from psychologists); joint efforts are required to create realistic databases labeled with more natural personality scores.

## Acknowledgements

This work was funded by CODI from Universidad de Antioquia, grant no. PRG2017-15530.

## References

- Alam, F. and Riccardi, G. (2014). Predicting personality traits using multimodal information. In ACM (Eds.), *Proceedings of the 2014 ACM multi media on workshop on computational personality recognition* (pp. 15-18). ACM. <https://dl.acm.org/doi/10.1145/2659522.2659531>
- Alammar, J. (June 27, 2018). The illustrated transformer. *Jay Alammar*. <http://jalammar.github.io/illustrated-transformer/>
- Allport, G. W. (1937). *Personality: A psychological interpretation*. Holt.
- Bellei, C. (2018). The backpropagation algorithm for Word2Vec. *Marginalia* <http://www.claudiobellei.com/2018/01/06/backprop-word2vec/>
- Biel, J. I., Tsiminaki, V., Dines, J., and Gatica-Perez, D. (2013). Hi YouTube! Personality impressions and verbal content in social video. In ACM (Eds.) *Proceedings of the 15th ACM on International conference on multimodal interaction* (pp. 119-126). ACM. <https://doi.org/10.1145/2522848.2522877>
- Buhrmester, M., Kwang, T., and Gosling, S. D. (2016). Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality data? In A. E. Kazdin (Ed.), *Methodological issues and strategies in clinical research* (pp. 133-139). American Psychological Association. <https://doi.org/10.1037/14805-009>
- Cambria, E., Das, D., Bandyopadhyay, S., and Feraco, A. (2017). Affective computing and sentiment analysis. In E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco (Eds.), *A practical guide to sentiment analysis* (pp. 1-10). Springer. <https://doi.org/10.1007/978-3-319-55394-8>
- Celli, F. (2012). Unsupervised personality recognition for social network sites. In ICDS (Eds.), *ICDS 2012: The Sixth International Conference on Digital Society* (pp. 59-62). IARIA. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.723.5551&rep=rep1&type=pdf>
- Celli, F., Lepri, B., Biel, J. I., Gatica-Perez, D., Riccardi, G., and Pianesi, F. (2014). The workshop on computational personality recognition 2014. In ACM (Eds.), *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 1245-1246). ACM. <https://doi.org/10.1145/2647868.2647870>
- da Silva, B. B. C. and Paraboni, I. (2018). Personality recognition from Facebook text. In A. Villavicencio, V. Moreira, A. Abad, H. Caseli, P. Gamallo, C. Ramisch, H. G. Oliveira, and G. H. Paetzold (Eds.) *International Conference on Computational Processing of the Portuguese Language* (pp. 107-114). Springer. [https://doi.org/10.1007/978-3-319-99722-31\\_1](https://doi.org/10.1007/978-3-319-99722-31_1)
- Das, K. G., and Das, D. (2017, December). *Developing lexicon and classifier for personality identification in texts* [Conference presentation]. 14th International Conference on Natural Language Processing (ICON-2017), Kolkata, India. <https://aclanthology.org/W17-7545.pdf>
- Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*. <https://arxiv.org/abs/1810.04805>
- Dey, S. (2018, April). Implementing a soft-margin kernelized support vector machine binary classifier with quadratic programming in R and Python. *Simple Data Science*. <https://sandipanweb.wordpress.com/2018/04/23/>
- Gosling, S. D., Rentfrow, P. J., and Swann Jr., W. B. (2003). A very brief measure of the Big-Five personality domains. *Journal of Research in Personality*, 37(6), 504-528. [https://doi.org/10.1016/S0092-6566\(03\)00046-1](https://doi.org/10.1016/S0092-6566(03)00046-1)
- Guan, Z., Wu, B., Wang, B., and Liu, H. (2020). Personality2vec: network representation learning for personality. In IEEE (Eds.) *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)* (pp. 30-37). IEEE. <https://doi.org/10.1109/DSC50466.2020.00013>
- Hassanein, M., Hussein, W., Rady, S., and Gharib, T. F. (2018). Predicting personality traits from social media using text semantics. In IEEE (Eds.) *2018 13th International Conference on Computer Engineering and Systems (ICCES)* (pp. 184-189). IEEE. <https://doi.org/10.1109/ICCES.2018.8639408>
- Jiang, H., Zhang, X., and Choi, J. D. (2020). Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (Student Abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10), 13821-13822. <https://doi.org/10.1609/aaai.v34i10.7182>
- John, O. P., Donahue, E. M., and Kentle, R. L. (1991). Big Five inventory (BFI) [Database record]. *APA PsycTests*. <https://doi.org/10.1037/t07550-000>
- John, O. P., Naumann, L. P., and Soto, C. J. (2008). Paradigm shift to the integrative Big Five trait taxonomy: history, measurement, and conceptual issues. In O. P. John, R. W. Robins, and L. A. Pervin (Eds.), *Handbook of Personality: Theory and Research* (pp. 114-158). The Guilford Press.
- Kazameini, A., Fatehi, S., Mehta, Y., Eetemadi, S., and Cambria, E. (2020). Personality trait detection using bagged svm over bert word embedding ensembles. *arXiv preprint*. <https://arxiv.org/abs/2010.01309>
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 14(2), 1137-1145. <https://dl.acm.org/doi/10.5555/1643031.1643047>

