Politecnico di Torino
1859

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (35thcycle)

Natural Language Processing in Personality Traits and Basic Human Values Estimation of Social Media Users

By

Simone Leonardi

******

Supervisor(s):
Prof. Maurizio Morisio, Supervisor
Dr. Giuseppe Rizzo, Co-Supervisor
Dr. Luca Ardito, Co-Supervisor

Doctoral Examination Committee:
Prof. Akshi Kumar , Referee, Manchester Metropolitan University
Prof. Erik Cambria, Referee, Nanyang Technological University
Prof. Viviana Patti, University of Turin
Prof. Derwin Suhartono, BINUS University
Prof. Luca Cagliero, Polytechnic University of Turin

Politecnico di Torino
2023

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Simone Leonardi
2023

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my wife. We grew together and we are building up our dream life by trusting and loving each other. I also dedicate this work to my parents and to my sister for being a support in many ways. I'm glad to be your husband, your son and your brother.

Acknowledgements

This dissertation comes at the end of a three and a half year journey. I surfed through the high and down, always supported by renowned professors and dearest friends. I had the luck of not living alone especially during the lockdown period and this pushed me to the final goal of completing the doctorate school. The most important person to thank is my wife. She advised me day by day when I was struggling and at the same time she is the reason why I kept pushing through hard times. I thank Giuseppe Rizzo for believing in me as a researcher and for his precious advice. Without his competence and mastering of the topic I worked on and without his attention in every detail of my work I would have never reached the end of this path. With his experience and network in this research area he made the process of research and publishing a really better experience. He has always encouraged me to give my best by deeply refining what I wrote on paper, by correcting the experiment procedures, by asking me more and more when I showed some laziness. Thanks Giuseppe. I would also thank my supervisor Maurizio Moriso for letting me explore the research area I choose but always making me think with different points of view. His advice was so precious to me especially for his capability to pinpoint the fundamental meaning of all my publications. I would have never asked for a different supervisor. I thank my co-tutor professor Luca Ardito for the important interaction of what an academic path means, for helping me to become a better teacher and researcher, for illustrating in detail the bureaucratic steps I must adhere to reach my academic goals. Thanks professors Marco Torchiano and Antonio Vetrò, the teaching experience with you has been a source of inspiration for my future career. Every teaching activity has been a pleasure. I learnt a lot observing and helping you in several university courses. Thanks Riccardo and Diego, you have been my source of inspiration and nevertheless my senior colleagues. You taught me everything a PhD student need. From endless support in Latex to understanding how a mission should be prepared, from knowing where the correct documents were to how a paper should be written to not be rejected too much. I will always thank you for being both technical supporters and dearest friends. Finally I want to thank all of my friends that have been at Lab1 for the entire journey or for a part of it. I surely want to thank Francesco, Edoardo and all the thesists. Thanks Isa, we got the master's degree together and a part of the PhD. You've been an amazing friend and I love the way you defined your life path, you showed me the strength of intentions. Last but not least I want to thank Mariachiara. We built what I think a real friendship should be. I will never forget the long calls and lunch meetings talking about our dreams and confiding in each other when we want to complain. Without your happiness and kindness this path would have been really poorer. Thank you all for making me the researcher and the person I am now.

In addition, several experimental activities that I conducted would not have been possible without the computational resources kindly provided by the hpc@polito project (http://hpc.polito.it), to which I am grateful.

Abstract

This PhD thesis presents a comprehensive study on the application of natural language processing and machine learning techniques for various tasks related to social media analysis. The thesis includes four main parts: estimation of personality traits using multilingual transformer-based models, mining micro-influencers from social media posts, automated classification of fake news spreaders, and development of an educational chatbot to support question answering on Slack. The proposed models and methods are trained and evaluated using real-world data and demonstrate high performance in various tasks. The first part of the thesis focuses on the estimation of personality traits using multilingual transformer-based models and datasets such as Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism (OCEAN) and myPersonality. The goal of this research is to develop a model that can accurately estimate personality traits in multiple languages, which is important for understanding and predicting user behavior on social media platforms. The second part explores the problem of mining micro-influencers from social media posts, and the proposed solution is a Multi Input Micro-Influencers Classifier (MIMIC). The third part addresses the issue of fake news spread on social media, and presents an automated classifier for identifying fake news spreaders, with the goal of breaking the misinformation chain. Finally, the thesis concludes with the development of an educational chatbot to support question answering on Slack and its evaluation. Overall, this thesis aims to contribute to the field of social media analysis by proposing novel and effective solutions for various tasks and provides insight and recommendations for future research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Textual information from social media and an increased computational power of Natural Language Processing (NLP) models created the opportunity to better analyze human behavior online. The large amount of data nowadays available on the internet and in particular published every day in social media offer the ideal source of information for language analysis experiments. This dissertation focuses on linguistic models developed through deep neural networks. The field of application of these models are personality traits and basic human values assessments plus the use case scenarios of micro-influencers and fake news spreaders detection. The language itself is a complex ensemble of rules, implicit knowledge, semantics and syntax structures. The evolution of models to analyze it show different approaches detailed in Chapter 2. Our contribution to the research field is the construction of deep neural networks expanding and improving the recent discoveries in this field. The application of the related solutions are all around us in our day by day life. Think of language translation, autocompletion of messages when typing a text on a smartphone, bad language restrictions on social media platforms, chatbot helping to solve a issue or completing an order online. Furthermore new economies are emerging while new social problems are flooding the web. So it is the case of the commercial and sensitization campaigns disseminated by social media influencers along with the uncontrollable spread of misinformation.

This PhD thesis presents a comprehensive study on the application of NLP and machine learning techniques for various tasks related to social media analysis. The background section provides an overview of the current state-of-the-art in the field and sets the context for the research presented in the subsequent chapters.

The first part of the thesis focuses on the estimation of personality traits using multilingual transformer-based models and datasets such as OCEAN and myPersonality. The goal of this research is to develop a model that can accurately estimate personality traits in multiple languages, which is important for understanding and predicting user behavior on social media platforms. The proposed model is trained on a large dataset of multilingual social media posts and is evaluated using standard metrics such as accuracy and F1 score. The model is also compared to existing state-of-the-art models for personality traits estimation, to demonstrate its superiority. Moreover, this research contributes to the field by showing how to adapt pre-trained transformer models to estimate personality traits in multiple languages and how to fine-tune them on a multilingual dataset.

The second part of the thesis explores the problem of mining micro-influencers from social media posts. Micro-influencers are individuals with a relatively small but engaged following on social media platforms, who are often more effective at promoting products or services than more traditional celebrity influencers. The proposed solution is a Multi Input Micro-Influencers Classifier (MIMIC), which uses a combination of NLP and machine learning techniques to identify and rank potential micro-influencers. The MIMIC model considers various factors such as the number of followers, engagement rate, content quality, and sentiment analysis to determine the micro-influencer score. The MIMIC model is evaluated using real-world data from social media platforms, and the results demonstrate its effectiveness at identifying high-quality micro-influencers with high precision and recall. This research also contributes to the field by providing a novel approach to identify micro-influencers that combines multiple inputs and various factors to achieve a more robust and accurate classification.

The third part of the thesis addresses the issue of fake news spread on social media. With the increasing use of social media as a source of news and information, it has become crucial to develop automated systems for identifying and flagging false information. The proposed solution is an automated classifier for identifying fake news spreaders, which is trained on a dataset of social media posts and users. The classifier uses advanced NLP techniques such as sentiment analysis, natural language understanding (NLU), and topic modeling to identify patterns and signals that indicate fake news. The goal is to break the misinformation chain by preventing the spread of false information on social media platforms. The classifier is evaluated on a benchmark dataset and demonstrates high performance in terms of precision and recall. This research also contributes to the field by providing a comprehensive approach to identify fake news spreaders that takes into account multiple aspects of the user behavior and content.

Finally, the thesis concludes with the development of an educational chatbot to support question answering on Slack. This chatbot is designed to assist students in a classroom setting, providing quick and accurate answers to their questions. The chatbot uses NLP and machine learning techniques to understand and respond to student queries, and its effectiveness is evaluated in a real-world classroom setting through user studies and surveys. The results demonstrate that the chatbot can effectively assist students in their learning process and improve their understanding of the subject matter. This research contributes to the field by providing a practical application of NLP and machine learning techniques in an educational setting and showing how chatbots can be effectively integrated into classroom instruction.

In conclusion, this PhD thesis presents a comprehensive study on the application of NLP and machine learning techniques for various tasks related to social media analysis. The research presented in this thesis demonstrates the potential of these techniques for understanding and predicting user behavior on social media platforms, identifying micro-influencers, identifying fake news spreaders, and supporting student learning. The proposed models and methods are evaluated using real-world data and demonstrate high performance in various tasks. The thesis also provides insights and recommendations for future research in the field. Overall, the research in this thesis contributes to the advancement of natural language processing and machine learning techniques and their application to social media analysis.

Chapter 2

Background and Related Work

This Chapter is an overview of relevant publications in the Natural Language Processing, Micro-Influencer and Fake News Spreaders Detection, combined with the assessment of Personality Traits and Basic Human Values scores assessment from text. These papers act as a baseline for the publications of my doctorate. They are both baselines and reference material to support the experiment developed. In the following are listed and analysed all the related work divided per context and topics.

2.1 Natural Language Models in Personality Traits Assessment

The estimation of personality traits has advanced on both the linguistic tool and machine learning fronts in the past ten years. As a result of Kosinski et al. research [1], the field has seen exponential growth in interest since 2013, when they outlined the mining of personal data about people that can be retrieved from social media platforms. The Bidirectional Encoder Representations from Transformers (BERT) model, which Google released [2] and improved the exploitation of latent features in text semantics, has also caused disruption in the field of natural language processing, which has had a disruptive year. By machine learning category, we categorized the related works. We discuss the studies proving the lexical hypothesis, which states that personality traits can be calculated using language, at the end of this section.

2.1.1 Supervised Learning Approaches

Carducci et al. in [3] built a Support Vector Machine (SVM) to run a supervised regression on the myPersonality dataset. With 300 dimensional word embeddings as the feature vector and each personality trait score as the target, they created and optimized a SVM. These word-based embeddings were discovered by querying Facebook's pre-trained FastText vocabulary. Quercia et al. [4] used a supervised approach to perform a regression on the myPersonality dataset. Instead of incorporating linguistic features, their model calculates personality traits based on the followers, followings, and listed count. Alam et al. work [5] is based on Facebook data by which they developed a Big5 personality trait recognition algorithm. After contrasting a number of supervised models, they decided on a Multinomial Naive Bayes (MNB) sparse modeling. Additionally, they made use of linguistic features by employing a bag-of-words strategy and tokens (unigrams) derived from Facebook status updates. The Term Frequency - Inverse Document Frequency (TF-IDF) [6] algorithm is used to turn these tokens into a vector of features. In place of this, Chaudhary et al. [7] applied the Myers-Briggs personality assessment [8]. Incorporating user profiles and comments from Kaggle, they used a Logistic Regression algorithm to categorize the Myers-Briggs personality type indicator (MBTI). Xue et al. [9] developed a hierarchical deep neural network by utilizing recurrent and convolutional models. They used a technology that concatenates and pools word embeddings in their work with the sentence level attention mechanism, which takes social media posts into account as a whole rather than just as a collection of words. In order to perform and improve regression on personality trait scores, they used a multi-layer perceptron gradient that was boosted with an SVM. They used doc2vec by Gensim to create document embeddings. In [10], Liu et al. constructed a character to word, followed by a word to sentence, embedding mechanism to take the industry-specific lexicon of social media posts into consideration. The PAN dataset, which is a collection of tweets from Twitter, was used to compute the root-mean-square error (RMSE) for each of the five Big5 personality traits. They created a structure consisting of a Word Bidirectional Recurrent Neural Network (Word-BiRNN), a Character Bidirectional Recurrent Neural Network (Char-BiRNN), a Rectified Linear Unit (ReLu layer), and a final linear layer for regression. Spanish and Italian were also used in their work. In [11], Majumder et al. used Word2Vec word embeddings to build a CNN that produced a fixed-length feature vector that was then expanded with 84 new features using Mairesse's library [12]. The computed document vectors are fed into a polynomial SVM classifier as well as a multi-layer perceptron (MLP) classifier for classification.

2.1.2 Unsupervised Learning Approaches

Unsupervised learning produces a different strategy because there isn't a target feature here that is comparable to the myPersonality with questionnaire. In [13], Celli et al. used an unsupervised approach to use the linguistic correlations that Mairesse et al. [14] had established to compute personality traits. As an illustration, there is a correlation between a certain personality trait's high score and the quantity of commas used in the user text or the number of first person plural pronouns. Celli et al. transformed Mairesse et al. 's research into features that can be retrieved directly from social media posts. The cited work by Mairesse et al. is not regarded as unsupervised; however, their research and associated findings served as an inspiration for the subsequent unsupervised methods. Another unsupervised model created and used in the field makes use of linguistics. Software called Linguistic Inquiry and Word Count (LIWC) analyzes a person's cognitive and emotional abilities. Kafeza et al. [15] used LIWC to analyze personality traits in a controlled manner, but they masked this information to determine which community on Twitter has the most influence. AdaWalk, a program designed by Sun et al. in [16] that can detect personality on a group-level granularity, is unsupervised and capable of doing so.

2.1.3 Semi-Supervised Learning Approaches

An example of a semi-supervised learning method used is the one in [17]. The Big Five Inventory (BFI) by Berkeley Personality Lab was used by Bai et al. It is a self-reported questionnaire used to assess the target Big5 personality dimensions. The Graduate University of Chinese Academy of Sciences (GUCAS) students who gave them permission were monitored and their online behavior was collected using the Renren social network[1]. Every social media exchange was assigned a label corresponding to one of the Five Factor Model's five personality traits. Lukito et al. investigated the relationship between personality traits and linguistic preferences

[1] http://renren.com/

in [18]. They used a Twitter dataset that was geolocated in Indonesia and Naive Bayes to calculate these correlations. To extract text features, they used Linguistic Inquiry and Word Count. Applying a conditional random field to the entire set of Big5 model's five labels, Iacobelli and Culotta discovered a correlation between Agreability and Emotion Stability in [19] (high Emotional Stability equals low Neuroticism). The labels were seen as binary rather than continuous by them. Using this methodology, structured objects can be predicted. Their findings suggest that alternative structured approaches should be taken into account when dealing with Big5 models to improve the prediction accuracy of a trait, in their case Neuroticism, by taking advantage of the fact that another trait is easier to predict and it is correlated to the former.

2.1.4 Lexical Hypothesis in Personality Assessment

The method to infer personality from a person's linguistic choices is driven by the lexical hypothesis. Numerous studies on the subject support this position [20, 21, 14, 22, 23]. Kumar et al., in [24], used written text to extract the personality traits. They gathered a sizable text corpus from Facebook, Twitter, and essays, correlating the social network architecture with personality questionnaires. The research by Pennebaker et al. [25, 26] raises questions about the reliability of assessing personality traits from text. This theory was proved by [27] where Weisbuch et al. defined and calculated users' spontaneity in creating social media posts without spending a lot of time overthinking what they were posting in terms of the usefulness of social media in this field. This discovery enables a more accurate computation of personality traits. Algorithms used in these fields have changed over time since 2003. Argamon et al. [21] extracted lexical features and applied SVM to make predictions. For the purpose of detecting neuroticism, they used the appraisal lexical taxonomy. To categorize words, this taxonomy uses hierarchical linguistic principles. Concepts like homonymy, antonymy, hypernymy, and synonymy connect the words. Novel studies have emerged as a result of deep learning techniques' increased adoption and the escalating availability of data and machine resources. Su et al. [28] used dialogue transcripts and Linguistic Inquiry Word Count (LIWC) to annotate grammar. By computing the word distribution within textual documents, deep learning techniques convert text into a vectorial space. These numerical arrays acquire mathematical properties, like similarity, that can be used as features in the construction of models.

Finally, Vaswani et al. created an architecture in [29] that consists of a stack for both an encoder and a decoder. They preserved both a positional encoding embedding that stated the probabilities of each word position as well as the definition of a word embedding made by its context. When computing word embeddings, they introduced the idea of attention. An output was created by mapping a query to a set of key-value pairs. Through the use of a compatibility function, this output is calculated as a weighted sum of the values. Devlin et al. [2] utilized and modified the Transformer model to enhance the extraction and deployment of word embeddings when computed at the sentence level. They added the idea of a masked token to get the masked token's word embedding using a machine learning-based prediction mechanism.

The lexical hypothesis is applied in the experiment described in this thesis, We work under the umbrella of supervised approaches. In these cases, we fine-tune and hone the encoding portion of the Transformer architecture [29]. The encoder was created as a component of our model for extracting Big5 personality traits from social media posts. We make use of a unique token in order to enhance sentence representation. The pooling phase from word embeddings to sentence embeddings, which resulted in information loss in earlier studies, is avoided by this decision. We also develop a neural architecture that, in comparison to other models, both supervised and unsupervised, can perform regression calculations with a lower mean squared error.

2.2 Micro influencer detection with NLP and sociological scores

Numerous studies used metrics to assess people's capacity for influence on social media. To determine the top influencers in Austria, Anger et al. created the SNP (Social Network Potential) score system in [30]. In a related study, Bakshy et al. [31] discovered how influence analysis directly affects marketing investments. Bigonha et al. also examined social media posts in [32] to find out if users were discussing a specific subject and to determine whether they were evangelists or detractors.

When it came time to identify social media influencers, Kiss and Bichler created additional scores in [33]. Influence was compared to a virus that spreads to other people living in a close-knit group with the disease's source. Similar to this approach, Leskovec et al. [34] embraced centrality and diffusion as metrics relating to the relationship structure in social media friendship graphs. This strategy shares elements with that created by Easley and Kleinberg in their study *Networks, Crowds, and Markets* [35], where they discussed cascading behaviors and disease transmission. The study by Eliacik et al. [36] examined microblogging. Independent of the micro blogging subject, they developed metrics to identify which users are central and function as influencers in the connected community.

Studies that examine influence from a linguistic perspective offer a different perspective. Chen et al. [37] conducted research in this area by examining word usage in social media to determine whether it correlates with Reddit's community values. Using written text, Kumar et al. [24] have also attempted to extract personality traits and social values. The psycho-sociological makeup of the online community was a topic they covered. They conducted a thorough analysis on a text corpus from Facebook, Twitter, and a collection of essays, correlating it with personality questionnaires and social network structures. The validity of personality trait extraction from text was shown by Pennebaker et al. [25, 26].

Weisbuch et al. [27] provided evidence for the applicability of this method on social media by demonstrating how the typical user tends to post spontaneously without giving it much thought, allowing accurate personality detection.

Since 2003, algorithms used in these research areas have changed. At first, only SVM with lexical features was used. Argamon et al. [21] use lexical features, such as the Appraisal lexical taxonomy, to identify the personality trait of neuroticism. In order to categorize words, lexical taxonomies use hierarchical structures that adhere to linguistic conventions. Word relationships are described by homonomy, antonymy, synonymy, hyponymy, and hypernymy.

Due to the availability of greater computational resources and an increase in the amount of available data, as in the study by Su et al. [28], many research groups have started using deep learning more recently. In this instance, Su used the grammar annotation feature of LIWC (Linguistic Inquiry Word Count) on dialogue text that had been extracted. The use of deep learning techniques to process natural language avoids the need to define a previous set of lexical rules. Through the analysis of word distribution within documents, deep learning transforms words into vector spaces. Words can be transformed so that they acquire mathematical properties like similarity, which can be used to compute the angle cosine between two words to determine how similar they are.

Later, Majumder et al. [11] used a CNN to derive a fixed-length feature vector from word2vec word embeddings, which they then extended with 84 more features from Mairesse's library [12]. The resulting computed document vectors were fed into a multi-layer perceptron (MLP) and a polynomial SVM classifier for classification.

In our work, we apply SVM to a regression to determine personality traits, and we create a different algorithm to determine values based on the community. When words from the same semantic region are grouped together, it acts as a gravitational field. To map micro-influencers and the traits and values they are correlated with, we test a wide range of classifiers.

Another study by Mandelbaum and Shalev [38] found results indicating that pre-trained vectors are universal feature extractors and can be applied across datasets when using word embeddings to categorize sentences. Global Vectors (GloVe) [39] are used in our work. GloVe offers pre-trained word embeddings built using a Wikipedia corpus. We establish the spatial word representation using embeddings as a vocabulary. Roy Schwartz et al. [40] in where they proposed an improved similarity prediction method and also defined the distinction between words similarity and association, described the significance and diversification of various approaches in word embeddings. They emphasized how word clustering is influenced by hyponymy, meronymy, and antonym.

2.2.1 Detection and Classification of Micro-influencers with social media graph scores, text and images

In the context of social network analysis (SNA), the identification or categorization of online influencers is a wide research area. Due to its socioeconomic impact, it has produced numerous publications and continues to gain importance [41–47]. In [48], Rabiger et al. discuss the characteristics of complex graphs made of nodes and connections to describe the dynamics of influence and interaction between users. The social graph varies depending on the problem being studied and the model used to describe friendships. Lü et al. divide the methods for finding influencers into two categories in [49]. The first one includes strategies for increasing influence through the identification of influential users in line with a diffusion model. The second category compiles solutions for measuring influence in social media graphs and looking for locally influential nodes using network scansion. Kwak et al. present a centrality-based strategy in their paper [50]. By analyzing structural data, they identify influencers. In accordance with graph theory, the definition of centrality used in their research is a gauge of a node's significance within a graph. The ability to adjust to various social network topologies in practical applications is essential for future improvements of this work. The extent to which a network collapses or becomes less functional when a node is removed and its connections are severed is measured by Chen in the parallel work [51]. Due to their ability to adapt to unstructured network topologies, machine learning approaches have become more prevalent in this field of study over the past ten years. To find important nodes in a social media graph, Fan et al. develop a deep reinforcement learning framework in [52]. They perform better when the learning set and data are of the highest quality, but this method is less effective with noisy data. The mixed approach Roelens et al. suggest in [53] makes use of network analysis and machine learning algorithms. By taking advantage of links between nodes, they apply an influence cascade throughout the network. Even though this method depends on network parameters, it also considers account information and user behavior to increase accuracy on large scale networks. In [54], Gan et al. create a fully data-driven methodology to find micro-influencers and a specific ranking system to match them with businesses exhibiting similar traits. They use the bilinear pooling technique that Kim et al. described in [55], which makes use of both visual and textual information. In order to balance the weights of each feature due to their various input sizes, the model applies a linear transformation and a non-linear activation function to each feature. The research done by Gan et al. In a K-buckets system, micro-influencer engagement power and similarity to the associated brand are measured. With a pipeline for natural language processing, Bashari et al.'s method in [56] focuses solely on text analysis. They gather User Generated Content (UGC) but do not take into account user interactions on the social media platform. They apply TF-IDF on the UGC after cleaning and preprocessing the data, weighting each word, and then map these features with captions and hashtags. In the second stage, Bashari et al. fed these features into two SVMs to categorize users in a supervised setting. In a similar way, Zheng et al. [57] Using their on-Demand Influencer Discovery (DID) framework, analyze keywords. Even in this case, they don't take into account user connections or popularity. Their algorithm uses a Language Attention Network to pick certain social media posts. related to the specified keyword and an Influence Convolution Network to influence the influence propagation on social media with local aggregation methods. A bidirectional recurrent neural network (RNN) is used to learn the hidden state of each word after each word is mapped to an one-hot encoding representation. The matrix hidden state is retrieved by a final attention layer, which also creates a classification output by combining it with an external topic seed to extract information from the original post that is relevant to the topic. In a recent study [58], Zhuang et al. used a multidimensional social influence (MSI) measurement approach to identify influencers. In terms of various influencing mechanisms on social media, they offer a framework that is thorough. Measurements of information influence, action influence, and structure influence are all described. Even though we draw inspiration from all of the research papers that have been presented, we create a model that can combine existing solutions into a broader framework, resulting in a pipeline that deals with textual, visual, and account-based information from social media.