- Kosinski, M., Matz, S. C., Gosling, S. D., Popov, V., and Stillwell, D. (2015). Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. *American Psychologist*, 70(6), 543. <https://doi.org/10.1037/a0039210>
- Mao, Y., Zhang, D., Wu, C., Zheng, K., and Wang, X. (2018). Feature analysis and optimisation for computational personality recognition. In IEEE (Eds.), *2018 IEEE 4th International Conference on Computer and Communications (ICCC)* (pp. 2410-2414). IEEE. <https://doi.org/10.1109/CompComm.2018.8780801>
- Mehta, Y., Fatehi, S., Kazameini, A., Stachl, C., Cambria, E., and Eetemadi, S. (2020a). Bottom-up and top-down: Predicting personality with psycholinguistic and language model features. In IEEE (Eds.) *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 1184-1189). IEEE. <https://doi.org/10.1109/ICDM50108.2020.00146>
- Mehta, Y., Majumder, N., Gelbukh, A., and Cambria, E. (2020b). Recent trends in deep learning based personality detection. *Artificial Intelligence Review*, 53(4), 2313-2339. <https://doi.org/10.1007/s10462-019-09770-z>
- Milgram, J., Cheriet, M., and Sabourin, R. (2006, October). "One against one" or "one against all": Which one is better for handwriting recognition with SVMs? [Conference presentation]. Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule, France. <https://hal.inria.fr/inria-00103955>
- Mikolov, T. (2015). word2vec: Tool for computing continuous distributed representations of words. *Google Code*. <https://code.google.com/archive/p/word2vec/>
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint*. <https://arxiv.org/abs/1301.3781>.
- Mohammad, S. and Kiritchenko, S. (2013). Using nuances of emotion to identify personality. *Seven International AAAI Conference on Web and Social Media*. <https://doi.org/10.48550/arXiv.1309.6352>
- Onan, A. (2015). Classifier and feature set ensembles for web page classification. *Journal of Information Science*, 42(2), 150-165. <https://doi.org/10.1177/0165551515591724>
- Onan, A. (2016). An ensemble scheme based on language function analysis and feature engineering for text genre classification. *Journal of Information Science*, 44(1), 1-20. <https://doi.org/10.1177/0165551516677911>
- Onan, A. (2017a). Hybrid supervised clustering based ensemble scheme for text classification. *Kybernetes*, 46(2), 330-348. <https://doi.org/10.1108/K-10-2016-0300>
- Onan, A. (2017b). A K-medoids based clustering scheme with an application to document clustering. In IEEE (Eds.), *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 354-359). IEEE. <https://doi.org/10.1109/UBMK.2017.8093409>
- Onan, A. (2018). Sentiment analysis on Twitter based on ensemble of psychological and linguistic feature sets. *Balkan Journal of Electrical and Computer Engineering*, 6(2), 69-77. <https://doi.org/10.17694/bajece.419538>
- Onan, A. (2019a). Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. *IEEE Access*, 7, 145614-145633. <https://doi.org/10.1109/ACCESS.2019.2945911>
- Onan, A. (2019b). Mining opinions from instructor evaluation reviews: a deep learning approach. *Computer Applications in Engineering Education*, 28(1), 117-138. <https://doi.org/10.1002/cae.22179>
- Onan, A. (2020). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience*, e5909. <https://doi.org/10.1002/cpe.5909>
- Onan, A., and Korukoglu, S. (2015). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, 43(1), 25-38. <https://doi.org/10.1177/0165551515613226>
- Onan, A., Korukoğlu, S., and Bulut, H. (2016a). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57, 232-247. <https://doi.org/10.1016/j.eswa.2016.03.045>
- Onan, A., Korukoğlu, S., and Bulut, H. (2016b). LDA-based topic modelling in text sentiment classification: An Empirical Analysis. *International Journal of Computational Linguistics and Applications*, 7(1), 101-119. <https://doi.org/10.1016/j.eswa.2016.06.005>
- Onan, A., Korukoğlu, S., and Bulut, H. (2016c). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications*, 62, 1-16. <https://doi.org/10.1016/j.eswa.2016.06.005>
- Pennebaker, J. W. and King, L. A. (1999). Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6), 1296-1312. <https://doi.org/10.1037/0022-3514.77.6.1296>
- Pennington, J. (2014). GloVe: Global Vectors for Word Representation. <https://nlp.stanford.edu/projects/glove/>
- Pennington, J., Socher, R., and Manning, C. D. (2014, October). Glove: Global vectors for word representation [Conference presentation]. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. <http://dx.doi.org/10.3115/v1/D14-1>
- Pérez, P. A. (2020). WEBERT: Word Embeddings using BERT. <https://doi.org/10.5281/zenodo.3964244>
- Powers, D. M. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint*. <https://arxiv.org/abs/2010.16061>
- Pratama, B. Y. and Sarno, R. (2015). Personality classification based on Twitter text using Naive Bayes, KNN and SVM. In IEEE (Eds.) *2015 International Conference on Data and Software Engineering (ICoDSE)* (pp. 170-174). IEEE. <https://doi.org/10.1109/ICoDSE.2015.7436992>
- Ranković, V., Grujović, N., Divac, D., and Milivojević, N. (2014). Development of support vector regression identification model for prediction of dam structural behaviour. *Structural Safety*, 48, 33-39. <https://doi.org/10.1016/j.strusafe.2014.02.004>
- Rehurek, R. and Sojka, P. (2010, May 2). *Software framework for topic modelling with large corpora* [Conference presentation]. LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta.