2.3 NLP in Fake News Spreaders Detection

The spread of online misinformation during the 2016 presidential election and on social media sites like Twitter and Facebook has resulted in numerous publications in this area [59–65]. In [59], Alcott et al. defined fake news as articles that are certain to be false and that deceive readers in [59]. An affirmation of a false place of birth is an example of fake news. Using information from public registries, it was possible to confirm that this information was false. From the perspective of data mining, Shu et al. described fake news in [60]. They discussed the differences between fake news and related data from social media platforms, which are primarily driven by malicious accounts and echo chambers, and traditional media, which has stronger psychological and sociological foundations. After being gathered, these false reports have been categorized based on their textual content and social context. In actuality, they are distinguished by a news item that has been verified as false, the use of a particular linguistic style, the explicit support or denial of the news's content, and, finally, the manner in which they spread throughout the social community.

Lazer et al. suggested in [61] that social media platforms and their mechanisms for disseminating content are ideal environments for fake news. They suggest doing research on the issue's resolution and developing bot (software-controlled accounts) and automatic content detection tools to assist with human supervision in order to avoid either corporate or governmental censorship. In [62], Stella et al. discussed the issue of bot detection and its impact on online communities in the context of the 2017 Catalan independence referendum. They investigated the metrics of the social graph, including sentiment analysis and the origin and destination of messages sent between groups. They discovered that social robots and people behave differently and that the former frequently causes inflammatory responses in people. The methods used in the works of Grinberg et al. [63], Guess et al. [64], and Pennycook et al. [65] were all driven by the characteristics of social media users and associated social graph metrics in relation to political elections. To better understand recurring patterns research on the viral content diffusion mechanism through social media spread [66–69]. In particular, Shu et al. [66] created Hoaxy, a Twitter monitoring tool, to better understand how fact-checking news and fake news spread differently. They found that social media bots were at the center of the diffusion network and that fact-checking news only affected the edges of the same network. Their entire body of work is based on social media graph metrics like in and out degree, PageRank, and network diffusion steps. Dhamal discovered in [67] that using highly influential nodes of a social network community to spread information on a multiple phase scenario does not increase the diffusion effectiveness, whereas using less influential nodes in succeeding phases maintains a high rate of diffusion. Goyal et al. explained in [68] how social media users are influenced by their neighbors when they take actions like sharing news. They created a mathematical model based on social media graphs to forecast the likelihood that a piece of information will spread through particular social media nodes. According to Vosoughi et al.'s analysis of true and false news diffusion behavior on Twitter in [69], social media bots spread both types of information at the same rate, suggesting that people play a significant role in the cascading spread of false information.

In parallel, other research works [70–73] examined the effects of network topologies and influence characteristics of specific nodes of the social media communities in the information diffusion mechanism. Zhang et al. [70] determined the likelihood that a node in a social media graph will spread information based on the behavior of its neighbors, a new metric called social influence locality was developed. They reaffirmed the idea that node behavior independently of each other's behavior and connection typologies is also significant. Guo et al. [71] compared the effects of key influences on a node's action in a social network graph between global and local influencers, and discovered that the local ones have a higher probabilistic footprint. Citing [72], Mansour et al. In the context of information spreading online, emphasize the role played by interpersonal influences between people having the same experience. According to this finding, analyzing user features can help predict users' social media sharing behavior more accurately. Citing Borges et al. [73], the researchers investigated the drivers behind user behavior in online communities that spread viral messages. They discovered that users who participate in the spreading of viral content avoid discussions about the actual content. Based on how users responded to viral content, they divided users into three categories. These are classified as heavy, socially- and search-driven. Because they interact and produce content on social media platforms frequently, frequent users are impacted by the significance of the content. Users who are socially motivated engage with content by sharing it primarily without providing additional context, whereas users who are search motivated do neither. The last two categories are more concerned with the effect that the information will have on how other users will perceive them as a person than they are with the significance of the news content.

In a similar vein, the computational power and efficacy of the linguistic tools and machine learning techniques used to extract information from social media posts and text in general have increased exponentially. The misinformation field has been examined using the NLP (Natural Language Processing) methodology since 2018, when the BERT model by Google [2] and the transformer-based architecture combined with the attention mechanism by Vaswani et al. were published [74]. Rather than finding a correlation between neutral tweets and the likelihood that a social media post will be shared [75, 76], Stieglitz et al. found a positive relationship between the quantity of words containing both positive and negative sentiment in [75]. As a result, in addition to the news's actual content, sentiment is also disseminated via social media. Jiang et al.'s research supports this idea in a political context [76].

As with linguistic cues, the personality of the user also affects whether or not fake news is shared. These characteristics have been used by a number of researchers to examine the relationship between personality traits and social media use [77–79]. An agent-based simulation of a social media interaction has been created in [77] by

Burbach et al. These agents were developed by modeling the responses to an online survey. Age, gender, education level, aspects of one's social network, and personality scores from the Dark Triad, Regulatory Emotional Self-Efficacy, and Five Factor Model were among the data that were retrieved. The virtual environment was built using Netlogo[2], and tests on agent interaction and the spread of false information within the network were conducted there. They discovered that social media graphs, the quantity of connections, and the centrality of nodes have a bigger influence than personality scores. Even though this project was tested in a simulation scenario, it suggests that the solution to the problem of fake news diffusion comes from a multifaceted approach that draws on both the psychology of users and the structure of social networks. According to Ross et al.'s description in their book [78], a user's personality can affect how they behave when using Facebook. Similar to how personality dimensions affect information diffusion in social media platforms, Heinstrom et al. described this phenomenon in [79].

Giachanou et al. dealt with fact-checkers and those who spread misinformation, they developed a user-centered CNN model [80]. The LIWC (Linguistic Inquiry Word Count) dictionary [81] and the Five Factor Model's linguistic features associated with personality traits were used to create a multi-input CNN. The 300-dimensional pre-trained GloVe embeddings [39] are used in their word-based model to convert textual tweets into a 2D embeddings matrix. These embeddings serve as the input for a convolutional layer, which then computes personality traits before merging them with manually extracted LIWC features. This strategy is novel because it considers both a user's linguistic patterns and personality traits in the context of fake news. Additionally, it suggests a solution that makes use of the social media users' behaviors and motivations. However, this work has some significant flaws. For example, the computed personality traits have not been verified using a ground truth dataset or a questionnaire, causing the initial error to propagate through the neural network's subsequent layers. This research project also has some issues with the labeling process because it heavily relies on the presence of specific words linked to fact-checking or false claims, such as hoax, fake, false, fact-checking, snopes, politifact leadstories, and lead stories. Additionally, tweets are classified as fake if they are retweets of original fake news. This labeling process is not completely error-proof, despite having been manually reviewed over 5,000 tweets. Furthermore, no consideration is given to categorizing a person's stance in order to judge whether

[2] https://ccl.northwestern.edu/netlogo/

they support or deny fake news. The binary classifier has room to be improved because the total number of users analyzed was under three thousand, and the final f1-score was below 0 point six.

These projects suggest that this area of study can be divided into two major categories. The first one employs natural language processing models to identify fake news content. In order to determine how misinformation spreads among social media users, the second area tracks the topologies of social networks.

2.4 Deep learning and multimodal approaches

Some recent developments in this research area demonstrate the effective correlation between personality traits and emotion detection. In [82], researchers highlight the strong link between personality traits and emotions. This paper proposes a novel multitask learning framework that predicts both personality traits and emotional behaviors. Different information-sharing mechanisms are explored and evaluated. To ensure high-quality learning, a model-agnostic meta-learning-like framework is adopted for optimization. The multitask learning model achieves state-of-the-art performance on renowned personality and emotion datasets, surpassing language model-based models in efficiency.

While with the survey in [83], Dehlim et al. describe how the emergence of personality computing has led to a proliferation of personality-aware recommendation systems. These systems address issues like the cold start and data sparsity problems. This survey classifies and examines personality-aware recommendation systems, making it the first of its kind. It compares the design choices, including personality modeling methods and recommendation techniques. Additionally, it highlights commonly used datasets and challenges in personality-aware recommendation systems.

Finally, in [84], Cambria et al. define that AI research has shown great potential but faces limitations in tasks requiring commonsense reasoning. This work proposes a commonsense-based neurosymbolic framework to address these limitations in sentiment analysis. By employing unsupervised subsymbolic techniques like language models and kernel methods, we build trustworthy symbolic representations for converting natural language to a protolanguage. This approach enables the extraction of polarity from text in an interpretable and explainable manner, overcoming the key limitations of current AI models.

In a parallel scenario, in [85], the authors show how in Asian social networks, collectivist culture plays a dominant role, influencing word-of-mouth and opinion leaders. This work introduces a modified spider monkey optimization-based approach for detecting opinion leaders. It utilizes modified node2vec graph embedding to generate lower-dimensional vectors as initial features. The population is divided into clusters using the k-means++ algorithm, with the cluster centers representing local and global leaders. These leaders form the seed set for opinion leaders. Node positions, including leaders, are updated iteratively. The proposed approach is tested using information diffusion and cognitive opinion dynamics models on real-life social networks, outperforming existing techniques in opinion leader detection.

In [86], Kumar et al. present a research work where they explain how Automated sarcasm detection in Hindi, a morphologically-rich and free-order language, presents a unique challenge due to limited linguistic resources. Context incongruity is crucial in detecting sarcasm, and linguistic, aural, and visual cues are utilized for prediction. This research introduces a hybrid deep learning model that incorporates word and emoji embeddings to detect sarcasm. Validated on a manually annotated Hindi tweets dataset, Sarc-H. The study demonstrates the importance of emojis in sarcasm detection and highlights the effectiveness of automated feature engineering for low-resource languages.

In [87], the researchers define that healthcare social networks play a crucial role in personalized and connected healthcare, but barriers exist in building resilient and robust technology-driven healthcare systems. Social media's susceptibility to rumors and fake news poses risks to society. Researchers have been studying information diffusion, particularly rumor propagation, which has gained significant attention. Traditional models focus on one-directional propagation, involving only supporters, while real-life situations involve both supporters and deniers. This paper presents a model for simultaneous rumor propagation and control. The Susceptible-Infected-Recovered-Anti-spreader model, based on epidemic spreading, captures multiple reactions to rumors, including posting, deleting, and debunking. The model also compares node centrality algorithms by simulating spreaders and anti-spreaders. Experimental results on real-world network datasets validate the proposed model.

In conclusion, in [88], Kumar et al. prove how exposure to false information can harm democracies, polarize public opinion, and fuel extremism. Detecting fake news in distributed platforms is challenging. To enhance trustworthiness, this article presents OptNet-Fake, a hybrid model for fake news detection. It utilizes a meta-heuristic algorithm to select useful features and trains a deep neural network on social media data. Textual data is processed using TF-IDF to extract d-D feature vectors. A modified grasshopper optimization algorithm selects salient features, which are then processed by convolutional neural networks with different filter sizes to obtain n-gram features. The extracted features are combined for fake news detection. The model is evaluated on real-world datasets, surpassing other algorithms and demonstrating superior performance.

2.5 Excursus on emotion categorization models and algorithms

Emotion categorization models and algorithms in natural language processing (NLP) aim to automatically identify and classify emotions expressed in text. These models and algorithms help analyze sentiment, mood, or affective states, allowing applications to understand and respond to human emotions in a more personalized manner.

Rule-based approaches involve manually defining a set of rules or patterns to identify and categorize emotions in text. These rules can be based on linguistic features, such as keywords, syntactic structures, or sentiment lexicons. While rule-based approaches are relatively straightforward to implement, they often lack flexibility and struggle with handling nuanced expressions of emotion.

Lexicon-based approaches utilize pre-defined emotion lexicons or dictionaries containing words associated with specific emotions. Each word in a given text is assigned emotion scores based on its presence in the lexicon. The scores are then aggregated to determine the dominant emotion(s) in the text. However, lexicon-based approaches may face challenges with polysemy (multiple meanings of words) and may not capture contextual nuances effectively.

Machine learning (ML) approaches involve training models on labeled emotion datasets to learn patterns and make predictions on unseen text. These models can be categorized as follows:

Supervised learning models, such as Support Vector Machines (SVM), Naive Bayes, or Neural Networks, are trained using annotated data where each text instance is labeled with an emotion category. These models learn to generalize patterns from the training data and classify emotions in new, unseen text. However, supervised learning approaches require a significant amount of labeled data for training, which can be time-consuming and expensive to create.

Unsupervised learning approaches aim to discover hidden patterns or structures in the data without the need for labeled examples. Techniques like clustering, topic modeling, or dimensionality reduction can be used to group similar texts based on the underlying emotions they convey. Unsupervised approaches are advantageous when labeled data is scarce or unavailable, but they may struggle to assign accurate emotion labels without human validation.

Transfer learning leverages pre-trained models, such as language models like BERT or GPT, which have been trained on large-scale text corpora. These models learn general language representations that can be fine-tuned for specific tasks, including emotion classification. Transfer learning approaches can achieve impressive results even with limited labeled data, as they capture the contextual information and semantic relationships present in the pre-training corpus.

Deep learning models, particularly recurrent neural networks (RNNs) or variants like long short-term memory (LSTM) or gated recurrent units (GRU), have been successfully applied to emotion categorization tasks. These models can capture sequential dependencies in text and model contextual information effectively. Attention mechanisms, which allow the model to focus on important words or phrases, are often incorporated to improve performance. Deep learning approaches require large amounts of labeled data and computational resources for training but have demonstrated state-of-the-art results in emotion classification.

It’s important to note that the performance of emotion categorization models heavily relies on the quality and representativeness of the training data, the diversity of emotions captured in the labels, and the domain-specificity of the models. Furthermore, combining multiple approaches or ensembling different models can enhance performance by leveraging their respective strengths.

Overall, emotion categorization models and algorithms using NLP have made significant progress in recent years, enabling various applications like sentiment analysis, chatbots, and personalized recommendation systems to better understand and respond to human emotions.

Chapter 3

Multilingual Transformer-Based Personality Traits Estimation

The adoption of language models in social application has been widely tested. From detection of cyberstalking [89] through the monitoring of social media users' depression [90]. As per every communication medium, the written language contains markers of personality, values and behaviors. This insight helps recruiters and candidates to better match an ideal job going beyond technical skills. In [91], Neal et al. proved how this approach led to a higher success rate in targeted job advertisements. The ability of software powered by artificial intelligence to better empathize with humans found an use case also for commercial use. The IBM (International Business Machines Corporation) developed Personality Insights along with an airline company in Japan to interact with customers on long travel [1]. These examples offer a partial proof that it is possible to retrieve psychological scores from language written or transposed from speech. They also went beyond by exploiting this information to improve and personalize the interaction with the user. Nevertheless there are some major defects that need to be addressed such as the lack of resources in understanding word polysemy and the overspecialization in the English language. Languages are complex and words are no more inspectable without their context. Furthermore when you move from a language and its related culture to another you do not have an exact mapping. In fact, you must consider different images and symbols for the same meaning. There is also a specific bag of words per country and sometimes per town. Not all words have a direct translation

[1] https://www.ibm.com/blogs/client-voices/ai-personalizes-japan-airlines-travel-experience/

into English so we cannot work with only this source of information. Given this premises, the search for a model to rule them led us toward the formulation of this two questions in our research:

RQ1 Is there a natural language processing model that is effective in the assessment of personality traits from written text?

RQ2 Can we make this model work multilingual and so make it work in a multicultural setup without gaining errors?

In this chapter, a deep neural network stacked with Transformers [29] shows how to answer the previous questions. It works considering sentences and documents as a whole. This characteristic removes the error of giving the same meaning and vectorial representations to two words with the same written form but different semantics. The MTPTE (Multilingual Transformer-based Personality Traits Estimator) converts sentences into multidimensional arrays. Then it computes the five personality traits as described by the theory of McRae and Costa [92]. The five traits are Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism. The specialized encoders collect and transform the information starting from social media posts. Then the model, trained on the myPersonality gold standard dataset, computes the desired personality trait through a regression. This approach reduces the means squared error of the baseline state-of-the art computed on the myPersonality dataset. In addition to that, MTPTE is able to compute personality traits on the top 104 languages per number of articles existing in Wikipedia. The code build is available in a public GitHub repository. [2]

The sections 3.1 and 3.2 of this chapter describe the five factor model and the myPersonality gold standard as released by Celli et al. [93]. It is no longer public, so if you want to access it you must write to the authors. Section 3.3 defines MTPTE neural architecture and it provides details of our approach. Section 3.4 reports the overall results obtained both with the English language alone and with the multilingual model. Section 3.5 is useful to open the discussion about achievement and improvement, but also to highlight flaws and limitations. There Weexplain the comparison between MTPTE and other models used as baseline. Section 3.6 lists the conclusive insights and the chances for future work in this direction.

[2] https://github.com/D2KLab/SentencePersonality

3.1 Five Factor Model

The Five Factor Models is a notorious reference model chosen among many psychometrics standards to assess the personality of a person. It is also known as the Big5 or OCEAN model for the five scores it describes. The model integrates various aspects of personality constructs efficiently as described by McRae et al. in [92]. Tupes and Christal in [94] along with the work of Digman [95] and the seminal study by Goldberg [96] concur in the corroboration of this model.

OCEAN is the acronym of the five scores:

- Openness to Experience computes how high is the tendency of a person to pursue new adventures by exiting his comfort zone. Vice versa low Openness means the person is bound to his roots and habits.
- Conscientiousness defines a person as plan oriented or messy and creative, led by improvisation. High score of Consciousness means the person needs to plan a lot ahead, while lower scores characterize him as spontaneous and ready to respond to contingencies.
- Extraversion tells a person is outgoing. High levels of this score are a signal of the love of being the center of attention. With lower scores the candidate prefers not to be noticed instead. A person is introverted if he likes to stay alone or in any case to be reserved.
- Agreeableness is a synonym of gregarious. High scores means altruism and blind trust towards other people. Usually, a person who is highly agreeable tends to meet new friends easily. If a person is not agreeable he is prone to not change his mind in an argument and be loyal to his view of the world.
- Neuroticism monitors the resistance to stressful situations. With a low level of neuroticism a person is emotionally more stable. High scores are associated with the beginning of depression, easy reactions and low resistance to gamble and marketing attacks.

A questionnaire on a Likert continuous 1-5 scale collects the scores mentioned in the Five Factor Model. In the work of MacRae and Costa [92], the authors also highlight subscore under the hat of the major five and they call them facets. In this thesis Wedo not investigate them but they are surely of interest for further work in this research field. We want to disclaim that users are not clustered in groups or for any other exploitation rather than assessing their personality scores.

3.2 myPersonality dataset

The training phase of the Multilingual Transformer-based Personality Trait Estimator requires a gold standard widely adopted by researchers to compare our solution with this. The reference dataset for personality traits assessment is myPersonality released by Celli et al. [93] and described and analyzed by Kosinski et al. [1]. They collected this data made of pairing between social media posts and answers by candidates through a web application disseminated on Facebook. The questionnaire given to users is a short version of the original MacRae and Costa NEO-PI-R [92]. In the questionnaire each answer to a question is a part of the final sum for a specific trait. "Warms up quickly to others" is an example sentence of the questionnaire. The user has to choose in the range $[1-5]$, where 1 means *strongly disagree* while 5 stands for *agree strongly*. This specific example is one of the questions associated with the Extraversion trait. This model receives as input 9913 samples, a subset called *myPersonality small*. There are two files in the dataset. The first has lines composed of id of the post, id of the author, plain text of the post, plus other information not useful in this application. The second file is simpler because it just contains the user id and the five scores obtained by the user with the questionnaire.

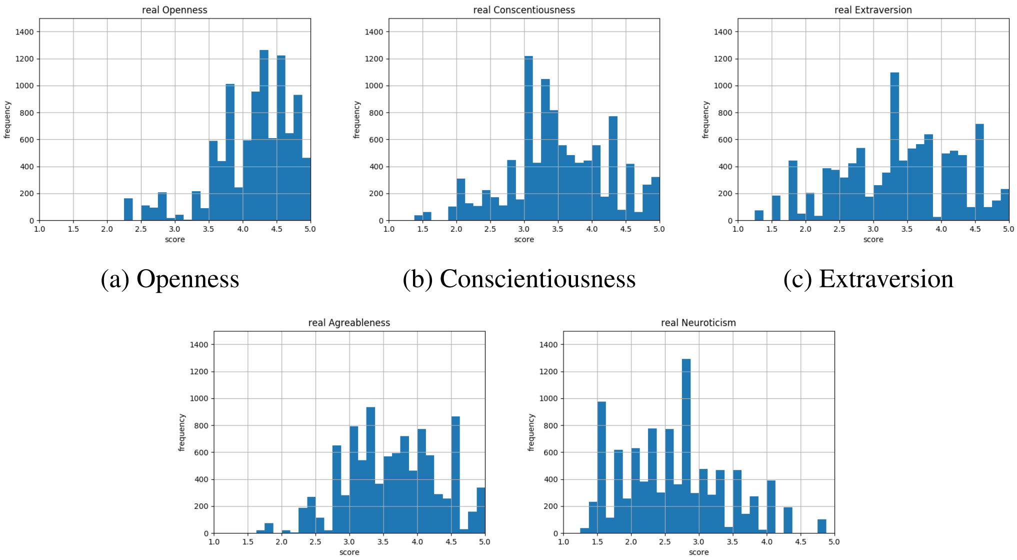
The first line of the first file is this one:

> 1,d2504ff7e14a20d0bb263e82b77622e7,"is very well rested. Off to starbucks to catch up with a friend.",2009-06-15 14:47:52,67

The second file appears as made up of lines like the one in this example:

> 605ff548660b7ed55d519b0058b9649e,4.20,4.50,4.25,3.15,2.05 [. . .]

Celli et al. collected these data starting in 2007 to conclude the experiment and the gathering of information in 2012. All users related data have been collected by obtaining a written explicit consent through a user agreement contract. Anyway, from 2018 this dataset is no longer publicly available and Kosinski explains the reasons in the project website. [3]



(a) Openness (b) Conscientiousness (c) Extraversion

(d) Agreeableness (e) Neuroticism

Fig. 3.1 Represents the real distribution of data inside myPersonality small. As the reader may notice they are far from a standard Gaussian bell. From left to right and from top to bottom are listed the five personality traits scores distributions. The plots show the answers are given by an heterogeneous set of candidates. At first glance it is also clear that the margins 1 and 5 of the scale are less represented, this condition will also impact the capability of any models to exactly predict these two scenarios in a future candidate that will be analyzed. There are also peaks suggesting that the questions tend to push users towards certain answers or that problematic or extreme personality characteristics are not present in the candidates samples. The graphics show scores grouped in a discrete scale to make more clear and readable the information of data distribution.

The Fig 3.1 depicts the data frequency distribution of the five scores collected through the questionnaire inside the myPersonality small dataset. In the following section I'll describe how Weused them to train the Multilingual Transformer-based Personality Trait Estimator.

[3] https://sites.google.com/michalkosinski.com/mypersonality

3.3 MTPTE Model

MTPTE (Multilingual Transformer-based personality trait Estimator) is a linguistic model built using a continuous regression-capable neural network architecture and a sentence-level attention mechanism. The model is evaluated against baselines already in existence using the *myPersonality* dataset as the gold standard. Using only the text posted on *Facebook*, it better predicts the social media user's personality. The steps to process the input text and create the stacked neural network for each of the five personality traits are described below. A stacked neural network is described as a collection of openly accessible neural network architectures whose features have been extracted at a middle layer of the network and have been concatenated to create a larger network.

3.3.1 The usage of Transformers in sentence embeddings

A dataset of textual social media posts serves as the basis for our analysis. Because we want to preserve the meaning expressed by the original social media author, the text in Figure 3.4 that is provided as input to our processing pipeline is given in its raw format. The removal of @, *http* and punctuation is the only text cleaning that is done, in accordance with this logic. The BERT-tokenizer breaks sentences into words and then into sub-strings to deal with words that are not in the vocabulary. We use *spacymoji*[4] to take into account emoticons because that's the social media context in which we work. We convert emoticons from their graphical to textual descriptions. After the initial tokenization process, each sentence from myPersonality is displayed as a list of tokens. After splitting, we add a special token (CLS, which stands for classification and is trained as a custom token in the model) to the token list at the top to perform sentence classification. You should be aware that we perform a regression task using the CLS token embedding. The BERT [2] architecture has adopted the CLS syntax to support a variety of tasks. Associating labels with this unique token trained with the Transformer architecture and the Attention mechanism as described in [29] is one of these tasks. The CLS token eliminates the requirement for word embedding concatenation and sum at the end of encoding. Through this behavior, the issue is moved from a word-based mechanism to a sentence-level one problem. SEP, or separator, a special token that was also trained as a custom token in the model, is

[4] https://pypi.org/project/spacymoji/

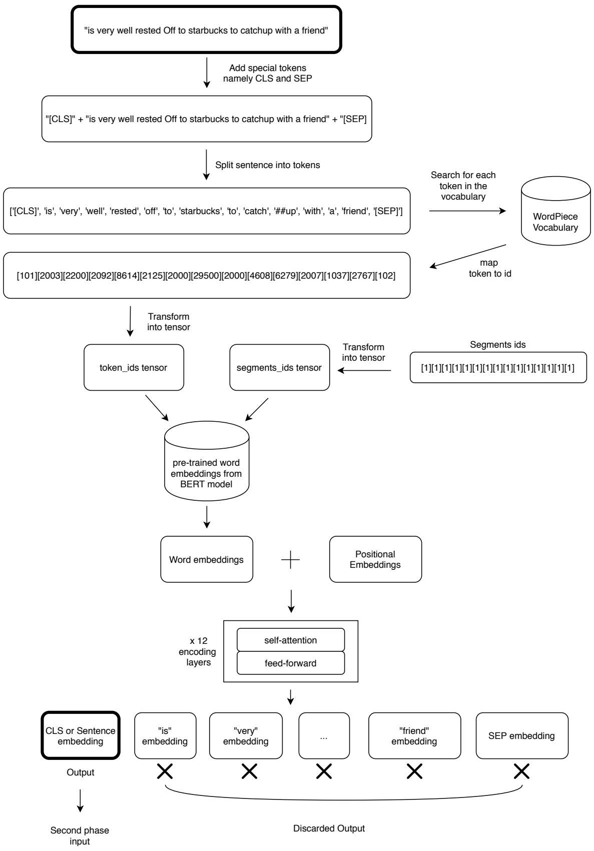


Fig. 3.2 Utilizing Transformer for tokenization and encoding. The processing of each sentence in the myPersonality dataset is depicted in the figure. After removing the punctuation, we add the SEP token (separation between sentences, pre-trained in BERT model as a custom token) at the end of the sentence and the CLS token (classification task special token, pre-trained in BERT model as a custom token) at the beginning of the sentence. Next, we divided the sentence into tokens. By dividing them into sub-tokens, the model is able to take into account words that are not in its dictionary. To mark the second part of the split words as not a standalone word, ## is placed before it. The WordPiece vocabulary's id is used to map tokens to, and the array so computed is transformed into a Tensor. In addition, a tensor called segments_ids tensor made of one, with the same length as the token_ids tensor, is required. When we perform a task that requires two sentences, for example, question answering or next sentence prediction, the segments_ids is useful to separate tokens belonging to the first sentence (zero) to the second's (one). For our situation, we only require one sentence, so we load segments_ids with it. In order to generate word embeddings from our tensors, we load pre-trained embeddings from the BERT model and add an initially random positional embedding. Twelve encoding layers with a self-attention network and a feed-forward network inside that encode the input into the final sentence embedding are shown at the bottom of the figure.

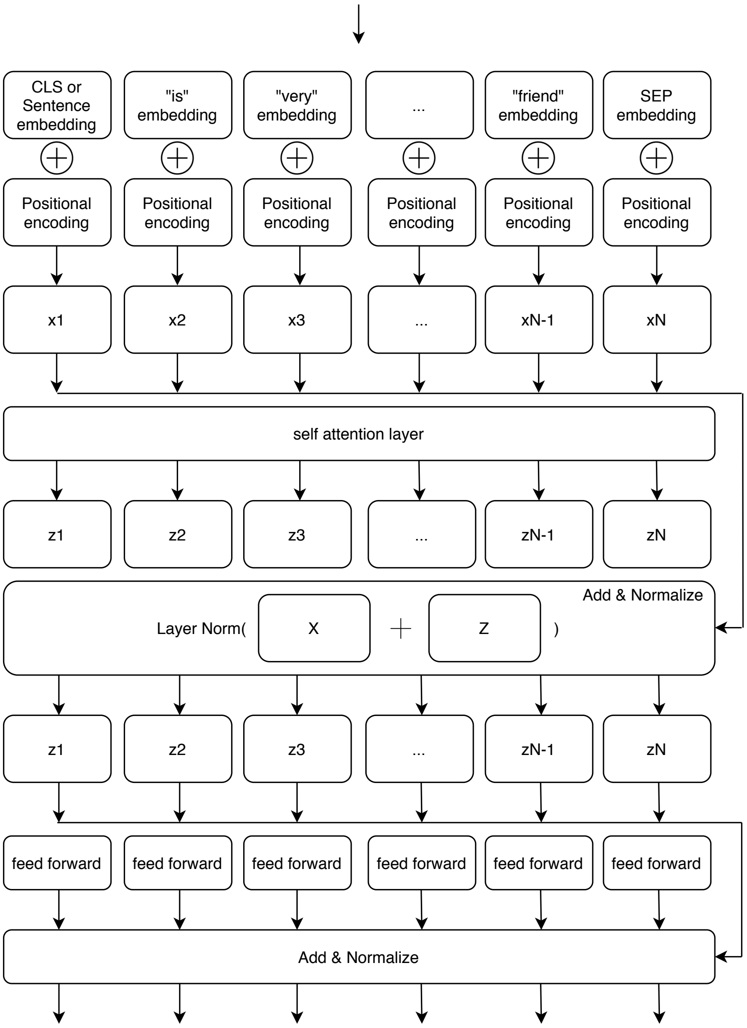


Fig. 3.3 This depicts one of the encoding layers mentioned in Figure 3.2 in its entirety. In the final architecture, there are twelve of these encoding layers. Each token's word embedding passes through these encoding layers before being transformed at the end.

added at the end of each sentence. Then, using Devlin et al. 's description [2], we use the BERT positional embeddings and the subsequent twelve encoding layers to create a word embedding for each token. After this stage, all embeddings but the CLS or sentence embedding are removed. The sentence embedding is chosen as a representative of the entire sentence because it was calculated taking into account the surrounding context, in this case the entire sentence. The optimal configuration of the BERT base model results in a single token with 768 dimensions. The authors of [2] suggest that the best number of features for comparing the outcomes of the following tasks: GLUE, SQUAD, NER, and MLNI, is 768, which comes from the empirical experiment described in that publication. We perform the following operations directly on this 768 dimensional array without the need for pooling or other types of aggregation because the sentence has been embedded. From input tokenization to word embedding to sentence embedding, Figure 3.2 depicts the text processing steps. The optimized and lighter version of BERT used in the encoder architecture is shown in Figure 3.2. The architecture was built upon the bert-as-a-service library version[5]. The decision to use a sentence-level attention mechanism was made under the presumption that a word that is used in multiple contexts cannot be represented by the same word embedding. The Winograd Schema Challenge (WSC)[6] must be solved to deal with this situation, which is known as polysemy in text. Hector Levesque provided the following example in 2011: "The trophy was too large to fit in the brown suitcase." Although it is obvious to the reader what was excessive, it is not simple for an intelligent agent. In sentences like the one reported by Google [7] - I arrived at the bank after crossing the road - there is another instance where polysemy in text needs to be addressed. In our case, where each word is important for comprehending the social media post, sentence-level attention mechanisms are appropriate. The weight of each word is greater in this context because we have medium and brief sentences in particular. To understand the meaning of the sentence and accurately predict the personality trait scores connected to it, we must carefully consider each word.

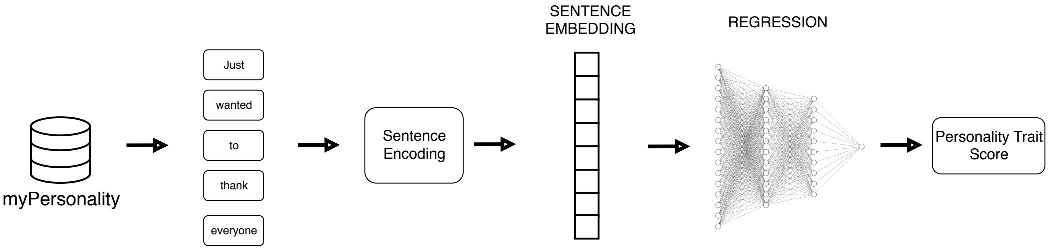


Fig. 3.4 The pipeline depicts the entire procedure from unprocessed text through sentence encoding and finally the stacked neural network to predict personality traits. Out of the five personality traits in the Big5 model, one is computed, as seen in the figure. The same pipeline is used to compute each of the five personality traits one at a time.

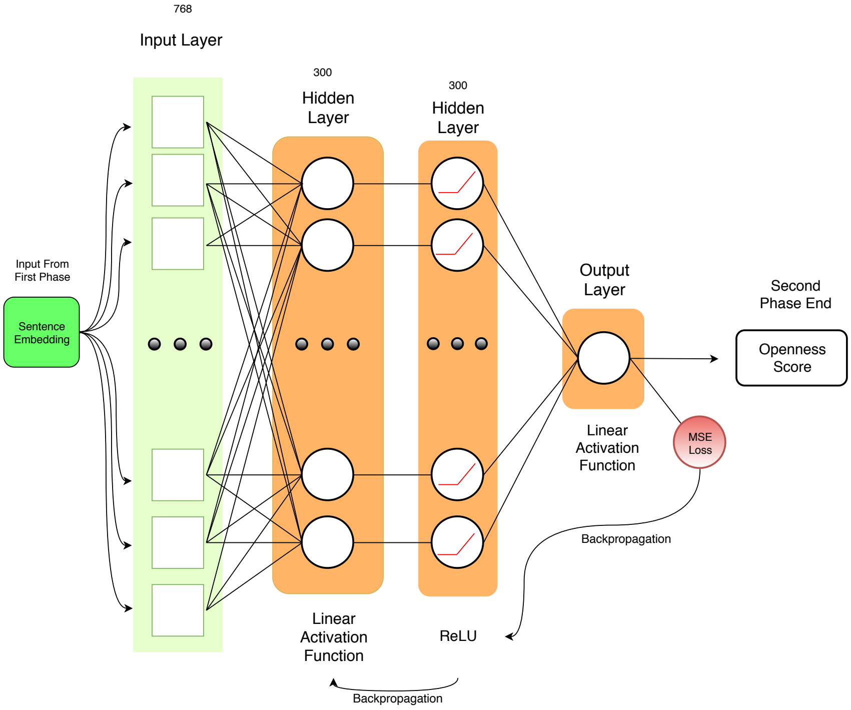


Fig. 3.5 Neural network that is stacked. Beginning with the sentence embedding from the encoding phase, we construct a stacked model with two hidden layers, each of which has a neuron with a Linear Activation function and a ReLu. Regression on the personality trait score is carried out in the output layer.

3.3.2 A stacked architecture

Our main contribution can be seen in the stacked neural network shown in Figure 3.5. It enables us to establish a new benchmark for personality trait inference from text. We significantly lower the mean squared error in trait prediction, and we also produce a smoother data distribution of the predicted scores, as demonstrated in Section 3.4. In fact, compared to earlier literature models [3, 4, 1], the scores at the tails of the data distribution are detected more accurately. The 768 dimensional CLS token from the encoding stage of our pipeline is fed into our stacked neural network, as shown in Figure 3.4. After that, we reduce the input's dimensionality to 300 dimensions using a linear activation function in a feed-forward layer. For each of the five personality traits (Openness, Consciousness, Extraversion, Agreeability, Neuroticism), we use the same architectural approach.

$$obj(\theta) = \frac{1}{2}\sum_i \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 = \frac{1}{2}\sum_i \left(\theta^\top x^{(i)} - y^{(i)}\right)^2 \tag{3.1}$$

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{3.2}$$

The analysis of Carducci et al. [3] and Landauer & Dumais [97] indicated that 300 features would be the ideal number to estimate personality traits. They discovered through empirical research that a word distribution is best represented by 300 features. We experimented with many different configurations for the number of neurons in this hidden layer, but when we tested with more than 300 neurons, neither the execution time nor the mean squared error improved. Equation 3.2's ReLu, which is an activation function in each neuron, is carried out by the subsequent layer. This decision enhances performance and expedites the learning process. Additionally, it helps to prevent vanishing gradient issues. When we back-propagate the error signal in a feed-forward network and it decreases/increases exponentially in relation to the distance from the final layer, we have a vanishing gradient. The regression is then carried out by a layer that consists of a single neuron and uses a linear activation function. The regression computes one personality trait $y = h(x)$. It

[5] https://github.com/hanxiao/bert-as-service

[6] http://commonsensereasoning.org/winograd.html

[7] https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Table 3.1 The list of parameters to configure the encoder of Figure 3.2.

Parameter	Value
hidden_size	768
num_hidden_layers	12
num_attention_heads	12
intermediate_size	3078 (768x4)
hidden_act	gelu
hidden_dropout_prob	0.1
attention_probs_dropout_prob	0.1
max_position_embedding	512

begins with the starting vector $x \in \mathscr{R}^n$, where $n = 768$. It uses a linear function $h_\theta(x) = \sum_j \theta_j x_j = \theta^\top x$ to represent $h(x)$, where $h_\theta(x)$ belongs to the linear function family parametrized by θ. We search the θ that minimizes the objective function in Equation 3.1.

3.3.3 Optimize the model

The configuration parameters for the *pooling_strategy* is the *CLS_token*. This choice sets the pooling_strategy to none, instead it creates a dummy token (CLS) at the beginning of the phrase.

In table 3.2, there is the list of parameters to optimize the architecture of the neural network. The choices made are the results of an empirical grid search experiment.

3.4 Results

We compare our model to earlier models, as well as to various configurations of the encoding and regression stages, in order to determine whether it advances the current state of the art. The myPersonality small dataset (9913 records) is used for the experiment.

Table 3.2 Parameters to configure Figure 3.5 neural network.

Parameter	**Value**
optimizer	**Adam**, Adagrad, SGD
learning rate	**1e-5**, 1e-2, 1e-7
loss function	Mean Squared Error Loss
batch size	**50**, 100, 200

Table 3.3 Using the myPersonality small dataset, the mean squared error was calculated by averaging the results of a 10-fold cross-validation. The best and worst results are those that are highlighted. Openness, Consciousness, Extraversion, Agreeability, and Neuroticism are all letters in the acronym OPE. The MSE (Mean squared Error) value is a measure of how well the outcome performed. In the case of IBM Personality Insights, when we used the raw text from myPersonality small as input and contacted their API, the response was that there was not enough text to make a prediction. Next, we made the decision to submit a query that collected all of the social media posts from a single user into a single block of text. The scores displayed in the table below suffer from a user-wise computation as opposed to a post-wise one. Our model is called SentencePersonality.

	Mean Squared Error (MSE)				
	OPE	CON	EXT	AGR	NEU
MTPTE Multilingual	**0.1759**	**0.3045**	**0.4750**	**0.2667**	**0.2911**
MTPTE	**0.2166**	**0.3556**	**0.5271**	**0.3117**	**0.3576**
FastText + Neural Network	0.3917	0.4824	0.6100	0.3643	0.5677
IBM Personality Insights	0.3769	0.5550	0.7483	0.4289	0.9303
Transformer + SVM	0.3867	0.5596	0.7579	0.5889	0.7240
Carducci et al. [3]	0.3316	0.5300	0.7084	0.4477	0.5572
Quercia et al. [4]	0.4761	0.5776	0.7744	0.6241	0.7225

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.3}$$

We contrast our findings in Table 3.3 in terms of mean squared error with the pre-existing baselines. As opposed to Mean Absolute Error (MAE), Mean squared Error (MSE) is more sensitive to not uniformly large variations in error. Light variations between predicted and actual scores must be accepted in the context of personality trait prediction. However, even in rare instances, we must quickly identify grave predictions that are off. Both the variance of the frequency distribution of error magnitudes and the variance of the errors themselves cause MSE to increase. We choose MSE because it allows us to compare our model to earlier ones in this field that also used MSE. Since none of the baselines used for comparisons provide R-squared data, we do not report it instead.

In Table 3.3, we compare the state of the art of Carducci et al. [3] with the results of Quercia et al. [4] and three different model configurations of our solution. The row labeled Transformer + SVM shows the MSE obtained from the encoding step in Figure 3.2, then using a support vector machine to perform the regression. In this case, none of the MSE obtained improved state-of-the-art technology. That is, only one coding step is required, but not enough to predict personality traits well. The FastText + Neural Network lines show the results, including the pretrained words from the FastText [8] word-level encoding engine. We only improved the final level on three of the five traits (CON, EXT, and AGR) because this scenario does not represent enough openness and neuroticism. These MSE scores show how well our model can exploit latent cues, including word-level text encoding, but more information is needed to exploit its full potential with all individual properties. Finally, improvements introduced by sentence-level attentional mechanisms in text encoding combined with neural networks can outperform state-of-the-art technologies in all five properties.

3.4.1 Baselines comparison

The histograms in Figure 3.6-3.10 show the distribution of personality trait data obtained from the myPersonality mini-survey, with predictions in the left figure,

[8] https://fasttext.cc/docs/en/english-vectors.html

middle figure, and Carducci et al. [3] shows the personality trait predictions obtained by the model. However, we use the MSE from Table 3.3 to verify the results. Always in the center column, our model results still follow a Gaussian shape, but they are more capable than the model by Carducci et al. [3] to represent marginal values at the far ends of the range $[1-5]$. To the left of each of the triplets is a histogram of the personality traits estimated by the personality test. These figures show how the actual values are expressed in the same way, but our automatic predictions in text analysis are still different shapes. However, the choices made were in the right direction towards more realistic calculations of personality traits.

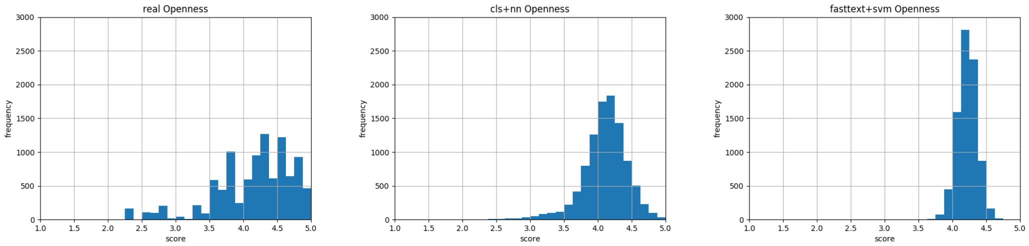


Fig. 3.6 Openness. Histograms representing data distribution of Gold Standard on the left, our model result in the center, previous state of the art by Carducci et al. [3] on the right.