Salminen, J., Rao, R. G., Jung, S. G., Chowdhury, S. A., and Jansen, B. J. (2020). Enriching social media personas with personality traits: a deep learning approach using the Big Five classes. In H. Degen, L. Reinerman-Jones (Eds.), *International Conference on Human-Computer Interaction* (pp. 101-120). Springer. [https://doi.org/10.1007/978-3-030-50334-5\\_7](https://doi.org/10.1007/978-3-030-50334-5_7)

Sarkar, C., Bhatia, S., Agarwal, A., and Li, J. (2014, November). Feature analysis for computational personality recognition using youtube personality data set. In ACM (Eds.), *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition* (pp. 11-14). ACM. <https://doi.org/10.1145/2659522.2659528>

Schölkopf, B., Smola, A. J., and Bach, F. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3), 199-222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>

Sun, X., Liu, B., Meng, Q., Cao, J., Luo, J., and Yin, H. (2019). Group-level personality detection based on text generated networks. *World Wide Web*, 23(3), 1887-1906. <https://doi.org/10.1007/s11280-019-00729-2>

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer. <https://doi.org/10.1007/978-1-4757-2440-0>

Vinciarelli, A. and Mohammadi, G. (2014). A survey of personality computing. *IEEE Transactions on Affective Computing*, 5(3), 273-291. <https://doi.org/10.1109/TAFFC.2014.2330816>

White, J. K., Hendrick, S. S., and Hendrick, C. (2004). Big Five personality variables and relationship constructs. *Personality and individual differences*, 37(7), 1519-1530. <https://doi.org/10.1016/j.paid.2004.02.019>

Xue, D., Hong, Z., Guo, S., Gao, L., Wu, L., Zheng, J., and Zhao, N. (2017). Personality recognition on social media with label distribution learning. *IEEE Access*, 5, 13478-13488. <https://doi.org/10.1109/ACCESS.2017.2719018>

## Appendix: results obtained with linear kernels

This section shows the results obtained in the three main experiments of this work regarding linear kernels.