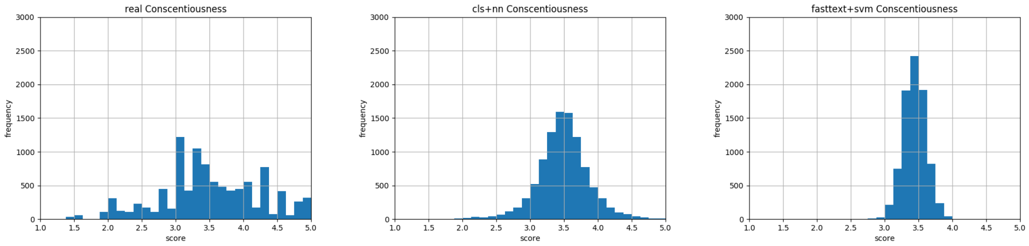


Fig. 3.7 Conscientiousness. Histograms representing data distribution of Gold Standard on the left, our model result in the center, previous state of the art by Carducci et al. [3] on the right.

$$D_{KL}(P||Q) = H(P,Q) - H(P) \tag{3.4}$$

$$H(P,Q) = \mathbf{E}_{x \sim P}[-log Q(x)] \tag{3.5}$$

$$H(P) = \mathbf{E}_{x \sim P}[-log P(x)] \tag{3.6}$$

$$D_{KL}(P||Q) = \mathbf{E}_{x \sim P}[log \frac{P(x)}{Q(x)}] \tag{3.7}$$

Table 3.4 Kullback Leibler divergence computed among probability distributions on Openness.

	Kullback Leibler divergence – OPENNESS			
	MTPTE	Transf. + SVM	Carducci et al. [3]	Real
MTPTE	0	1209.348	807.355	**36.159**
Transf. + SVM	-	0	25.65	1337.239
Carducci et al. [3]	-	-	0	1067.897
Real	-	-	-	0

Table 3.5 Kullback Leibler divergence computed among probability distributions on Conscientiousness.

	Kullback Leibler divergence – CONSCIENTIOUSNESS			
	MTPTE	Transf. + SVM	Carducci et al. [3]	Real
MTPTE	0	281.968	375.6	565.094
Transf. + SVM	-	0	79.327	**377.122**
Carducci et al. [3]	-	-	0	609.411
Real	-	-	-	0

Table 3.6 Kullback Leibler divergence computed among probability distributions on Extraversion.

	Kullback Leibler divergence – EXTRAVERSION			
	MTPTE	Transf. + SVM	Carducci et al. [3]	Real
MTPTE	0	689.846	318.312	**1019.066**
Transf. + SVM	-	0	465.049	1814.447
Carducci et al. [3]	-	-	0	1368.251
Real	-	-	-	0

Table 3.7 Kullback Leibler divergence computed among probability distributions on Agreeableness.

	Kullback Leibler divergence – AGREEABLENESS			
	MTPTE	Transf. + SVM	Carducci et al. [3]	Real
MTPTE	0	259.779	382.841	471.031
Transf. + SVM	-	0	255.15	891.557
Carducci et al. [3]	-	-	0	**266.071**
Real	-	-	-	0

Table 3.8 Kullback Leibler divergence computed among probability distributions on Neuroticism.

	Kullback Leibler divergence – NEUROTICISM			
	MTPTE	Transf. + SVM	Carducci et al. [3]	Real
MTPTE	0	378.843	572.621	**407.553**
Transf. + SVM	-	0	424.558	1130.947
Carducci et al. [3]	-	-	0	551.615
Real	-	-	-	0

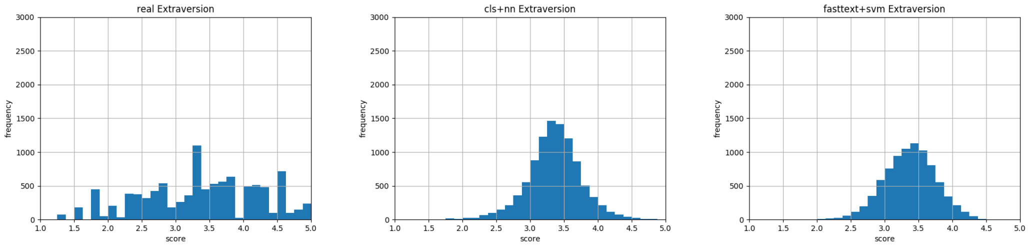


Fig. 3.8 Extraversion. Histograms representing data distribution of Gold Standard on the left, our model result in the center, previous state of the art by Carducci et al. [3] on the right.

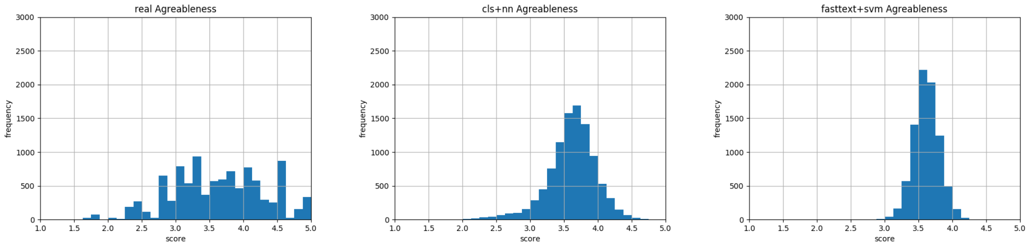


Fig. 3.9 Agreeableness. Histograms representing data distribution of Gold Standard on the left, our model result in the center, previous state of the art by Carducci et al. [3]on the right.

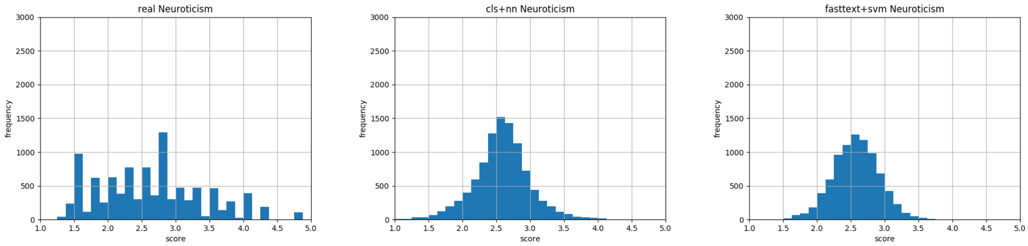


Fig. 3.10 Neuroticism. Histograms representing data distribution of Gold Standard on the left, our model result in the center, previous state of the art by Carducci et al. [3] on the right.

$$D_{KL}(P||Q) = \sum_{i} P(i) log \frac{P(i)}{Q(i)} \tag{3.8}$$

A probability distribution Q's Kullback-Leibler divergence [98], which is determined by Equation 3.4, is a gauge of how closely it resembles the real probability distribution P. In fact, we anticipate that personality traits gleaned from text will be as accurate as possible in predicting personality traits. In this sense, we must consider our results to be approximations of the actual ones. The Kullback-Leibler divergence is expressed as a discrete summation in Equation 3.8. The entropy of P, which is used in Equation 3.6, is defined there. A low entropy indicates that the data in P is highly predictable. Cross-entropy in Equation 3.5 calculates the difference between what the true distribution represents and how well it is approximated by the approximating distribution. According to the order indicated by Equation 3.4, we consider the first column in Tables 3.4-3.8 to be the Q probability distributions and the first line to be the P probability distributions. Because Kullback-Leibler is not commutative, we specify this. These findings demonstrate how our model performs the best in terms of approximating the distribution of the personality traits of neuroticism, extraversion, and openness that were obtained from questionnaires.

With these findings, we can respond to the initial research question. In fact, by developing a pipeline made of an encoder and a neural architecture, as shown in Figure 3.4, we were able to lower the error in the automatic personality trait assessment from written text that was previously present. So, in order to assess personality traits, sentence encoding with Transformer and deep learning are both effective. Figure 3.4 depicts the entire pipeline that was used to resolve this problem, including both the encoder phase and the regression phase.

Our second research question obtain an answer with our multilingual model.[9] The model only works with one language at a time, so it is crucial to note that it is not cross-lingual. The model is unable to carry out the regression correctly if the sentence to be processed contains, for instance, Spanish, French, and Chinese words mixed together. Sentences composed of words from one language at a time are used as input for the regression.

The myPersonality small described in Section 3.3 has been adopted as the dataset for the multilingual configuration.

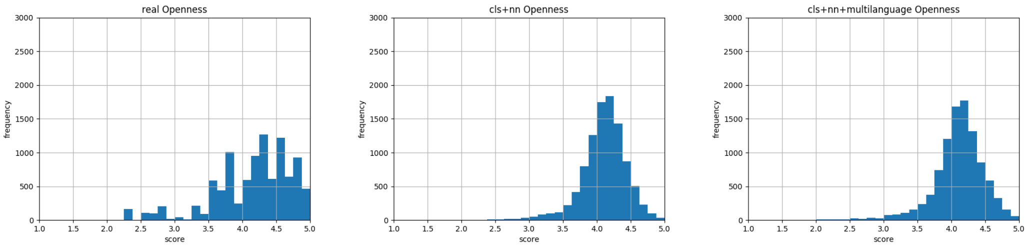


Fig. 3.11 Openness. Histograms representing data distribution of Gold Standard on the left, our English model result in the center, our multilingual model result on the right.

[9] https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip

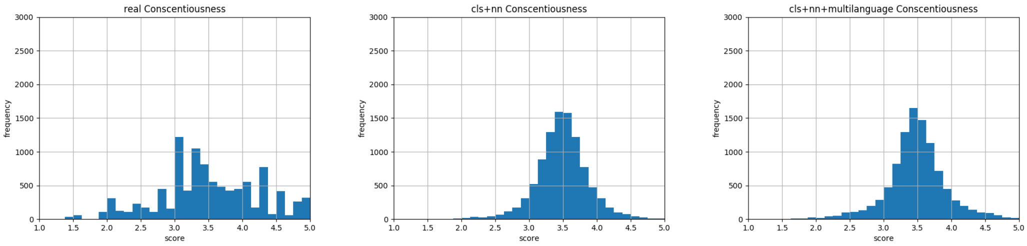


Fig. 3.12 Conscientiousness. Histograms representing data distribution of Gold Standard on the left, our English model result in the center, our multilingual model result on the right.

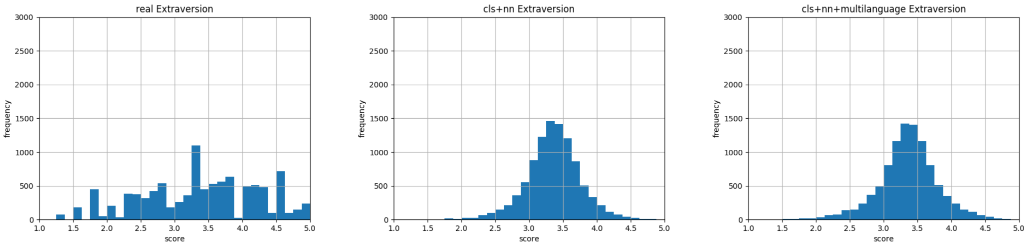


Fig. 3.13 Extraversion. Histograms representing data distribution of Gold Standard on the left, our English model result in the center, our multilingual model result on the right.

This multilingual model has been trained using 104 different languages.[10] The word embeddings of 104 different languages can be calculated using this multilingual model. We start with these embeddings and run the regression as per Section 3.3. The shape of the multilingual model for the predicted data distribution, as seen in Figures 3.11-3.15, resembles that of the model made with the predicted English distribution. We must look at the MSE in Table 3.3 and the Kullback-Leibler divergence between them in Table 3.9 in order to detect an actual improvement in the results obtained. The data show that by using a model that can comprehend 104 languages, we are able to more accurately approximate the data distribution of the myPersonality dataset and reduce the mean squared error. In conclusion, we chose a sentence embedding model that was trained in a multilingual environment and changed this component in the encoding stage of our pipeline 3.4 to respond to the second research question.

[10] https://github.com/google-research/bert/blob/master/multilingual.md#list-of-languages

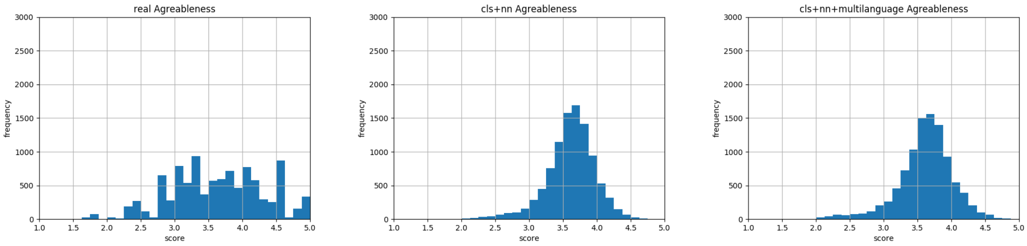


Fig. 3.14 Agreeableness. Histograms representing data distribution of Gold Standard on the left, our English model result in the center, our multilingual model result on the right.

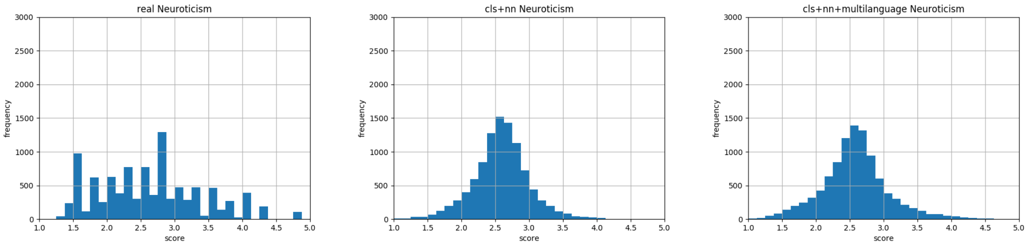


Fig. 3.15 Neuroticism. Histograms representing data distribution of Gold Standard on the left, our English model result in the center, our multilingual model result on the right.

3.5 Discussion

The mean squared error reported in Table 3.3 was used in Section 3.4 to demonstrate how well our results performed in comparison to the state-of-the-art at the time. Over each of the five personality traits, we saw an improvement of 30 percent on average. We dealt with the issue of prior systems' lack of discriminative power. Figures 3.6-3.10 show this result, which indicates that our model is more discriminative in this regard. These figures show a greater representation of scores at the tails of the range from 1 to 5. This accomplishment is further highlighted by the Kullback-Leibler divergence in Tables 3.4-3.8, which shows how effectively our model can approximate the real data distribution using predicted data. The remaining margin of

Table 3.9 Kullback Leibler divergence comparing English and multilingual results.

	Real OPE	Real CON	Real EXT	Real AGR	Real NEU
MTPTE Mult.	**34.71**	**543.10**	**878.87**	**381.82**	**255.51**
MTPTE Eng	36.15	565.09	1019.06	471.03	407.55

improvement is nevertheless also highlighted by the histograms. In fact, because the mean squared error reduction target tends to predict scores closer to the interval's center, our data distribution still largely retains a Gaussian shape. This evidence suggests that in order to more accurately describe and utilize the latent features, the neural network must be expanded or the encoding process modified. In order to create a more conducive environment for transfer learning, we only used the myPersonality dataset's raw text. Plain text social media posts are easier to access than other private and personal data. Our solution, which relies only on raw text, is effective, but more analysis with more features is required. Additionally, in order to remove the strict correlation of personality traits as fixed in time and to achieve the results obtained, we operated at the level of a single post rather than at the level of a user. This implies that depending on the content he writes, a single user will receive various personality trait scores. Due to the decision to work at the sentence level, another focus of our work is on avoiding a pooling system in the context of word embeddings. When we choose the CLS token rather than performing pooling among all of the word embeddings of the sentence under analysis, Figure 3.2 clarifies this passage. Additionally, when the pooling strategy is set to CLS_TOKEN, this option can be seen in our repository in the file *create_train_table_whole_lines.py*. This decision is crucial because it enables us to avoid information loss when averaging, adding, or concatenating word embeddings. This prevents the knowledge gathered from being impoverished and used to improve trait predictions during the regression phase. Through its IBM Personality Insights[11] product, IBM has made the assessment of personality traits available for commercial use. The Five Factor Model is one of the models adopted by IBM. Using their API, we compared their prediction of personality traits from the myPersonality dataset with ours, as shown in Table 3.3. The results from IBM Personality Insights are less favorable than ours, but because the API's word count restrictions force it to operate at the user level, it groups all of a user's social media posts into one query so that there is enough text to be input. According to IBM, the model they created requires at least several hundred words to calculate personality traits; otherwise, the predictions are not meaningful.

[11] https://www.ibm.com/watson/services/personality-insights/

3.6 Conclusion

We described a model to process social media posts, encoding each one at the sentence level into a high dimensional array space. To perform a regression on each of the five personality traits in the Big5 model (Openness, Extraversion, Agreeability, Consciousness, and Neuroticism), we processed this collection of 768 features using a stacked neural network. Although our model uses the same parameters for all five personality traits, it computes each one separately and adjusts the weights accordingly. In a controlled setting, we predicted expected personality traits based solely on the textual content of each social media post. These traits are represented as numbers in the continuous range 1 to 5. The myPersonality gold standard, which provided the target characteristics. In comparison to the current state of the art, the obtained results reduced the mean squared error. The Kullback-Leibler data distribution divergence also supports them. These findings were obtained using a 104-language multilingual model, which was then replicated using only the English text. A repository open to the public contains the code we created.[12] We will conduct additional research in this area in addition to improving the model and testing it using various data sources, including the PAN-AP-15 datasets [99] and [100]. When used in conjunction with conversational agents and virtual assistants, we want to evaluate the effectiveness of our work. We will perform the experiment over a longer period of time to track and analyze whether and how the personality traits of experiment candidates change over time. Based on characteristics of human personality, we will adjust conversational agents' responses. In order to enhance the digital experience in terms of recommendations and empathy, we want to compute them in real time during the conversation. This strategy aims to promote creativity and serendipity. Additionally, we are researching how much influence personality traits have on the creation of viral content and how to use it to spread accurate scientific news and encourage good behavior. In order to improve the objectivity of the facts against the author's personal opinion, we want to develop a tool to remove emotions and personality traits from newspaper articles. Our goal is to create conversational agents that are empathetic and learn alongside the user, becoming proactive in the conversation rather than merely passive in question-answering or command-execution. In this instance, the agent develops its behavior by observing the user it is assisting.

[12] https://github.com/D2KLab/SentencePersonality

Chapter 4

Mining micro-influencers from social media posts

The state-of-the-art model presented in Chapter 3 opens a new perspective on the computation of personality features from text. A gold mine for unstructured written data is available on social media. The analysis of language on these media is a natural application of our findings. In particular we want to use the language market to determine users' characteristics. Giving the fact that only five features are not representative at all of the complexity of human actions, this chapter introduces another sociological analysis tool from language by Schwartz [101] and a list of social graph metrics. This chapter describes what is the impact of a software that determines the capacity of a user to influence other people making them perform certain actions or change point of views. The developed models is able to detect this particular kind of users based on what they write and so their personality traits, basic human values and social media graphs.

These users make viral content spread across the world. Although the majority of them are thought to be useless or junk, they can also be used to spread goodwill. People with influence power have the ability to infect large segments of the population and encourage socially accepted behaviors.

Influencers in literature are described as having influence and using social media platforms. They have a propensity for influencing others to alter their opinions and behaviors. [102] In order to identify influencers, current studies primarily examine how well their posts are known and covered by the media. These techniques aim to identify candidates using the scores provided by social media platforms by gauging the response of followers to particular posts (i. e. Graphs from social media are also examined [33, 31, 30] (comments, sharing). However, they fail to take into account how the psychological aspects of the influence mechanism.

This study suggests a method for automatically identifying micro-influencers and highlighting their character traits and shared values through computational analysis of their writings. Micro-influencers are a particular class of influencers that are harder to find, less well-known, but have greater engagement power over their communities [103]. The procedure begins by gathering user writings on popular subjects. The micro-influencers gold standard dataset is created using a rule-based methodology after a filtering and processing stage.

The lexical hypothesis [104] can be used to calculate personality traits and community-based values scores on micro-influencer writings because words are represented in vector space as embeddings. When words are analyzed at the word level, micro-influencer traits are revealed in what users post on social media. With the help of the research done by Schwartz et al. [101], we demonstrate the potential of approaching the influencing mechanism on a community level. Through the work of McCrae and Costa [92] applied to the study of micro-influencing. These methods' predictions for the Five Factor Model and the Basic Human Values scores are used as feature vectors in a SVM classifier.

Three research questions have been established:

- RQ1: What is the gold standard for finding micro-influencers among social media posts?
- RQ2: How can I determine a person's personality and values based on their community from a text?
- RQ3: How can a feature vector made up of personality traits and community-based scores be used to categorize micro-influencers?

The remainder of this chapter is organized as follows: Section 4.1 introduces Basic Human Values, while Section 4.2 outlines the suggested method for identifying micro-influencers. Section 4.3 provides a preliminary experimental assessment of the proposed method. Section 4.4 discusses the outcomes and advancements. A summary of the experiment's findings and a description of future research are provided in Section 4.5, where we also draw a conclusion.

4.1 Basic Human Values

The theory of Basic Human Values (BHV) [101] identifies ten motivating factors. It defines the dynamics and specifies distinct value orientations that people from all cultures can recognize. It seeks to be the field's guiding theory. A method of organizing the various needs, drives, and objectives put forth by different theories.

Each of the ten fundamental principles can be sum up by stating its primary motivational objective:

- Self-Direction. Choosing, creating, and exploring on one's own.
- Stimulation. Life's challenges, novelty, and excitement.
- Hedonism. For one's own enjoyment and sensual gratification.
- Achievement. Success on a personal level by proving one's ability over social expectations.
- Power. Control or dominance over people and resources, as well as social status and prestige.
- Security. Stability, harmony, and safety in relationships, society, and oneself.
- Conformity. Control over behaviors, inclinations, and urges that could upset or hurt someone others and transgress social conventions.
- Tradition. Commitment to and acceptance of the traditions and beliefs that the self is provided by traditional culture or religion.
- Benevolence. Preserving and improving the welfare of the people one is with frequent face-to-face interactions.
- Universalism. Understanding, admiration, tolerance, and welfare protection. of all people and for the environment.

Groups and individuals represent these requirements cognitively (linguistically) as particular values that they communicate about in order to coordinate with others in the pursuit of the goals that are important to them. Three universal requirements of the human condition needs of individuals as biological organisms, requirements of coordinated social interaction, and survival and welfare needs of groups have been translated into ten motivationally distinct, broad, and fundamental values.

4.2 Detecting Micro Influencers

The rules used to create the micro-influencer gold standard definition are formalized in the sections that follow. In the following section, we'll go over how, in the specific case of the micro-influencer field, we used the lexical hypothesis to define an algorithm that pulls personality traits and values from user writings.

4.2.1 Micro influencers gold standard and dataset creation

In the area of micro-influencers, we lay out the guidelines for producing a dataset with annotations that will serve as the industry's gold standard. We handle the entire pipeline, including data retrieval, label definition using our oracle, and final dataset creation. [1] The following is a description of the rules established for the data that was collected. Candidates with follower counts between 1,000 and 100,000 were subject to these rules. The labels assigned by our oracle as micro-influencers in the gold standard are explained by these scores. We calculate the average Embeddedness of all potential micro-influencers. The same is true for interaction and engagement. These averages serve as a boundary. A potential micro-influencer is classified as one if each of the three scores exceeds the relative threshold. Interaction, engagement, and embeddedness all carry equal weight.

Easley et al.'s embeddedness score [35] is derived from them. There, the authors discuss neighborhood overlap in a community where at least two influential members discuss related ideas, information spreads quickly. Two potential micro-influencers are considered to be part of the same community if they have nearly identical numbers of followers. When comparing the followings of two micro-influencers

[1] Visit https://doi.org/10.6084/m9.figshare.11309669 to access the dataset that includes source tweets, associated computed scores, and our gold standard.

discussing the same subject (i.e. Artificial Intelligence), we see that many of the first micro-influencer's followers also appear as followers of the second. This is equivalent to having more than two micro-influencers write about the same subject.

We reward a micro-influencer if his followers show up on many lists of followers of the other micro-influencers as a result. When compared to the total number of micro-influencers in a given area, a larger audience split indicates greater influence capability. In addition, numerous other micro-influencers who are discussing the same subject and sharing his followers will amplify the message that this micro-influencer writes.

$$Embeddedness_i = \frac{\sum_j |\{Follower_i\} \cap \{Follower_j\}|}{NumberOfFollowers_i} \tag{4.1}$$

The potential micro-influencer we are analyzing is denoted by i in the previous equation, and the other potential micro-influencers on our list are denoted by j.

Equation 4.2 calculates the interaction score. We decide how many retweets each tweet receives from followers t for each tweet. We are motivated by the Kittl and Anger Interactor Ratio [30]. The fact that people who can retweet but do not follow the potential micro-influencer are not taken into account is significant.

A micro-influencer needs to make his followers more devoted. He must engage with them, and in return, his supporters will express his views. We have our attention on his adherents. A micro-influencer can persuade sporadic users as well, but we do not take them into account because we want to know whether his regular circle of followers engages with him.

$$Interaction_i = \frac{\sum_t RetweetByFollower_{i,t}}{TotalFollowers_i \times NumberOfTweets_i} \tag{4.2}$$

Equation 4.2 normalizes the score based on the overall number of tweets, as a micro-influencer with a large number of previous tweets has an advantage over newly arrived micro-influencers.

The Grin tool [2] has modified the engagement score. Engagement is calculated by adding the likes and retweets and dividing that total by the followers of the potential

[2] https://www.grin.co

User	Embeddedness	Interaction	Engagement	Micro influencers
0	0.487231	0.0	0.000937	0
1	0.734960	0.0	0.000616	0
2	1.973224	0.833333	0.001889	1
3	0.808098	1.171428	0.063291	1
4	0.325655	0.0	0.000311	0

Table 4.1 Gold Standard illustration results. The first column lists the users who have been made anonymous using an auto-incremental integer number. The actual scores calculated for those users are shown in the following three columns. The sixth decimal place is used to round up scores. The user's label is displayed in the last column, with 1 denoting a micro-influencer and 0 denoting a non micro-influencer.

micro-influencer. This result is then divided by the total number of tweets they have published.

$$Engagement_i = \frac{\sum_t (Likes_{i,t} + Retweets_{i,t})}{TotalFollowers_i \times NumberOfTweets_i} \tag{4.3}$$

This rating represents a micro-influencer's overall capacity for information dissemination. A message will spread both inside and outside of his community more quickly the higher the engagement as measured in Equation 4.3.

In Table 4.1, a portion of the gold standard dataset is displayed. The first column of the table contains an auto-incremental integer number that has been used to make the users anonymous. The next three columns display actual scores that were computed using the rule that was previously discussed in this section. The user's label is displayed in the last column, with 1 denoting a micro-influencer and 0 denoting a non micro-influencer. The three scores (Embeddedness, Interaction, and Engagement) and their respective thresholds for defining a gold standard in the micro-influencer field are used in this section to respond to the RQ1.

4.2.2 Mining basic human values and personality traits from text

4.2 Knowing whether micro-influencers use recurrent lexical expressions in relation to particular topics is important when working with computational linguistics. From GloVe, we employ a pre-trained word embeddings model. With regard to Twitter, GloVe has a refined model.[3] GloVe performs better with concise text structures. As a result, the word representation as expressed by word embeddings is appropriate for our context. Short sentences with lots of domain-based tokens, like hashtags and handles, are what make a tweet a tweet. If not, the pre-training phase's effectiveness in defining word embeddings will be diminished. Therefore, we need a model that has been trained taking this context into account. If not, the pre-training phase's effectiveness in defining word embeddings will be diminished. Therefore, we need a model that has been trained taking this context into account. Both the BHV (Basic Human Values) and FFM (Five Factor Model) scores are computed using an algorithm that we develop. Examples of words for each human value are provided in Basic Human Values Research [101]: *selfdirection*, *stimulation*, *hedonism*, *achievement*, *power*, *security*, *conformity*, *tradition*, *benevolence*, *universalism*. By averaging word embeddings provided as examples, we produce a centroid for each Basic Human Value. As an illustration, the following words are listed for benevolence: *helpful*, *honest*, *forgiving*, *responsible*, *loyal*, *friendship*, *love*, *meaningful*. In the GloVe pre-trained vocabulary, each of these words has a 300 dimensions array representation. The first dimension of *helpuful* is summed with the first dimension of *honest*, *forgiving*, *responsible*, *loyal*, *friendship*, *love* and *meaningful*. The total is then divided by seven. For the following 299 dimensions, this procedure is repeated. At the conclusion, we obtain the centroid of the benevolence community value's 300 coordinates that were computed. 10 centroids—one for each community-based value—remain after this phase.

Following this configuration phase, as depicted in Figure 4.1, each word written by the micro-influencer candidate is parsed one at a time.

The word is then assigned to the centroid that is closest to it in terms of euclidean distance after we calculate the distances between it and each community-based reference centroid. Each Basic Human Value's word count is calculated by multiplying the total number of words used in it by the inverse of the distance between its centroid and the centroid produced by averaging the spatial representation of all the micro-influencer words related to it. A score per BHV per micro-influencer is what we end up with.

$$\min\left(d(p,q_i)\right), i \in SchwartzSampleCentroids \tag{4.4}$$

[3] http://nlp.stanford.edu/data/glove.twitter.27B.zip

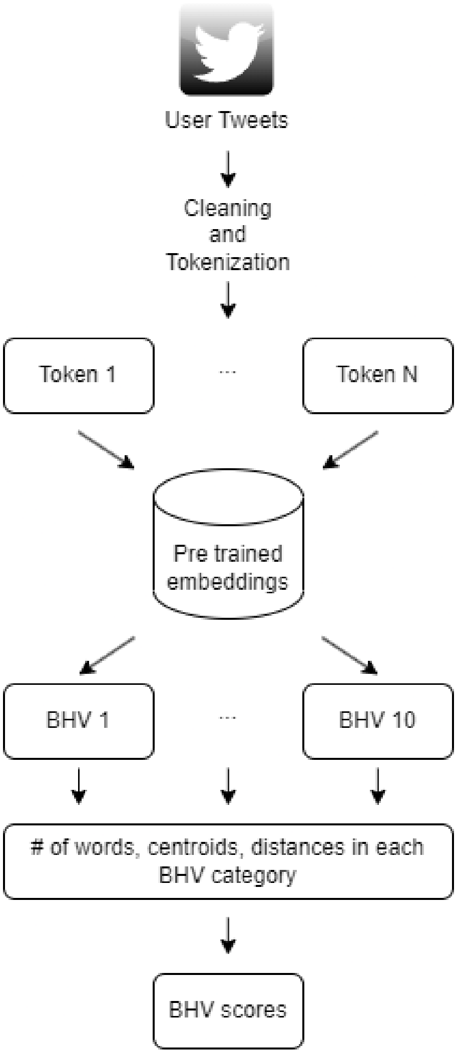

User Tweets
Cleaning and Tokenization
Token 1
...
Token N
Pre trained embeddings
BHV 1
...
BHV 10
# of words, centroids, distances in each BHV category
BHV scores

Fig. 4.1 BHV score prediction pipeline. User tweets are cleaned and tagged, and each token has a unique 300-dimensional matrix representation of GloVe embeddings. Each token in the matrix is mapped to the nearest reference center of the underlying human value. Then add the number of words in each group and multiply by the reciprocal of the average of those words. This course earns you 10 Community Points per user.

sd	st	he	ac	po	se	co	tr	be	un
634.1	860.4	68.6	494.7	91.9	3290.3	527.8	28.1	836.1	476.8
520.1	594.8	72.7	347.4	88.3	3387.6	580.3	23.1	761.8	410.1
542.9	747.5	73.5	418.9	89.5	2754.4	553.6	39.6	878.7	585.9
255.6	256.7	4.7	156.1	14.8	955.3	281.8	9.83	238.8	166.7
796.4	569.9	90.6	202.5	145.4	3739.6	474.9	31.9	958.8	438.5

Table 4.2 Community based scores. These five rows represents five randomly selected users in our collction that are all potential micro-influencers. They remain after the first filter on the number of followers, as described in Figure 3.4. The ten columns of this table show the final score obtained after user tweet corpus analysis using the procedures described in Section 4.2. self direction, stimulation, hedonism, achievement, power, security, conformity, tradition, benevolence, universalism.

$$SS_i = nw_i * \left(\frac{1}{d(avgEmb_i, Schwartz_i)} \right) \tag{4.5}$$

Equation 4.4 determines the minimum euclidean distance between a word under analysis and the centroids of the Schwartz example. The closest cluster of Basic Human Values is then given the word's placement.

Equation 4.5 is used to calculate each community's final score based on user values. We multiply the total number of words used in a community-based value by the inverse of the euclidean distance between the example centroid and the averaged centroid of the words used in that semantic cluster. Ten BHV scores are finally obtained for each user. Table 4.2 displays the first subset of the dataset, which contains the BHV ratings of the users who were subjected to analysis.

The Five Factor Model scores are retrieved by modifying a Carducci et al. [3] method. This approach works with the myPersonality dataset[4], which combines social media posts from the same users who responded to the questionnaire with FFM score obtained from a psychological questionnaire. We develop a Support Vector Machine algorithm to retrieve the traits of writings pertaining to psychological outputs for each of the FFM dimensions. The Support Vector Machine regression algorithm's C and γ parameters are chosen using a grid search. This method evaluates various C and γ values to determine which one improves the performance of the regression. By changing the C parameter, we can change the types of samples that are

[4] https://sites.google.com/michalkosinski.com/mypersonality

taken into account. For example, low values of C consider almost all of the samples in the dataset, whereas high values only take into account samples that are close to the hyperplane's edge. γ is another crucial parameter because it affects how flexible or rigid the hyperplane is because it affects the kernel as well. Overfitting may take place if the gamma value is excessively high. Each trait in the Five Factor Model has its own regression model. Without the aid of a psychological questionnaire, we use them to forecast the FFM personality traits of a micro-influencer based solely on what he writes.

With the definition and use of the two earlier techniques for the prediction of Five Factor Model and Basic Human Values scores, we respond to our RQ2. The text of micro-influencers' social media posts is actually used to extrapolate personality traits and values based on the community.

Prediction pipeline for BHV scores. User tweets are cleaned up and tokenized, and each token has a 300-dimensional array representation derived from GloVe embeddings. Each token array-shaped is assigned to the closest centroid of the Basic Human Values reference. The total number of words in each cluster is then multiplied by the inverse of the average position of these words. With this method, each user receives ten community-based scores.

4.3 Results

We use the social media platform Twitter to retrieve user writings. In our setting, we choose Twitter users who have recently posted on hot topics. Users with fewer than 1,000 followers and more than 100,000 followers are deleted. A user is no longer viewed as a micro-influencer once he or she leaves this audience size range because of the size of his or her following [103]. We download both the list of followers' ids and the user's tweet text corpus for each user. For data cleaning, we employ the Natural Language Toolkit.[5] As a first step, we save tweets as a tsv (tab separated value) file right after the downloading phase, stripping the original text of all newlines and tabulations. The subsequent steps are used in the second phase.

- stop-word removal: ”Uhm, where is the leader? @johnsmith #officelife. :)” to ”, where leader? @johnsmith #officelife. :)”

[5] https://www.nltk.org

- punctuation removal: ",where leader? @johnsmith #officelife :)" to "where leader @johnsmith #officelife :)"
- emoticon removal: "where leader @johnsmith #officelife :)" to "where leader @johnsmith #officelife"
- handle and url removal: "where leader @johnsmith #officelife http:/.../" to "where leader #officelife"

Hashtags (#) are maintained to emphasize the topics in each processed tweet. After text has been cleaned, it is tokenized at a space-based level, and each token that has been analyzed is searched in the corresponding pre-trained and fine-tuned embeddings vocabulary.

For the FFM and BHV models described in Section, where we demonstrate how to calculate them, we provide tokenized and vectorized words as input. Five scores are generated for the FFM (Five Factor Model) and ten scores are generated for the BHV (Basic Human Values) as a result of this process. The three supervised classifiers, SVM, Random Forest, and CNN, as well as an ensemble model, XGBoost, take the FFM and BHV scores as inputs, and the previously computed micro-influencer labels as expected outputs.

A hyperplane that maximizes the difference between two classes (in this case, micro-influencer and not-micro-influencer) is created using SVM in supervised learning. Equation 4.6 demonstrates how the Support Vector Machine aims to maximize the distance between the micro and not micro-influencer categories.

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b \tag{4.6}$$

$$K(x,x') = exp(-\frac{||x-x'||^2}{2\varsigma^2}) \tag{4.7}$$

We select a RBF (Radial Basis Function) as the kernel (Equation 4.7), where x and x' stand in for two model features. The RBF kernel is a mathematical technique for learning a non-linear classification rule that, for the transformed data points, corresponds to a linear classification rule. In order to distinguish the classes, the computation is carried out in a higher-dimensional space. After that, it is projected into a lower dimension to reveal the transformed function.

The Random Forest ensemble method, in contrast, builds a large number of decision trees and outputs the class that represents the mean prediction of all the trees combined, or the mode of the classes. Two categories apply in our situation: micro-influencer or not. We use 10 BHV scores and 5 FFM scores to calculate the $max_feature$ parameter[6], which is 15. To handle the unbalanced dataset, we employ a class-weight of 1:10. We prioritize the micro-influencer class more.

We test Convolutional Neural Network (CNN) as a classifier using the user's BHV and FFM scores as input, and the micro-influencer label as the expected output. With two layers of a sequential model built in Keras, *Adam* is used as an algorithm to optimize the adaptive learning rate. We use *relu* as the activation function. $Cross-entropy$ has been used as the loss function.

We employ XGBoost[7], a distributed gradient boosting library that has been optimized, to boost the classification performance. XGBoost combines various models using an iterative process. They receive training in succession, which minimizes mistakes made in earlier stages.

$$\begin{aligned}\text{obj}^{(t)} &= \sum_{i=1}^{n}(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^{t}\Omega(f_i) = \\ &= \sum_{i=1}^{n}[2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant \qquad (4.8)\end{aligned}$$

The XGBoost objective function is formalized in Equation 4.8. Trees begin to be pruned backward after XGBoost makes splits up to the $max-depth$. The computational efficiency of this depth-first method is increased. In order to avoid overfitting, it penalizes more complicated models using LASSO and Ridge regularization. XGBoost handles various types of sparsity patterns in the data more effectively and naturally accepts sparse features for inputs by automatically learning the best missing value based on training loss. After examining the outcomes in terms of validation metrics, as shown in Table 4.6, we ultimately decide on SVM. We assess the classifier's performance in identifying micro-influencers using validation metrics such as recall, precision, and F1-score. The following gives a description of validation metrics.

[6] The number of features to consider when looking for the best split.

[7] https://xgboost.readthedocs.io/en/latest/

Recall highlights the model's economic vulnerability because our tool identifies micro-influencers among a large number of non-micro-influencer users; as a result, if the number of *false negatives* is high, it produces few reliable results. So that there are few *false negative* results, recall should be as close to one as possible. Precision reveals the existence of users who are thought of as micro-influencers but are not acting as desired. The *F1-score* is the harmonic mean between *recall* and *precision*; it is helpful to understand the compromise that results in not much effort being expended manually removing the micro-influencer incorrectly predicted while, on the other side, not leaving out too many genuine positive results.

The number of candidates who are correctly classified in relation to the anticipated result is calculated using validation metrics in the binary case, as shown in the following. The quantity of correctly identified micro-influencers is measured by the *tp* (true positive) indicator. *Tn* (true negative) is a measure of the number of incorrectly categorized micro-influencers. *False positives* (fp) are the number of non-microinfluencers who were mistakenly identified as such. The number of microinfluencers labeled as not being microinfluencers is represented by the *false negative* (fn).

$$recall = \frac{tp}{tp + fn} \tag{4.9}$$

$$precision = \frac{tp}{tp + fp} \tag{4.10}$$

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{4.11}$$

Recall, *precision*, and *f1-score* are computed using *stratified k-fold*.

Support Vector Machine, Random Forest Classifier, CNN, and XGBoost are used to perform stratified k-fold.

In a cross-validation procedure known as stratified k-fold, the entire dataset is divided into k subsets, the classifier is trained on the first fold, and the outputs are predicted using the second fold. We decide on $k = 10$ and store validation metrics after each iteration. To get the scores shown in Table 4.6, we average all the validation metrics at the end of the process.

We employ stratified k-fold because it enables us to keep track of micro-influencer samples throughout the fold. Actually, it becomes clear from the statistics in Table

4.3 that the classes are seriously out of balance, with relatively few micro-influencers in relation to the total number of users retrieved.

The statistics in Table 4.3 reflect one of the main challenges in this process. One over ten or less percent of the total users analysed are micro-influencers. In contrast to well-known influencers, it is more difficult to find these users, as discussed in the beginning of this chapter. In our preliminary experiment, we used a half million tweets per topic, but only a small portion of them are relevant to the topic used in the first Twitter query when we search for micro-influencers who are talking about a trending topic.

These two factors emphasize the necessity of investing a lot of time and energy in the initial stages, especially when our dataset is small and all newly searched users are noisy. Our system will require ongoing fine-tuning and updates in the future as it manages a set of cases that are both larger and more important than the initial scenario and for which there are scant data.

The outcomes of our classifier, as shown in Table 4.6, are also reflective of this initial effort. As desired in our specific context, this table shows the best recall score. SVM outperforms the other three approaches in terms of overall performance. Recall quantifies the proportion of users who, despite being micro-influencers, are not counted in the specific context of micro-influencer detection. Very few micro-influencers can escape detection if the metric is close to 1. It's crucial to identify every micro-influencer because they can be hard to find.

Offgrid topic produceS better results because it has a higher percentage of micro-influencers than other topics do. SVM recall is always greater than that of other classifiers. When it comes to *biodynamic* and *greenliving*, CNN performs poorly. This is a result of a dearth of micro-influencer cases. It becomes apparent from looking at the precision score how many people who are not micro-influencers are occasionally given that designation. This situation results from the stratified k-fold's $class_weight$ parameter being set to 1:10 due to the unbalanced presence of a small number of positive cases in comparison to the total number of users under analysis. Finally, XGBoost ensemble methods are not helpful in our environment because they favor precision over recall.

Our RQ3 is addressed by the classification of micro-influencers using an SVM classifier and a feature vector made up of personality traits and community-based scores.

Topic	Number of users	Micro influencers	Total tweets per topic
offgrid	146	10.96 %	407,957
plasticfree	190	6.32 %	560,655
biodynamic	70	8.57 %	201,010
greenliving	153	5.23 %	454,312
womenintech	238	3.78 %	644,772
sustainable	219	5.02 %	658,854

Table 4.3 statistics pertaining to the specified micro-influencer gold standard dataset. Topic-based user groups are created using hashtag searches. The disparity between the total number of users retrieved and the most effective micro-influencer is highlighted by the second and third columns. The second decimal place is used to round percentages. The final column displays the total number of tweets per topic that were found by adding all user tweets that were analyzed.

User	selfdirection	stimulation	hedonism	achievement	power	security	conformity	tradition	benevolence	universalism
0	634.143	860.415	68.613	494.715	91.957	3290.369	527.879	28.087	836.087	476.888
1	520.099	594.802	72.731	347.464	88.334	3387.620	580.317	23.104	761.851	410.143
2	542.946	747.503	73.518	418.922	89.590	2754.447	553.618	39.676	878.727	585.977
3	255.612	256.765	4.789	156.130	14.810	955.365	281.866	9.873	238.837	166.703
4	796.353	569.985	90.697	202.510	145.414	3739.612	474.904	31.913	958.819	438.540

Table 4.4 Scores based on the community. Following the initial filter on the number of followers, as shown in Figure 3.4, these users are all potential micro-influencers. In the first column, the user's anonymized identifier is reported. After analyzing user tweet corpora using the methods outlined in Section 4.2, the results are displayed in the next ten columns.

User	Openness	Conscentiousness	Extraversion	Agreableness	Neuroticism
0	4.208	3.683	3.176	3.953	2.608
1	4.149	3.638	3.103	3.966	2.608
2	4.219	3.732	3.331	4.037	2.608
3	4.154	3.758	3.291	4.011	2.608
4	4.197	3.826	3.292	4.071	2.608

Table 4.5 scores on personality traits. According to Figure 3.4, all of these users are potential micro-influencers who were not eliminated by the first filter based on the quantity of followers. The first column contains the user's anonymized identifier. The final score after analyzing the user's tweet corpus in accordance with Section 4.2's procedures is shown in the next five columns.

	SVM			Random Forest			CNN			XGBoost		
Topic	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
offgrid	0.42	1.00	0.58	0.49	0.44	0.50	0.63	0.31	0.42	0.50	0.57	0.52
plasticfree	0.38	0.75	0.50	0.48	0.30	0.30	0.40	0.29	0.34	0.31	0.31	0.31
biodynamic	0.40	0.60	0.47	0.01	0.01	0.01	0.01	0.01	0.01	0.17	0.27	0.20
greenliving	0.26	0.50	0.32	0.50	0.13	0.20	0.01	0.01	0.01	0.23	0.13	0.17
womenintech	0.21	0.55	0.30	0.27	0.05	0.07	0.31	0.09	0.14	0.14	0.18	0.14
sustainable	0.23	0.47	0.30	0.48	0.11	0.14	0.6	0.27	0.38	0.13	0.16	0.13

Table 4.6 Comparing the classifiers under consideration experimentally. As desired in our specific context, this table shows the best recall score. SVM outperforms the other three approaches in terms of performance overall.