**Table 10.** Results for personality trait estimation with linear kernel

Trait	Feature	C, ε	R <sup>2</sup>	ρ	MAE	RMSE
Extr	Word2Vec	1e-4, 1e-1	0,08 ± 0,00	0,29 ± 0,01	0,77 ± 0,00	0,94 ± 0,00
	GloVe	1e-4, 1e-1	0,10 ± 0,00	0,31 ± 0,01	0,76 ± 0,00	0,93 ± 0,00
	Fusion	1e-4, 1e-4	0,11 ± 0,00	0,33 ± 0,00	0,76 ± 0,00	0,92 ± 0,00
	<b>BERT-b</b>	<b>1e-4, 1e-1</b>	<b>0,14 ± 0,00</b>	<b>0,37 ± 0,01</b>	<b>0,74 ± 0,01</b>	<b>0,91 ± 0,01</b>
	BERT-l	1e-4, 1e-1	0,09 ± 0,00	0,30 ± 0,01	0,76 ± 0,00	0,94 ± 0,00
Agr	Word2Vec	1e-4, 1e-2	0,18 ± 0,00	0,41 ± 0,03	0,62 ± 0,01	0,80 ± 0,01
	GloVe	1e-4, 1e-1	0,14 ± 0,00	0,36 ± 0,02	0,64 ± 0,01	0,83 ± 0,01
	<b>Fusion</b>	<b>1e-4, 1e-1</b>	<b>0,26 ± 0,00</b>	<b>0,45 ± 0,01</b>	<b>0,59 ± 0,00</b>	<b>0,77 ± 0,00</b>
	BERT-b	1e-4, 1e-4	0,21 ± 0,00	0,43 ± 0,01	0,62 ± 0,01	0,79 ± 0,01
	BERT-l	1e-4, 1e-1	0,17 ± 0,00	0,38 ± 0,01	0,63 ± 0,00	0,81 ± 0,01
Cons	Word2Vec	1e-4, 1e-1	0,14 ± 0,00	0,39 ± 0,01	0,55 ± 0,00	0,71 ± 0,00
	GloVe	1e-4, 1e-1	0,14 ± 0,00	0,38 ± 0,01	0,56 ± 0,00	0,72 ± 0,00
	<b>Fusion</b>	<b>1e-4, 1e-1</b>	<b>0,15 ± 0,00</b>	<b>0,41 ± 0,01</b>	<b>0,55 ± 0,00</b>	<b>0,71 ± 0,00</b>
	BERT-b	1e-4, 1e-1	0,14 ± 0,00	0,38 ± 0,01	0,56 ± 0,01	0,72 ± 0,01
	BERT-l	1e-4, 1e-1	0,14 ± 0,00	0,39 ± 0,01	0,55 ± 0,01	0,71 ± 0,01
Emot	Word2Vec	1e-3, 1	0,07 ± 0,00	0,19 ± 0,03	0,60 ± 0,00	0,75 ± 0,01
	GloVe	1e-4, 1e-1	0,04 ± 0,00	0,16 ± 0,05	0,60 ± 0,01	0,77 ± 0,01
	Fusion	1e-4, 1	0,06 ± 0,00	0,19 ± 0,02	0,59 ± 0,01	0,76 ± 0,01
	<b>BERT-b</b>	<b>1e-4, 1e-1</b>	<b>0,07 ± 0,00</b>	<b>0,23 ± 0,02</b>	<b>0,58 ± 0,01</b>	<b>0,75 ± 0,01</b>
	BERT-l	1e-4, 1e-1	0,03 ± 0,00	0,16 ± 0,04	0,61 ± 0,01	0,77 ± 0,01
Open	Word2Vec	1e-4, 1e-4	0,01 ± 0,00	0,14 ± 0,01	0,57 ± 0,00	0,71 ± 0,00
	GloVe	<b>1e-4, 1e-4</b>	<b>0,02 ± 0,00</b>	<b>0,19 ± 0,02</b>	<b>0,56 ± 0,00</b>	<b>0,71 ± 0,00</b>
	Fusion	1e-4, 1e-4	0,02 ± 0,00	0,17 ± 0,03	0,57 ± 0,01	0,72 ± 0,00
	BERT-b	1e-4, 1	0,00 ± 0,00	-0,04 ± 0,03	0,59 ± 0,00	0,73 ± 0,01
	BERT-l	1e-4, 1e-1	0,01 ± 0,00	-0,05 ± 0,03	0,59 ± 0,00	0,73 ± 0,01