4.4 Discussion

Psychologists use the Five Factor Model and the Basic Human Values to calculate personality traits and social values. Then, we used them as features in the extraction of these scores from text. The first model is focused on a single individual's characteristics, and the second model is focused on a larger community level. These models are chosen as being representative in the influence mechanism among social media users. According to our theory, a micro-influencer needs to have unique personality traits and share the same values as the neighborhood he lives in.

Anyhow, as noted in Carducci et al. [3], the automatic classification of personality traits introduces a preliminary error in the computation of the features. Additional errors are introduced by the automatic classification of community values. We intend to investigate a validation dataset generated using questionnaires to try to reduce them.

At the same time, a consensus definition of micro-influencers must result from a longer-term experiment in which candidates are tracked and their performances are dynamically scored. However, since we start from the premise that the gold standard's rules already serve as validation, no human validation process is necessary to categorize a user as a micro-influencer. These arguments are helpful in understanding how to properly interpret our findings and in demonstrating that the research should concentrate on making these procedures better.

Word embeddings available in the released dataset[8] origins from two different sources: FastText[9] for Five Factor Model and GloVe[10] for Basic Human Values. These choices are motivated by experimental trails.

Micro-influencer classification lacks historical comparison points for baselines. In order to compare various classifiers, we used metrics for standard quality like precision, recall, and f1. From these findings, additional research should keep an eye on advancements. Our research suggests using NLP techniques for in-depth social analysis in a previously untapped user niche of micro-influencers.

We demonstrated in this paper how to identify micro-influencers and how to draw attention to their personality traits at the individual (FFM) and community (BHV) levels by examining their writings. To retrieve fresh data on various topics, the entire procedure can be repeated. It is also possible to extract FFM and BHV scores from various text sources by modifying pre-trained word embeddings and fine-tuning them in other fields of interest.

Our top-scoring model has excellent recall, but we still need to work on sharpening its accuracy, perhaps by locating more instances of micro-influencers to better train the classifiers.

Future research in this area may use updated computational linguistics algorithms and may take into consideration revised definitions of micro-influencers that are now widely accepted. We eliminated emoticons from the data during data cleaning, as explained in Section 4.2. Therefore, more research is needed to examine the information loss caused by this procedure and the micro-influence that emoticons have.

Finally, more information in this area relates to the analysis of audio and video features to draw attention to other characteristics important for comprehending the process of micro-influence.

[8] https://doi.org/10.6084/m9.figshare.11309669

[9] https://fasttext.cc/docs/en/english-vectors.html

[10] http://nlp.stanford.edu/data/glove.6B.zip

4.5 Conclusion

The method for locating micro-influencers is demonstrated in this chapter and how to draw attention to their personality traits on an individual basis (FFM) and community (BHV) level examination of their writings. It is possible to repeat the process to retrieve fresh data on various subjects. Additionally, FFM and BHV scores are enabled by modifying pre-trained word embeddings and perfecting them in other fields of interest. Moreover, FFM and BHV scores are made possible by altering pre-trained word embeddings and perfecting them in other areas of interest. Furthermore, FFM and BHV scores are made possible by altering pre-trained word embeddings and perfecting them in other areas of interest. In addition, the ability to modify pre-trained word embeddings and improve them in other areas of interest allows for FFM and BHV scores. Also, the ability to modify pre-trained word embeddings and improve them in other areas of interest enables the calculation of FFM and BHV scores. Furthermore, FFM and BHV scores are made possible by altering pre-trained word embeddings and perfecting them in other areas of interest. The extraction from various text sources. High recall scores are displayed by our best model, but we still need to sharpen it up, perhaps by finding more examples.

Chapter 5

MIMIC: a Multi Input Micro-Influencers Classifier

The findings obtained in Chapter 4 highlighted the opportunity given by the analysis of micro influencers. In that work, we explored only a subset of the features involved such as personality traits, basic human values and few social media graph metrics. In the published paper presented in this chapter, we will describe a broader and deeper assessment of micro-influencer characteristics. Our new model exploits the power of a transformer based neural network both with textual and visual input data.

Due to their ability to generate intense audience interest in a particular topic, micro-influencers are valuable components in the marketing strategies of businesses and institutions. In recent years, this type of social media user detection has been handled by numerous scientific approaches and commercial tools. These strategies use techniques like deep neural networks, rule-based machine learning models, and graph analysis on text, images, and account information. In order to generalize them across various input data sets and social media platforms, this work compares the solutions already in use and suggests an ensemble method. The implemented solution combines statistical machine learning models on structured data and deep learning models on unstructured data. On Twitter and Instagram, we retrieve both posts with multimedia and information about social media accounts. To create feature vectors for an eXtreme Gradient Boosting (XGBoost) classifier, these data are mapped. In order to compare the performance of our approach to baseline classifiers and build a rule-based gold standard dataset, 60 different topics were examined. By contrasting our model's accuracy, precision, recall, and f1 score with various configurations and architectures, we are able to demonstrate the efficacy of our work. The best-performing model we used yielded an accuracy of 0.98.

The recent COVID-19 pandemic demonstrated how businesses that advertise their goods on social media platforms and influencers on those platforms were successful even while the country was under lockdown[105–107]. Due to this incident, people have been using social media sites more frequently to look for information and guidance. A sizable portion of the global population's social and economic behavior was influenced by the presence of content producers who could effectively sponsor messages and products from both private businesses and public institutions[108]. Influencers are social media users who have a lot of power in literature. Micro-influencers are a particular type of influencer[102]; they are more engaging and focus their content on a small number of interesting subjects. They have fewer followers (5k–100k) than well-known influencers, but they can persuade a larger proportion of their community[109]. These qualities translate into a higher return on investment for those who use them. The social graph data, text, and images that can be retrieved from social media posts are used in existing studies to identify influencers and micro-influencers. Instead of empathizing with followers who share similar tastes about the emotions evoked by images or words, researchers are prompted to prefer specialized solutions in order to maximize the accuracy in the classification of particular characteristics of micro-influencers, such as their ability to reach the greatest number of followers[56, 53, 58]. This research suggests a method for categorizing micro-influencers that combines text processing, image captioning, and social media graph features. Our process compiles a list of users who have written about the chosen topics. To create a balanced dataset for each topic, we filter social media accounts that don't match a set of metrics that characterizes them as potential micro-influencers. We eventually compile and examine their writings to verify the data from the preliminary filters.

We have outlined the following three research queries:

- RQ1: In the context of social media on the internet, what is the gold standard for categorizing micro-influencers?
- RQ2: How can I categorize micro-influencers using extensive data from multiple sources?

- RQ3: In the classification task for micro-influencers, is a gradient boosting ensemble method superior to deep neural network models?

The following sections' content is listed below. In Section 5.1, we describe how we develop a ranking strategy, an XGBoost model, and a rule-based Gold Standard to categorize micro-influencers. In Section 5.2, we present the findings from the evaluation of our model's recall, accuracy, and precision in comparison to various baseline classifiers. We highlight the challenges that emerged during the experiments in Section 5.3 and suggest some potential solutions to address them. In the final Section 5.4, we summarize the results and recommendations of our work and offer some recommendations for future research into this area.

5.1 Multi Input Micro-Influencers Classifier

The procedure we used to gather information from Instagram and Twitter in order to create a benchmark dataset is detailed in the section that follows. The following section shows our process for developing a micro-influencer classification model. We also go into detail about how we make use of the textual and visual elements of social media posts. The application of XGBoost (eXtreme Gradient Boosting) to improve the performance of our model is covered in the final section of this chapter.

5.1.1 Dataset creation and gold standard

We established guidelines to produce a gold standard for the study of micro influencers in the form of an annotated dataset. We handled every step of the pipeline, including the creation of the final dataset, setting up rules and thresholds to identify users as micro influencers, and obtaining data from social media. Following is a description of the defined rules. According to Brewster and Lyu [109], they adhere to the definition of a micro influencer. Days since the user account creation are tracked by age (*Age*). How many users follow a potential micro influencer is measured by their number of followers (*Followers Count).* As the following idea is unilateral, the user being followed is not required to follow the action's initiator back. The user must have between 5,000 and 100,000 followers in order to get past the filter. Users outside of this range are categorized as nano influencers, macro-influencers, or not influencing at all. A user's average daily gain in followers is indicated by their growth rate in followers (*Followers growth rate*). The user's capacity to grow his community consistently is scored. This indicator describes the user's potential to eventually connect with more people who are interested in the subjects he writes posts about. This score must be greater than 4 to pass.

$$Followers\_growth\_rate_i = \frac{Followers\_count_i}{Age_i} \tag{5.1}$$

In equation 5.1, *i* stands for the i^{th} user in our dataset. *Followers following ratio* is a score calculated to identify and weed out potential fake accounts known as bots that automatically post on social media and follow random people on social media in an effort to get a follow-back. Yang et al.'s research [110] thoroughly examines this definition. This strategy allows the fake account to expand quickly under certain circumstances, but it leaves a trail in the voluminous amount of followed accounts. Due to these factors, we set a filter to be above 2 for this score. The ratio of the user's followers to the accounts they follow must be at least two to one.

$$Followers\_following\_ratio_i = \frac{Followers_i}{Following_i} \tag{5.2}$$

In equation 5.2, *i* is the i^{th} user in our dataset.

When an account complies with Twitter's requirements for being deemed authentic, notable, and active, the label *Verified* is applied. The notable aspect of this label indicates that the user is in the top .05% of followers or mentions for your location, and it may be used as evidence of notability for some categories, even though the authentic and active rules allow us to exclude fake accounts. In accordance with this definition, we choose to only accept users who have this label set to False as micro-influencers because it indicates that they aren't already well-known as macro-influencers or brand-celebrities.

Tweet frequency and *status count* are the final two criteria chosen to filter out micro-influencers on Twitter.

The user's level of platform activity is measured by their *tweet frequency*. The calculation is the sum of the number of tweets sent over the days the account has been active. The cutoff for this score is greater than 10. The user cannot consistently amuse his audience if he falls below this threshold.

Score	**Threshold**
followers_count	>5k and <100k
followers_growth_rate	>4
followers_following_ratio	>2
verified	false
tweet_frequency	>10
statuses_count	>200

Table 5.1 In the initial phase of dataset collection and Gold Standard definition, Twitter scores and thresholds were used to identify users as micro-influencers. These cutoff points adhere to Brewster et al.'s [109] definition of a micro-influencer.

$$Tweet\_frequency_i = \frac{Statuses\_count_i}{Age_i} \tag{5.3}$$

In equation 5.3, i stands for the i^{th} user in our dataset.

The number of tweets the potential micro-influencer has posted on the platform is represented by their *statuses count*. Because there is insufficient content for our framework to analyze, we exclude users with less than 200 tweets in their timeline.

Table 5.1 contains a list of all the criteria used to categorize users as general micro-influencers on the social media platform Twitter. In order to balance our Gold Standard while maintaining the caliber of our samples, three of these scores are also used to gather non-micro-influencers. If any of the Table 5.1 metrics, with the exception of *statuses count* and *tweet frequency*, have negative scores, that person is also considered a non-microinfluencer. A non-micro-influencer according to the Gold Standard has more than 200 tweets in his timeline and a tweet frequency of at least ten, as anything less could be a fake account or a user who is too new and has insufficient produced content to be considered. By changing the names of the scores in Instagram to reflect the definition provided by this social media, we used a similar strategy. The number of followers for the chosen user is listed under the heading *Followers*. In this instance as well, a user must have between 5k and 100k followers in accordance with Brewster's definition to be regarded as a potential micro-influencer[109]. When a user first creates an account on the platform, it records how many posts he has made. This is called his *media count*. The user must have more than 200 Instagram posts in order for our framework to process them. The threshold for this score is once again 200. The proportion of his followers to the total number of social media posts is known as *followers per media*. The cutoff for this number is two. The *followers following ratio* is the same as what Twitter uses; in fact, a user with a ratio of less than two is likely a spam or fake account that follows others automatically in an effort to get a follow in return, increasing his number of followers. As per Twitter, in the Instagram dataset collection and Gold Standard definition, a user must have at least 200 media counts, less than 5k followers, or more than 100k followers while denying the other thresholds in order to be considered a non-micro-influencer and be involved in the dataset balancing. We create a Gold Standard with 60k tweets total, 300 unique users, 30 heterogeneous topics, and 200 tweets per user for the Twitter section. With 25 posts per user, we gather 15k posts across 30 topics from 300 users. Table 5.2 contains the aforementioned details. There are 600 users in the Gold Standard, whether they are considered micro-influencers or not. This section answers the RQ1 by using the ratings listed in Table 5.1 and their corresponding cutoffs to establish a gold standard for the micro-influencer industry.

Social	**Attribute**	**Value**
Twitter	number of topics	30
Twitter	unique users	300
Twitter	total tweets	60k
Instagram	number of topics	30
Instagram	unique users	300
Instagram	total posts	15k

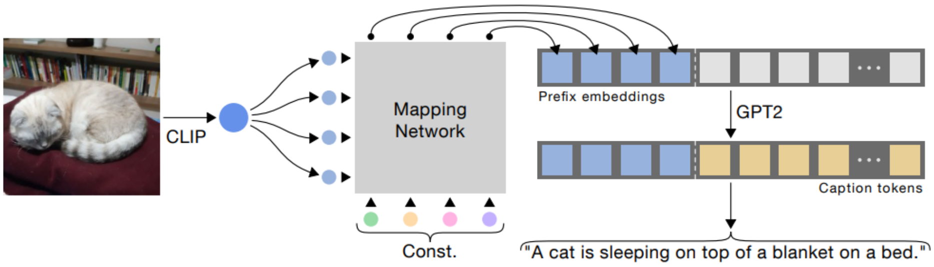
Table 5.2 Gold Standard dataset numbers. We collected a total of 75k social media posts, a total of 600 different users over 60 unique topics from Twitter and Instagram social media posts.

5.1.2 Micro Topic with Image Captioning and Text Processing

All of the general metrics used in literature and discussed in the Background Chapter 2 can be used to identify general micro-influencers. We observe a lack of specific topic-centric metrics to determine whether a user is a micro-influencer for both a general topic and one in particular. By creating four new topic-related scores, we close this gap. By creating two topic extraction pipelines for social media posts—one for visual content and the other for text—we compute these scores. Data gathered on Instagram is specific to *image captioning*. For each image that is retrieved, we create a textual description. Scores on expertise in microtopics are then computed using the caption. An stacked neural network with two modules makes up the architecture.

A language model converts the features of an image that a computer vision model has extracted into a meaningful sentence. This pipeline is depicted in Figure 5.1 and described in detail in the work of Mokady et al. [111].

Fig. 5.1 ClipCap model presented in the [111] by Mokady et al. to translate image features into meaningful image captions.



The visual encoder uses the pre-trained CLIP (Contrastive Language-Image Pre-Training) to extract image features and translate them into CLIP embeddings. An output prefix from the mapping network's processing of the embeddings is concatenated to the output of the dataset's textual embedding of the caption. The extended embeddings, made up of prefix plus caption embeddings, are fed to the Generative Pre-trained Transformer GPT-2, which is described in detail in the research work by Radford et al. [112] The CLIP model output features and the GPT-2 mapped captions embedding are combined in the final layer of this neural network to infer the caption. The language model outputs probabilities used to generate the following word in the sequence for future predictions, selecting the most likely, for each token created by the sentence splitting into words. The Common Object in Context COCO dataset [1] serves as the industry standard for image captioning in our work. 330k images and their captions are included in this Google dataset. Five distinct captions are included for each figure. After being trained, the ClipCap model shown in Figure 5.1 is capable of generating a textual description of an image taken from an Instagram social media post. All input data is text after image captioning. We calculate six additional scores—two for Twitter and four for Instagram—to determine whether a micro influencer possesses particular expertise and a keen interest in particular subjects.

With respect to the total number of tweets written in the user account, *Topic % in tweets* calculates the proportion of tweets that contain the topic searched. This proportion is defined in Equation 5.4 where i represents the i^{th} user in our dataset.

[1] cocodataset.org

$$Topic\_\%\_in\_tweets = \frac{Total\_tweets\_with\_topic_i}{Total\_tweets\_written_i} \tag{5.4}$$

Equation 5.5, where *i* represents the i^{th} user in our dataset, defines this ratio as the number of words in the entire user timeline divided by the percentage of words that are equal to the topic searched. We called this parameter *Topic % in words*.

$$Topic\_\%\_in\_words = \frac{Total\_topic\_words_i}{All\_words\_written_i} \tag{5.5}$$

In relation to the total number of captions written by the user in his account, *Topic_%_in_captions* counts the number of captions, which are image descriptions on Instagram, that are identical to the topic searched. Equation 5.6, where *i* represents the i^{th} user in our dataset, can be used to describe it.

$$Topic\_\%\_in\_captions = \frac{Total\_captions\_topic_i}{Total\_captions\_i} \tag{5.6}$$

Equation 6, 5.7 where *i* represents the i^{th} user in our dataset, defines this ratio as the number of words in all the captions divided by the percentage of words matching the topic searched. We name this metric *topic_%_in_caption_words*

$$Topic\_\%\_in\_cap\_words = \frac{Total\_topic\_words_i}{All\_caption\_words_i} \tag{5.7}$$

In relation to the total number of image captions processed, *Topic_%_in_pictures* counts the number of captions for images that contain the searched topic. Equation 5.8, where *i* represents the i^{th} user in our dataset, gives the definition of this ratio.

$$Topic\_\%\_in\_pictures = \frac{Total\_img\_cap\_topic_i}{Total\_img\_cap_i} \tag{5.8}$$

Equation 5.9, where *i* represents the i^{th} user in our dataset, defines this ratio as the proportion of words in all processed image captions that match the topic searched. *Topic_%_in_picture_words* measures this proportion as a percentage.

$$Topic\_\%\_img\_cap\_words = \frac{Total\_topic\_words_i}{All\_img\_cap\_words_i} \tag{5.9}$$

We established a ranking strategy for micro-influencers with the help of the computed scores. The scoring system is listed in Tables 665 and 666. These tables' categories column lists the metrics calculated after data retrieval in the General Statistics section. The score assigned to each percentile is displayed on the other column headers. The tables' topic statistics section is paired with scores calculated from text posts and image caption searches related to the target topic. Each user in our dataset is given a score, and only if their score is higher than the average of all the users' scores is that user designated as a micro-influencer for that particular topic.

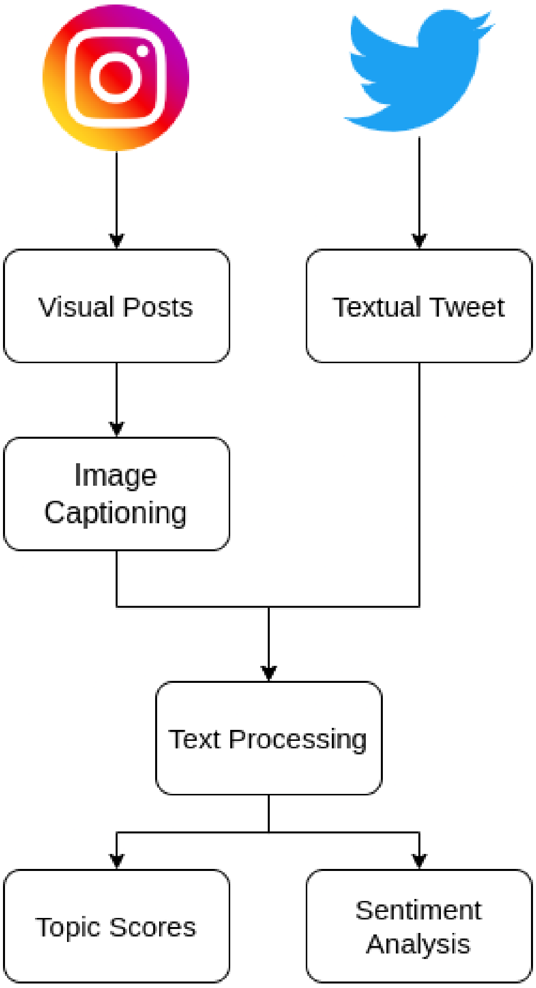
General Statistics					
			Points		
Categories	2	4	6	8	10
Followers count	5k - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	P_{80} - 100k
Followers growth rate	4 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Followers following ratio	2 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Tweet frequency	10 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Topic statistics					
	5	10	15	20	25
Topic % in tweets	0 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Topic % in words	0 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Maximum range score	20	40	60	80	100

Table 5.3 Twitter Micro Topic Influencer selection ranking

General Statistics					
			Points		
Categories	2.5	5	7.5	10	12.5
Followers count	5k - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	P_{80} - 100k
Followers growth rate	2 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Followers following ratio	2 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Topic statistics					
	2.5	5	7.5	10	12.5
Topic % in captions	0 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Topic % in cap words	0 - P_{20}	P_{20} - P_{40}	P_{40} - P_{60}	P_{60} -P_{80}	>P_{80}
Topic % in pictures	0 - P_{92}	P_{92} - P_{94}	P_{94} - P_{96}	P_{96} -P_{98}	>P_{98}
Topic % in pictures words	0 - P_{92}	P_{92} - P_{94}	P_{94} - P_{96}	P_{96} -P_{98}	>P_{98}
Maximum range score	20	40	60	80	100

Table 5.4 Instagram Micro Topic Influencer selection ranking

Fig. 5.2 Data processing flowchart outlining the gathering of tweets and Instagram posts and the subsequent feature extraction. The image captioning step is part of the pipeline for visual posts, as shown in Figure 5.1, but it is not necessary for text posts.



5.1.3 Sentiment Analysis

The sentiment analysis of tweets, Instagram post descriptions, and Instagram image captions brings the features evaluation to a close. Positive, neutral, and negative sentiment percentages for each post are the outputs. The scores per user are calculated as the average of each sentiment across all of the posts in the user's timeline, yielding a total of 9 scores—three for Twitter and six for Instagram. We make the decision to calculate these scores in order to find a topic-specific pattern paired with topics. A topic might be better suited for positive messages than a topic for neutral or negative ones. Barbieri et al.'s model, which we used, in [113]. Cardiff Twitter Roberta Base

Sentiment is the model's full name. These steps are used to clean the text before feeding it to the neural network. Emojis are converted into text, stop words are removed. We also take down the links. Our final step is to lemmatize words.

Figure 5.2 shows the entire pipeline, from data retrieval to topic score calculation and sentiment analysis. Following this stage, we are prepared to feed the described features to our classifier. Following is a description of our Multi Input Micro Influencer Classifier (MIMIC).