**Fusion:** Word2Vec + GloVe; **BERT-b:** BERT base; **BERT-l:** BERT large.

Source: Authors

**Table 11.** Results for bi-class system: weak presence vs. strong presence of the trait considering linear kernel

Trait	Feature	C	Accuracy	Sensitivity	Specificity	F1-score	AUC
Extr	Word2Vec	1e-3	60,3 ± 1,7	58,3 ± 1,6	62,2 ± 2,3	60,3 ± 1,6	0,63 ± 0,01
	GloVe	<b>1e-3</b>	<b>61,6 ± 1,3</b>	<b>63,8 ± 1,9</b>	<b>59,4 ± 1,2</b>	<b>61,6 ± 1,3</b>	<b>0,65 ± 0,01</b>
	Fusion	1e-4	61,0 ± 1,1	63,1 ± 1,4	59,0 ± 1,8	61,0 ± 1,1	0,65 ± 0,01
	BERT-b	1e-3	60,8 ± 1,1	62,5 ± 1,8	59,2 ± 1,6	60,8 ± 1,0	0,66 ± 0,01
	BERT-l	1e-4	61,2 ± 1,3	65,6 ± 1,6	57,0 ± 2,0	61,1 ± 1,3	0,67 ± 0,01
Agr	Word2Vec	1e-3	59,4 ± 1,4	59,4 ± 1,4	59,4 ± 1,9	59,4 ± 1,3	0,65 ± 0,01
	GloVe	1e-3	61,6 ± 1,0	60,3 ± 2,3	62,7 ± 1,2	61,6 ± 1,0	0,66 ± 0,01
	Fusion	1e-3	59,0 ± 1,3	55,9 ± 2,5	61,7 ± 1,9	59,0 ± 1,3	0,64 ± 0,01
	<b>BERT-b</b>	<b>1e-3</b>	<b>62,4 ± 1,6</b>	<b>60,3 ± 1,5</b>	<b>64,3 ± 2,5</b>	<b>62,4 ± 1,5</b>	<b>0,67 ± 0,01</b>
	BERT-l	1e-4	60,7 ± 1,8	59,2 ± 3,1	61,9 ± 1,8	60,7 ± 1,8	0,66 ± 0,01
Cons	Word2Vec	1e-4	61,9 ± 1,8	68,6 ± 2,4	55,7 ± 2,2	61,8 ± 1,8	0,66 ± 0,02
	GloVe	1e-4	61,8 ± 1,1	66,4 ± 1,0	57,5 ± 1,9	61,8 ± 1,1	0,66 ± 0,01
	<b>Fusion</b>	<b>1e-4</b>	<b>62,7 ± 0,7</b>	<b>69,1 ± 1,9</b>	<b>56,8 ± 1,2</b>	<b>62,6 ± 0,7</b>	<b>0,69 ± 0,01</b>
	BERT-b	1e-4	61,8 ± 1,0	66,5 ± 1,2	57,5 ± 1,3	61,8 ± 1,0	0,67 ± 0,01
	BERT-l	1e-4	62,2 ± 0,6	66,4 ± 1,2	58,2 ± 1,7	62,1 ± 0,6	0,67 ± 0,1
Emot	Word2Vec	1e-3	55,3 ± 1,4	49,2 ± 3,6	61,3 ± 2,4	55,1 ± 1,5	0,58 ± 0,01
	GloVe	1e-2	50,9 ± 1,9	46,2 ± 2,6	55,7 ± 2,9	50,8 ± 1,9	0,55 ± 0,01
	Fusion	1e-4	54,9 ± 1,5	43,7 ± 2,0	65,9 ± 2,3	54,3 ± 1,5	0,59 ± 0,01
	<b>BERT-b</b>	<b>1e-3</b>	<b>57,6 ± 2,5</b>	<b>56,1 ± 3,4</b>	<b>59,1 ± 2,5</b>	<b>57,6 ± 2,5</b>	<b>0,62 ± 0,02</b>
	BERT-l	1e-4	55,5 ± 1,3	53,1 ± 2,0	57,9 ± 1,3	55,5 ± 1,3	0,58 ± 0,02
Open	Word2Vec	1e-3	54,6 ± 1,5	53,8 ± 3,0	55,5 ± 2,7	54,6 ± 1,5	0,56 ± 0,01
	GloVe	1e-3	53,1 ± 1,5	50,7 ± 2,8	55,6 ± 2,2	53,1 ± 1,5	0,55 ± 0,02
	<b>Fusion</b>	<b>1e-4</b>	<b>57,4 ± 1,9</b>	<b>53,5 ± 2,1</b>	<b>61,3 ± 2,1</b>	<b>57,4 ± 1,9</b>	<b>0,58 ± 0,02</b>
	BERT-b	1e-3	55,0 ± 1,9	51,8 ± 3,1	58,1 ± 1,4	54,9 ± 1,9	0,56 ± 0,01
	BERT-l	1e-4	55,1 ± 1,8	51,5 ± 2,0	58,9 ± 2,6	55,1 ± 1,8	0,56 ± 0,01

**Fusion:** Word2Vec + GloVe; **BERT-b:** BERT base; **BERT-l:** BERT large.

Source: Authors

**Table 12.** Tri-class classification results with linear kernel

Trait	Feature	Accuracy	F1-score	UAR	κ
Extr	Word2Vec	40,7 ± 1,0	40,7 ± 1,1	40,7 ± 1,1	0,11 ± 0,02
	GloVe	41,6 ± 1,3	41,5 ± 1,3	41,6 ± 1,3	0,12 ± 0,02
	<b>Fusion</b>	<b>43,4 ± 1,1</b>	<b>43,1 ± 1,1</b>	<b>43,7 ± 1,2</b>	<b>0,15 ± 0,02</b>
	BERT-b	39,4 ± 1,0	39,4 ± 1,0	39,4 ± 1,0	0,09 ± 0,02
	BERT-l	43,0 ± 0,9	43,0 ± 0,9	43,2 ± 0,9	0,15 ± 0,01
Agr	Word2Vec	44,6 ± 1,5	44,6 ± 1,5	44,5 ± 1,5	0,17 ± 0,02
	GloVe	<b>47,1 ± 1,3</b>	<b>47,0 ± 1,3</b>	<b>47,0 ± 1,3</b>	<b>0,20 ± 0,02</b>
	Fusion	46,2 ± 1,2	46,3 ± 1,3	46,1 ± 1,2	0,19 ± 0,02
	BERT-b	45,3 ± 1,4	45,2 ± 1,3	45,3 ± 1,3	0,18 ± 0,02
	BERT-l	45,6 ± 1,2	45,5 ± 1,2	45,5 ± 1,2	0,18 ± 0,02
Cons	Word2Vec	41,7 ± 1,1	41,8 ± 1,1	41,6 ± 1,1	0,13 ± 0,02
	GloVe	43,9 ± 1,2	43,9 ± 1,1	43,8 ± 1,1	0,16 ± 0,02
	<b>Fusion</b>	<b>45,3 ± 1,6</b>	<b>45,6 ± 1,6</b>	<b>45,4 ± 1,6</b>	<b>0,18 ± 0,02</b>
	BERT-b	45,0 ± 1,6	45,2 ± 1,6	44,9 ± 1,6	0,17 ± 0,02
	BERT-l	43,3 ± 1,2	43,5 ± 1,2	43,2 ± 1,2	0,15 ± 0,02
Emot	Word2Vec	<b>41,3 ± 1,3</b>	<b>41,2 ± 1,3</b>	<b>41,3 ± 1,2</b>	<b>0,12 ± 0,01</b>
	GloVe	40,3 ± 1,4	40,2 ± 1,4	40,4 ± 1,4	0,10 ± 0,02
	Fusion	39,4 ± 1,5	39,4 ± 1,5	39,4 ± 1,5	0,09 ± 0,02
	BERT-b	37,8 ± 1,6	37,6 ± 1,5	37,8 ± 1,6	0,07 ± 0,02
	BERT-l	36,8 ± 2,1	36,5 ± 2,0	36,7 ± 2,1	0,05 ± 0,03
Open	Word2Vec	38,0 ± 1,4	38,1 ± 1,4	37,9 ± 1,4	0,07 ± 0,02
	GloVe	38,6 ± 1,5	38,6 ± 1,6	38,6 ± 1,5	0,08 ± 0,02
	<b>Fusion</b>	<b>38,9 ± 0,7</b>	<b>38,8 ± 0,7</b>	<b>38,6 ± 0,7</b>	<b>0,08 ± 0,01</b>
	BERT-b	32,5 ± 1,0	31,9 ± 1,0	31,8 ± 1,0	-0,03 ± 0,02
	BERT-l	36,2 ± 1,4	35,8 ± 1,3	35,7 ± 1,3	0,03 ± 0,02

**Fusion:** Word2Vec + GloVe; **BERT-b:** BERT base; **BERT-l:** BERT large.

Source: Authors