5.1.4 Model and Pipeline

Microinfluencer classification is a supervised binary classification problem. In order to determine which model was the most appropriate given our input scores as described in Section 5.1 and the anticipated output, we tested six different models. With the help of XGBoost, Random Forest Classifier, Support Vector Classifier (SVC), Multi-Layer Perceptron (MLP), Logistic Regression, and Stochastic Gradient Descent (SGD), we conduct the experiment. They create two labels: one to indicate whether a user is a micro influencer in the broad sense and another to indicate whether a user is a micro influencer for a particular topic that is input to the model. In Figure 5.3, the summary of this pipeline is shown. The Grid Search CV of the Scikit Learn library chooses the fine tuning parameters for these algorithms. The XGBoost model (eXtreme Gradient Boosting) produces the best outcomes. An amalgam of numerous weak classification models makes up the Gradient Boosting prediction model. The term "stage-wise model" describes it. It permits the optimization of any differentiable loss function, enabling a better tuning based on the solution of the chosen issue. A parallel tree boosting system is offered by XGBoost. The low interpretability of the generated results, however, is one of XGBoost's major disadvantages. On various subsets of the training dataset, XGBoost trains a huge variety of models before choosing the one that performs the best overall. The parallelization (training on multiple CPU cores), regularization (penalties mechanisms to prevent overfitting), non-linearity, cross-validation, and scalability of the XGBoost algorithm are some of its key characteristics. These features enable the algorithm to handle very large data volumes without sacrificing performance. With the help of this model, we are able to respond to RQ2 by choosing an XGBoost model that can handle the multiple inputs provided by our scores retrieval on our dataset.

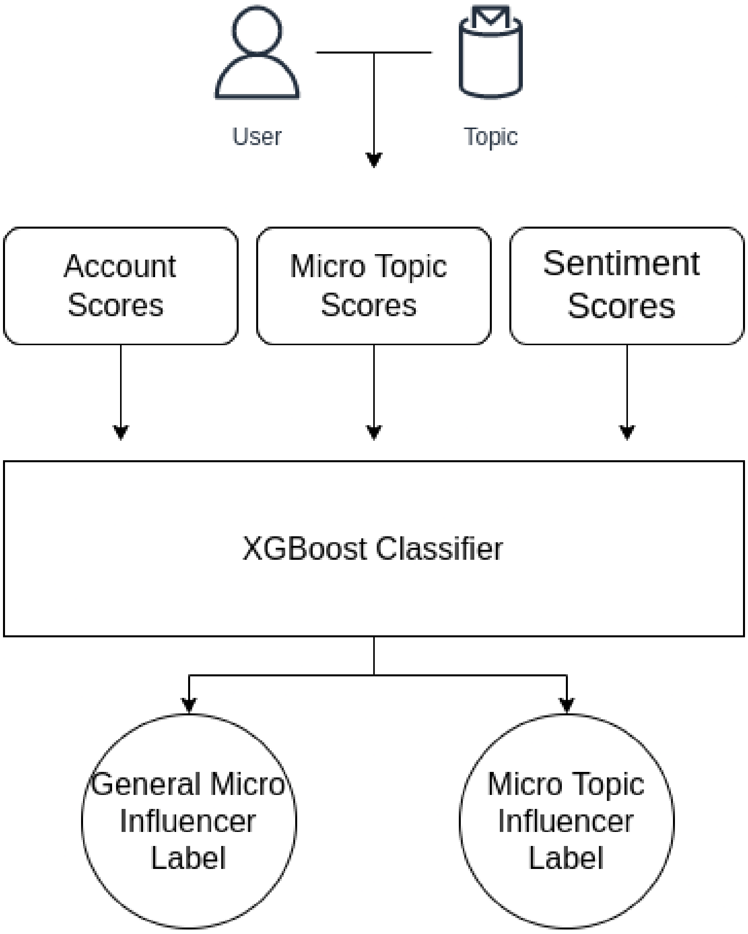


Fig. 5.3 Micro influencer classifier pipeline with multiple inputs. This is the general schema for data retrieval from a social media platform, score calculation, classification model selection, and final labels definition for both general micro influencers and micro topic influencers.

5.2 Results

For the purpose of retrieving user writings and image posts, we use the social media sites Twitter and Instagram. We choose users who have recently posted on the subjects we chose for the experiment. The Table 5.1 thresholds are used to filter and label users. We download the most recent 200 posts for each user. In the case of Instagram, the pictures are converted into captions and saved as a text post. The text cleaning process described in Section 5.1 is applied to an individual user's entire body of written work. The initial dataset is then split into training and test sets at an 80/20 ratio, balancing the number of micro and non-micro influencers for both the general case and the topic-specific case. The following feature lists were gathered for each classifier:

Twitter features: followers count, age, followers growth rate, followers following ratio, tweet frequency, topic % in tweets, topic%in words, positive sentiment, neutral sentiment, negative sentiment.

Instagram features: followers count, followers growth rate, followers following ratio, topic % in captions, topic % in cap words, topic % in pictures, topic % in pictures words, positive sentiment captions, neutral sentiment captions, negative sentiment captions, positive sentiment captions words, neutral sentiment captions words, negative sentiment captions words.

Recall, precision, and accuracy metrics, as described in Equations 5.10, 5.11 and 5.12, have been adopted as measures to evaluate the efficacy of various models. In this study, users are divided into two categories: micro-influencers and non-micro-influencers. In the second of our two scenarios, we label users as either micro-topic influencers or non-micro-topic influencers. We classify both instances using a binary system. Recall, precision, f1-scores, and accuracy are the metrics we use to evaluate the validity of our model. Following the descriptions, we define true positive, true negative, false positive, and false negative. The user tp (true positive) is appropriately categorized as a micro-influencer (label 1). A user is correctly identified as a non-micro-influencer (label 0) by the tn (true negative) label. False positives are users who are considered micro-influencers despite not being micro-influencers themselves. A micro-influencer referred to as a "false negative" is categorized as a non-micro-influencer.

$$recall = \frac{tp}{tp+fn} \tag{5.10}$$

$$precision = \frac{tp}{tp+fp} \tag{5.11}$$

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{5.12}$$

The XGBoost model consistently achieves the best results during the validation process. A deep neural network that uses BERT for sequence classification and then maps all of the sequence embeddings into a user embedding before max pooling to derive a classifier with a final neuron layer to classify micro influencer is also less effective than XGBoost. In Tables 5.5 and 5.6, these findings are displayed.

	Twitter General Micro Influencer Classification Metrics			
Model	Accuracy	Precision	Recall	f1-score
XGBoost	0.99	0.99	0.99	0.99
BERT-based	0.80	0.81	0.80	0.80
SVM	0.73	0.74	0.73	0.73
MLP	0.90	0.90	0.90	0.90
LR	0.77	0.77	0.77	0.77
SGD	0.52	0.31	0.52	0.39

Table 5.5 Comparison metrics between different models tested. Twitter general influencer classification results.

	Twitter Topic Micro Influencer Classification Metrics			
Model	Accuracy	Precision	Recall	f1-score
XGBoost	0.93	0.93	0.93	0.93
BERT-based	0.79	0.81	0.80	0.80
SVM	0.88	0.90	0.88	0.89
MLP	0.63	0.62	0.63	0.62
LR	0.88	0.90	0.88	0.89
SGD	0.65	0.69	0.65	0.66

Table 5.6 Comparison metrics between different models tested. Twitter topic influencer classification results.

In the Instagram case, as shown by Tables 5.7 and 5.8, XGBoost performs better than all the other models in each of the four computed validation metrics. It’s intriguing to see that it also outperforms a CNN-based model that convolutionally extracts features from a sequence of text and images before feeding them into a single layer neural network to compute the final label.

We can also satisfactorily respond to the third and final research question, RQ3, as a result of these findings.

5.3 Discussion

There are various methods for evaluating influencers, and while many of them also apply to the situation of micro influencers, others do not. We concentrate on the

	Instagram General Micro Influencer Classification Metrics			
Model	Accuracy	Precision	Recall	f1-score
XGBoost	0.98	0.98	0.98	0.98
CNN-based	0.77	0.75	0.76	0.76
SVM	0.60	0.60	0.60	0.60
MLP	0.65	0.65	0.65	0.65
LR	0.60	0.60	0.60	0.60
SGD	0.55	0.55	0.55	0.55

Table 5.7 Comparison metrics between different models tested. Instagram general influencer classification results.

	Instagram Topic Micro Influencer Classification Metrics			
Model	Accuracy	Precision	Recall	f1-score
XGBoost	0.98	0.98	0.98	0.98
CNN-based	0.73	0.74	0.73	0.73
SVM	0.62	0.62	0.62	0.61
MLP	0.63	0.63	0.63	0.63
LR	0.58	0.58	0.58	0.58
SGD	0.50	0.50	0.50	0.50

Table 5.8 Comparison metrics between different models tested. Instagram topic influencer classification results.

latter case, adopting a multi-input approach to take into account a broad range of input features that can be retrieved either directly from the initial post or after a stage of image or text processing. We calculate social media account metrics for users along with topic-specific features. This method of finding and categorizing micro influencers in a given field better satisfies the needs of private businesses and governmental organizations. The main challenge is data retrieval, even though the pipeline we developed is simple with the stacking of existing models. In fact, Instagram is more restrictive than Twitter, and using a library like Instaloader to gather social media posts is not the best performing option, but it is the only one that can be exploited. Twitter, on the other hand, already provides a developer API platform to collect data, even though the timeout makes the collection really slow. We are also aware that when images are translated into textual captions, some significant visual elements may be lost. However, it is challenging to understand how to match these features with the ultimate goal of classifying micro influencers; this approach still needs to be further investigated. Although the gold standard dataset has been created from scratch in accordance with business rules, it is advisable to perform a user supervision on the assigned label in the future after verifying the effectiveness of these users' micro-influencing abilities. This automatic procedure is useful for both identifying micro-influencers and ranking them among other users with similar characteristics.

5.4 Conclusion

This work introduces a new framework for Twitter and Instagram to gather and categorize micro influencers in general and in cases pertaining to particular topics. By using validation metrics, we were able to demonstrate that XGBoost is the most efficient model for carrying out this task, receiving the features gathered with an overall accuracy above 0.93. Replicating the procedure will allow you to increase the dataset and investigate more subjects. The majority of this work is dedicated to writing text for Instagram and Twitter. a progression toward comprehension. Users' communication abilities might be based on a visual information study and performance. Text to video conversions for Instagram Reels and Instagram Stories. Additionally, some of the metrics used in this study account for the topic word's presence throughout the entire text that was obtained. This strategy can be expanded by using topic detection algorithms to capture even synonyms or periphrasis. Even though there are still many unexplored avenues, we established the viability of our model and produced a fresh dataset that can be used for more in-depth investigation.

Chapter 6

Automated Classification of Fake News Spreaders to Break the Misinformation Chain

The proficient classification of micro influencers as demonstrated in Chapter 4 and 5 set the basements for a further application. In particular, during the COVID-19 pandemic crisis, the WHO (World Health Organisation) asked for support also in the containment of fake news diffusion. The WHO called this phenomenon of misinformation related to the pandemic an infodemic. Our research team decided to offer a contribution to this hazard providing a tool to support the breaking of the misinformation chain. The work done with micro influencers in the previous chapter acts as a starting point for this new challenge.

People tend to spread false information on social media platforms with ease and without doing any fact-checking. They are not malicious in intent, in theory, but their sharing creates a diffusion mechanism that is risky for society. The reasons for this behavior have been linked to a wide range of social and personal outcomes, but it is difficult to identify the users. The existing solutions demonstrated how this field can benefit from linguistic signal analysis in social media posts along with network topology exploration. These programs have some drawbacks, such as a single-minded focus on shared fake news and a lack of understanding of the user types who spread it. The computational method we suggest in this paper can be used to identify these users as fake news spreaders for a specific topic by extracting features from their social media posts. We are able to begin the analysis with 300K user engagements from a microblogging platform online thanks to the CoAID dataset, and we then add to them by expanding the dataset with a collection of more than 1M share actions and their associated users' writings on the platform. The suggested method converts a group of Twitter posts written by CoAID dataset users into a high-dimensional matrix of features, which is then used by a deep neural network architecture based on transformers to carry out user classification. By contrasting the precision, recall, and f1 score of our model with various configurations and with a reference classifier, we demonstrate the efficacy of our work. The state-of-the-art was raised by 4% thanks to our f1 score of 0.8076.

The World Health Organization (WHO) declared Covid-19 a pandemic on March 11, 2020, and information about the virus and precautions to take to stop it from spreading has since flooded social media platforms and traditional media. People responded with non-adaptive coping mechanisms at the same time because of the uncertainty and ambiguity surrounding the COVID-19 information. Ha et al. in [114]. says that when communicating with the public, it's important to include a positive message that can lower anxiety in addition to status updates and behavioral advice. When this tactic fails, people often react by fabricating harmful scenarios as fake news to protect themselves in an effort to downplay the perceived threat. These results show that the spread of false and misleading information on social media was accelerated by the COVID-19 pandemic. In 2020, there was a sharp increase in the number of incidents where someone received fake news and quickly spread it on social media without checking its veracity. according to Cinelli et al. Citing [115] and the World Health Organization [1], it is stated that fake news has become a global pandemic, and that effective countermeasures are needed to reduce human effort in spotting false information and slow its spread. according to the study by Oshikawa et al. [116]. To solve this issue, language models have been applied extensively. Islam and others in this context identified fake news with several deep learning architectures[117]. Jiang et al. in [76], on the other hand, They looked into linguistic signals to find emotional markers in text and found that when a user is encouraged to read fact-checking articles, he engages differently on social media. Similar to this situation, Glenski et al.[118] response, appreciation, elaboration, and question were a few of the different types of responses to false information that were observed. Understanding a user's position on the content he is sharing is essential to

[1] https://www.who.int/news-room/spotlight/let-s-flatten-the-infodemic-curve

correctly identifying whether the user is a supporter or a critic of the content. These studies all show how the characteristics of the users who spread fake news are related to the diffusion mechanisms for it. The fake news phenomenon on social media has actually led to the definitions of fake news spreaders and checkers. Social media users who promote and spread false information are known as fake news spreaders. On the other hand, genuine news consumers are social media users who support and share genuine news. We define checked fake news as information that fact-checking organizations have determined to be false after a human review process.

According to the cited research projects in Chapter 2, there is an increasing need for automated solutions to assist fact-checking organizations in their monitoring efforts and to raise public awareness of the importance of double-checking content before sharing it. In fact, according to Vosoughi et al.[69], when a user disseminates false information, he increases the community's confidence in the material, exponentially expanding its reach. When the information spreads quickly, the authorities must make a significant effort to show it to be false. The research already done in this area has demonstrated how well language models combined with social media interaction analysis can catch false information. We contend that while these solutions undoubtedly have a significant social impact, they do not thoroughly analyze the false information from the perspective of the user disseminating it. This gap is filled by the analysis of user writings and behavior on social media platforms, which explicitly expresses his viewpoint. In order to categorize a user's propensity for spreading false information, the solutions put forth do not investigate the encoding of the user's timeline into sentence embeddings. They also don't compare machine learning models that make use of social media graph features with natural language processing methods.

As a result, we conducted our research to address the following research questions:

- RQ1 Is deep learning and transformer-based sentence encoding used. effective at categorizing fake news peddlers in the context of COVID-19 news?
- RQ2 In the context of COVID-19 news, which gold standard can be used to categorize those who spread false information?

In this study, we introduce the FNSC (Fake News Spreader Classifier), a Transformer based stacked neural network that combines our deep learning model and the Transformer's ability[74] to compute sentence embeddings to categorize users who spread false information about COVID-19. This model turns batches of tweets into sentence embeddings and uses them in a supervised manner to categorize users. We begin by gathering tweet authors and their timelines from the dataset created by Limeng and Dongwon [119] in order to thoroughly examine the information they shared about COVID-19 and determine whether they agreed with the news they shared using a stance detection model. Using linguistic features, we demonstrate that our model produces cutting-edge results. We also tested the performance of our model using only social media metrics, and the results were less favorable than those from linguistic metrics.

The code we built is available in a publicly accessible repository.[2] The CoAID dataset is also available in a publicly accessible repository.[3] We shared the extended version of CoAID on Figshare.[4]

The rest of this essay is organized as follows. In Section 6.1, we explain the gold standard for the Spreader and Checker classification challenge and how we extend the CoAID dataset. The deep learning model and our strategy are described in Section 6.2. When using our architecture on the CoAID extended dataset, we report the experimental results in Section 6.3 for the Spreader and Checker classification. In Section 6.4, we go over the outcomes of our methodology and explain the decisions we made regarding the baseline comparison and the linguistic model. Finally, we wrap up with some final thoughts and future projects in Section 6.5.

6.1 Fake news spreader gold standard

There are two main resources in the CoAID dataset[119]. The first one is a table that lists details about both true and false news about COVID-19, including the news' URL, a link to the organization that verified it, as well as its title, content, abstract, publish date, and keywords (see Table 6.1).

The second is a list of tweet ids that includes both fake and real news along with a covered author reference id. Four categories—false and true claims, false and true news—are assigned to tweet ids. The latter, however, have a clear URL

[2] https://github.com/D2KLab/stopfaker

[3] https://github.com/cuilimeng/CoAID

[4] https://doi.org/10.6084/m9.figshare.14392859

Type	Features
News Information	ID, Fact-checking URL, Information URLs, Title
News Information	Article title Content, Abstract, Publish date, Keywords
User Engagement: tweets	ID, Tweet ID

Table 6.1 Feature descriptions of News Table and User Engagement in CoAID dataset.

redirecting to the news while the former are just opinions without any URLs inside. Since we need both the content of the article and the content of the tweet to perform the stance classification, we chose to work with the last few. The CoAID dataset for COVID-19 contains 296,000 user engagements and 4,251 fact-checked news items. The checked fake news consists of articles that fact-checking organizations have already proven to be false. Six of these organizations are taken into account in this project: LeadStories[5], PolitiFact[6], FactCheck.org[7], CheckYourFact[8], AFP Fact Check[9], and Health Feedback[10]. The information was gathered between December 1, 2019, and November 1, 2020, with a publishing date between those two dates. We take the tweets with links to real news or fake news sources and identify the social media users who posted them. The user id of each author is hidden due to privacy restrictions in the CoAID dataset, so we must use a Twitter API query to retrieve them. By retrieving for each user his entire timeline starting on December 1, 2019, we create an expanded version of CoAID. The linguistic model to categorize fake news spreaders is one of the two main contributions of this research project, along with our extended version of the CoAID dataset, which is publicly available on Figshare[11]. Figure 6.1 describes the retrieval pipeline in detail. 2012 tweets on average are posted by each of the 11465 users we track.

Because the downloaded tweets contain the Twitter shortened version of the original posted links, our text preprocessing phase and data cleaning includes an initial phase of URL extension. The shortened URL https://t.co/3g8dLgoDOf, for instance, must be made longer to https://www.dailymail.co.uk/health/article-9225235/Rare-

[5] https://leadstories.com/hoax-alert/
[6] https://www.politifact.com/coronavirus/
[7] https://www.factcheck.org/fake-news/
[8] https://checkyourfact.com/
[9] https://factcheck.afp.com/
[10] https://healthfeedback.org/
[11] https://doi.org/10.6084/m9.figshare.14392859

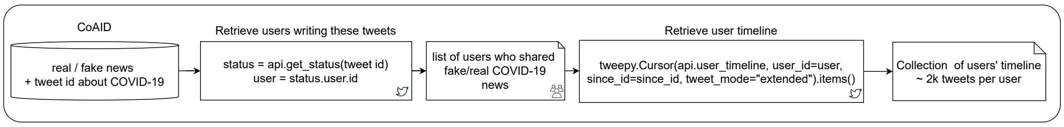

CoAID
real / fake news
+ tweet id about COVID-19
Retrieve users writing these tweets
status = api.get_status(tweet id)
user = status.user.id
list of users who shared fake/real COVID-19 news
Retrieve user timeline
tweepy.Cursor(api.user_timeline, user_id=user, since_id=since_id, tweet_mode="extended").items()
Collection of users' timeline
~ 2k tweets per user

Fig. 6.1 Spreaders and Checkers Timelines Retrieval extending CoAID dataset.

COVID-arm-effect-leaves-people-got-Modernas-shot-itchy-red-splotch.html. The now-extended link is looked up in the CoAID dataset, and if a match is found, the text from the original tweet and the CoAID news abstract are used as input to perform the stance detection.

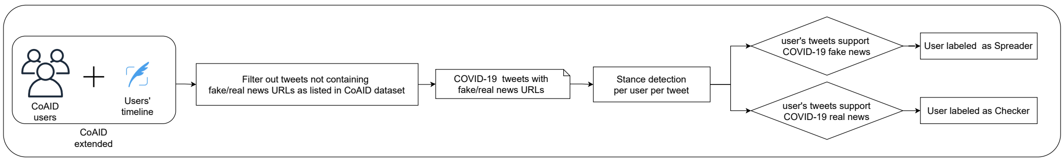

CoAID users
Users' timeline
CoAID extended
Filter out tweets not containing fake/real news URLs as listed in CoAID dataset
COVID-19 tweets with fake/real news URLs
Stance detection per user per tweet
user's tweets support COVID-19 fake news
user's tweets support COVID-19 real news
User labeled as Spreader
User labeled as Checker

Fig. 6.2 Pipeline to label each user as a Spreader or a Checker.

The stance classification model is an altered version of the Aker et al. [120] within the framework of our use case scenario. This word-based Random Forest classifies the source tweet in relation to other tweets using Bag of Words, Part of Speech Tagging, Sentiment Analysis, and Named Entity Recognition. The Gate Cloud community[12] further optimized the pipeline, and the source code is accessible in a publicly accessible repository[13]. The pipeline now functions with pre-trained multilingual BERT embeddings. The stance classification output indicates whether a tweet affirms, disputes, questions, or comments on the news item that is linked. We ignore the query and comment cases while counting the affirmative and negative ones. A user is classified as a Spreader or a Checker depending on how much fake news they support. If the number of true and false news stories is equal, the user is eliminated. Figure 6.2 shows the pipeline that describes this process. While a user is attempting to disprove the detected fake news, the stance classification algorithm will not label him as a Spreader. We now have an expanded version of the original CoAID dataset that can be used as the industry standard after data retrieval, cleaning, and user labeling. Table 6.2 contains a summary of this dataset's statistics. There are 5333 Spreaders and 6132 Checkers, and each one generates an average of 19 tweets in support of fake news and 55 tweets in favor of legitimate news.

[12] https://cloud.gate.ac.uk/

[13] https://github.com/GateNLP/StanceClassifier

Total number of Users	11465
Spreaders	5333
Checkers	6132
Average number of tweets per user	2012
Total number of tweets	23,068,006
Average number of tweets supporting Fake News per Spreader	19
Average number of tweets supporting Real News per Checker	55

Table 6.2 The CoAID extended dataset statistics.

For the classification of users disseminating false information about COVID-19, we developed the industry-standard dataset. A list of real tweet ids that were obtained from Twitter, a list of mapped used ids for privacy reasons, and a label designating the tweet author as a Spreader or Checker are all included in the extended version of the CoAID dataset. Table 6.3 contains a list of five rows from the CoAID extended dataset that were chosen at random. As stated in Section 6.1, this gold standard responds to the RQ2.

user_mapped_id	**tweet_id**	**label**
2442	1340854864562311168	1
8885	1346408723330314241	0
6260	1367096980762226688	1
10728	1285659580677193734	0
1956	1352412905199681538	1

Table 6.3 For identifying fake news spreaders, the CoAID extended gold standard dataset was used. In order to protect privacy, the Twitter user id is transformed into the user mapped id. The numerical identification of the tweet provided by the Twitter platform is represented by the term "tweet id.". If a user is a Spreader or a genuine news Checker, it is indicated in the third column.

6.2 Automated Classifier

We develop a linguistic model based on a sentence-level attention mechanism strengthened by a neural network architecture that distinguishes between real news checkers and fake news spreaders. The gold standard on which we test our model in three different configurations and contrast it with the work by Giachanou et al.[80] is the extended version of the CoAID dataset. We create a Spreader and Checker classifier using a text-based linguistic model. Here, we'll go over how to build a stacked neural network and how to analyze the input batch of tweets. A stacked neural network is defined as a collection of publicly accessible neural network architectures whose features have been extracted at a middle layer of the network and have been concatenated to create a larger feature set. This strategy is used in both the ensemble with social media metrics as a comparison model and the sole text model.

6.2.1 Transformer-based Tweet Embeddings

As a collection of tweets and their authors, we have organized the CoAID extended dataset. This data format draws our focus to a model that is largely user-centered. The written output of an author is represented by this collection of unformatted tweets, batch-processed by user. We depict the characteristics of a single tweet by turning the text into a sentence embedding. The multi-headed attention layers in Figure 3.3's BERT encoder are how we carry out this operation.

Equation 6.1 describes how the attention mechanism works. Given a sequence of n d-dimensional vectors $x = x_1, ..., x_n \in R^d$, and a query vector $q \in R^d$, the attention layer parametrized by $W_k, W_q, W_v, W_o \in R^{dxd}$ computes the weighted sum in Equation 6.1. W_k represents the matrix of *key weight* vectors, while W_q is the matrix of *query weight*, then W_v is the *value weight* matrix and finally the W_o is the *output* matrix of weight by which the concatenated attention head are multiplied by. The training phase of these four matrices is described in detail in [29]. In self-attention, every x_i is used as the query q to compute a new sequence of representations. Each attention head, A in the equation, is composed of the four W matrices (W_k, W_q, W_v and W_o) that are learnt during training. The W_o and W_v are elements of the weighted average of word vectors and Wq and Wk are involved in computing the α_i weights. Equation 6.2 computes the multi-headed attention, M in the equation,where N_h is an independently parameterized attention layer applied in parallel to obtain the final result.

$$
\begin{aligned}
A_{W_k,W_q,W_v,W_o}(\mathbf{x},q) &= W_o \sum_{i=1}^{n} \alpha_i W_v x_i \\
\alpha_i &= softmax(\frac{q^T W_q^T W_k x_i}{\sqrt{d}})
\end{aligned}
\tag{6.1}
$$

$$
M(\mathbf{x},q) = \sum_{h=1}^{N_h} AW_k, W_q, W_v, W_o(\mathbf{x},q) \tag{6.2}
$$

$$
u_j = \max_{1 \leq i \leq 768} c_{ij} \tag{6.3}
$$

An intermediate synthesis of a user's textual production can be found in the batch of embeddings that is produced. In the last part of this Section, the last step of the combination process is explained. We are able to represent sentences at the sentence level thanks to the tweet transformation from text to tweet embedding, and the user embedding u produced by the max pooling 1d described in Equation 6.3 is processed for the user classification task. In Equation 6.3, the initial matrix of tweet embeddings is C = $(c_{ij})_{1 \leq i \leq m, 1 \leq j \leq 768}$, where m is the total number of tweets collected for the user in analysis and 768 is the dimensions of each tweet embedding.

Because no data cleaning is done to the original text, it still retains its full meaning as intended by the author. The tweet is broken up into words and then smaller tokens, as seen in Figure 6.3. Out of vocabulary words can be handled by the BERT-tokenizer by breaking them up into shorter substrings. Each tweet appears as a string-type list of tokens after the tokenization phase is complete. It is crucial to include a unique CLS token at the start of the tweet. This particular custom token is employed for the classification task. The twelve successive encoding layers of the architecture then process the list of tokens to convert each input token into an output word embedding. In Figure 6.4, a sample encoding layer is shown. Except for the CLS, all word embeddings are ignored. Since the attention mechanism and the surrounding context both played a role in how the CLS embedding was created, we use it as a representation of the tweet as a whole. In the case of a brief tweet, the entire sentence serves as the surrounding context. A single token has 768 dimensions, which corresponds to the BERT base model's ideal setup [121]. The authors of the empirical experiment described in [121] proposed the number 768 as the optimal number of features based on comparisons of the outcomes of four different tasks:

named entity recognition, general language understanding evaluation, and multi-genre natural language inference. By using the Hugging Face transformers library[14], we were able to modify the original BERT architecture with the multilingual pre-trained embeddings for our particular context. Sentence level attention mechanisms work well in our situation because each word is important for comprehending the social media post. The weight of each word is greater in this context because we mostly use medium and short sentences. The user is represented by a list of all the embeddings of his tweets from his timelines after each tweet has been converted into one. Each embedding adds to the dense feature set that the FNSC's subsequent layers use to perform user classification.

[14] https://huggingface.co/transformers/model_doc/bert.html

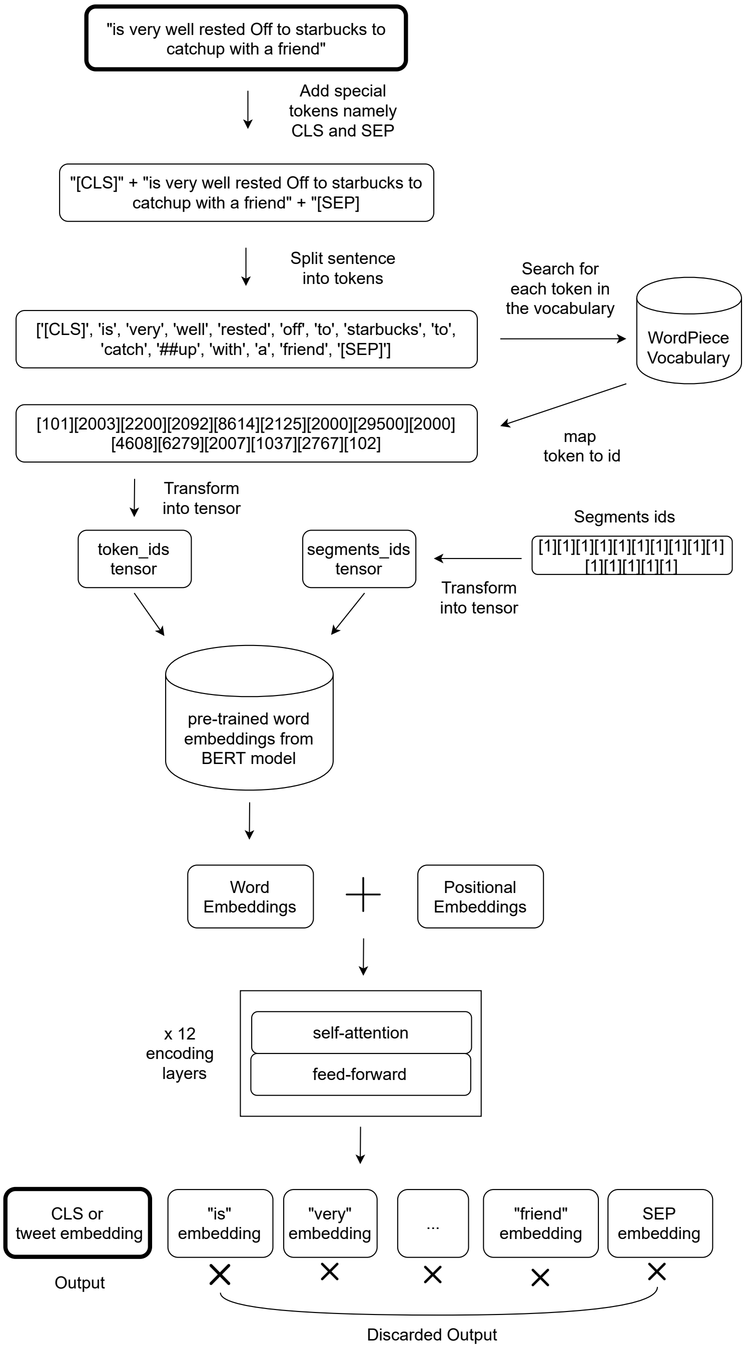


Fig. 6.3 Transformer to tokenize and encode data. Each tweet in the CoAID extended dataset is processed as seen in the figure. We start the tweet with a CLS token (classification task special token) and end it with a SEP token (separation between sentences). Once divided into tokens, we sent the tweet. To identify it as a non-standalone word, the second part of the split words is preceded by two hashtags. The WordPiece vocabulary's id is used to map tokens to, and the array thus computed is converted into, a Tensor. A tensor called segments_ids tensor made of 1s that is the same length as the token_ids tensor is also required. When we complete a task that requires two sentences, the segment_ids are helpful for separating tokens from the first sentence (0s) from the second sentence (1s). In our situation, we only require a single sentence, so we load segments_ids with 1. In order to generate word embeddings from our tensors, we load pre-trained embeddings from the BERT model and add initially random positional embeddings to them. Twelve encoding layers with a feed forward network and self attention are located at the bottom of the figure and encode the input into the final tweet embedding.

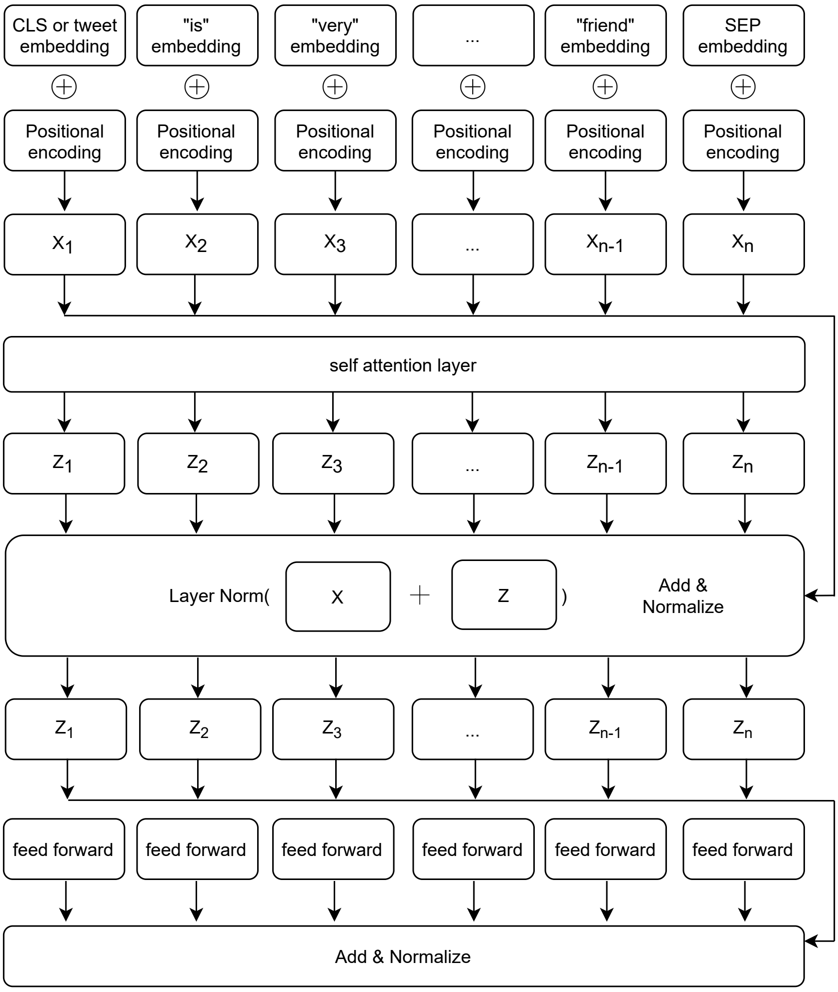


Fig. 6.4 This depicts one of the encoding layers mentioned in Figure 6.3. In the final architecture, there are a total of twelve of these encoding layers. Each token's word embedding goes through these layers of encoding before being transformed at the end.

6.2.2 Fake News Spreader Classifier

Our deep learning model is presented in Figure 6.5 as the Fake News Spreader Classifier. It receives a batch of tweets, converts them into 768-dimensional arrays, and stores them in a bidimensional tensor (there are 768 tweets per author). In order to extract the highest float value for each of the 768 dimensions, the intermediate FNSC layer now performs a 1d max pooling. After conducting empirical tests, we made the decision to extract the highest value in order to create a dense representation of a user's timeline without sacrificing the most distinctive characteristics of each

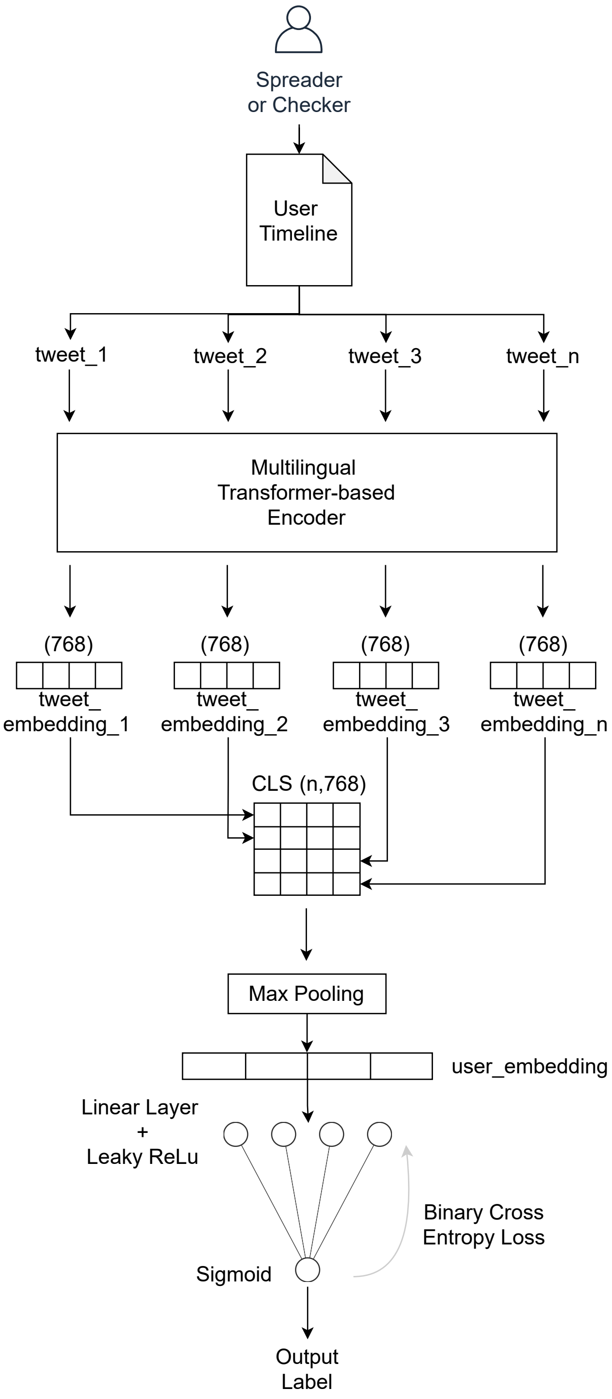


Fig. 6.5 FNSC (Fake News Spreader Classifier) architecture description.

user. The user level embedding is then processed by a combination of a Linear Layer coupled with a Leaky ReLu activation function and the output layer, which is a sigmoid function, after this stage.

The Linear Layer uses the linear function $h_\theta(x) = \sum_j \theta_j x_j = \theta^\top x$ to represent $h(x)$, where $h_\theta(x)$ acts for the linear function family parameterized by θ.

When the unit is not in use, the Leaky ReLu function, denoted by the symbol f(x) in Equation 6.4, allows for a modest positive gradient. Making this decision enhances performance and expedites the learning process. The vanishing gradient issue is also avoided using it. When the error signal is back-propagated and it decreases/increases exponentially in relation to the distance from the final layer, we have a vanishing gradient in the feed-forward network. When it comes to the final binary classification, the single neuron with the sigmoid activation function returns a probability that is used in conjunction with a threshold at 0.5. As a loss function for the architecture, we use BCE-loss (Binary Cross Entropy Loss). To determine how far the prediction is from the expected output, use the equations 6.5 and 6.6, which also tune the weights of the neural networks using error back propagation. In particular, y is the label (1 for Spreader and 0 for Checker), and p(y) is the predicted probability that the sample would be a Spreader for all N samples in the batch. In order to calculate the probability of being a Spreader, the formula adds log(p(y)) to the loss. On the other hand, it increases the Checker samples by log(1-p(y)). The Equation 6.6 is condensed into the Equation 6.5.

$$f(x) = \begin{cases} x & \text{if} \quad x > 0 \\ 0.01x & \text{if} \quad x \geq 0 \end{cases} \tag{6.4}$$

$$H_p(q) = -\frac{1}{N_{pos} + N_{neg}}[\sum_{i=1}^{N_{pos}} log(p(y_i)) + \sum_{i=1}^{N_{neg}} log(1 - p(y_i))] \tag{6.5}$$

$$H_p(q) = -\frac{1}{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i)) \tag{6.6}$$

The architecture shown in Figure 6.5 is adaptable because it is not dependent on the volume of tweets a user posted to his timeline. Due to the uneven weight of each Tweet in the intermediate user embedding representation, a user with more than a thousand tweets is, in any case, better represented than one with few tweets.

6.2.3 Parameters in Model Optimization

Both the encoding phase and the classification phase architectures require the definition of hyper-parameters. We defined these parameters in Table 6.4 with regard to the architecture of Figure 6.3.

- **Pre trained embeddings**, the starting point of the original BERT weights to further fine tune the model based on our data.
- **Tokenizer max length**, the maximum number of tokens accepted by the BERT-tokenizer.
- **Return Tensor**, the return tensor format after encoding.
- **Hidden Size**, the number of neurons in each hidden layer.
- **Hidden Layers**, the number of layers represented with the self-attention plus feed-forward.
- **Attention Heads**, this number tunes the self-attention mechanisms described in the work of Vaswani et al. [29].
- **Intermediate Size**, it represents the number of neurons in the inner neural network of the encoder feed-forward side.
- **Hidden Activation Function**, it is the non-linear activation function in the encoder. *GeLu* is the Gaussian Error Linear Unit.
- **Dropout Probability**, this number represents the probability of training a given node in a layer where 0 is no training and 1 is always trained.
- **Maximum Position Embedding**, it is the maximum sequence length accepted by the model.

Table 6.5 Parameters to configure the neural network of Figure 6.5: optimizer, learning rate, loss function batch size.

Parameter	Value
optimizer	**Adam**, Adagrad, SGD
learning rate	**2e-5**, 1e-2, 1e-7
loss function	Binary Cross Entropy Loss
batch size	8, 16 , **32**

Table 6.4 Parameters chosen to configure the encoder architecture of Figure 6.4

Parameter	Value	Parameter	Value
pre trained embeddings	bert-base-multilingual-cased	tokenizer max length	128
return tensor	pt	hidden size	768
num hidden layers	12	num attention heads	12
intermediate size	3078 (768x4)	hidden act	gelu
hidden dropout prob	0.1	max position embedding	512

The following parameters must also be optimized for our neural network architecture, as seen in Table 3.2.

- **Optimizer**, it changes the weights of the neurons based on loss to obtain the most accurate result possible.
- **Learning Rate**, it is the correction factor applied to decrease the loss. Too high values of learning rate lose some details in weights setting, too low values may lead the model to a very slow convergence.
- **Loss Function**, it computes the distance between predicted values and actual values.
- **Batch Size**, it is the number of training examples utilized in one iteration.

Table 6.5 provides a list of the parameters that we selected for our neural network architecture. The decisions we made are supported by empirical data.

6.3 Results

To determine whether we are advancing the state of the art in the classification of fake news spreaders, we analyze the results of the aforementioned model in this section. With the list of features listed in Table, we compare our model to a prior one by Giachanou et al.[80] as well as various machine learning and deep learning techniques.

The CoAID extended dataset, which contains 23M tweets and 11465 users, was used for the experiment. Table 6.7 compares our model's output to that of other configurations and earlier state-of-the-art output using recall, precision, and f1 score as validation metrics. We averaged the outcomes from each split using a tenfold cross validation. To confirm the model is successful in each split of the CoAID extended dataset, we perform ten fold cross validation.

Model	**precision**	**recall**	**f1**
Fake News Spreader Classifier	**0.8042**	**0.8110**	**0.8076**
RF Fake News Spreader Classifier	0.7977	0.8104	0.804
Giachanou et al. [80]	0.7789	0.7536	0.7660
Mixed Fake News Spreader Classifier	0.7364	0.7430	0.7234

Table 6.6 Precision, recall, and f1 scores were calculated as a comparison between our proposed Fake News Spreader Classifier model, our RF Fake News Spreader Classifier, which is a Random Forest model utilizing Twitter user's information, and the baseline by the work of Giachanou et al. and a mixed model that uses tweet embeddings and user account information from Twitter as inputs. Results indicate that the Fake News Spreader Classifier is the most efficient in user classification, and this is supported by the overall higher scores.

In the binary case, validation metrics determine how many candidates are accurately categorized in relation to the anticipated result, as shown in the following. *tp* (true positive) represents the number of Spreaders correctly classified. *tn* (true negative) represents the number of Checkers correctly classified. *fp* (false positive) represents the number of Checkers classified as Spreaders. *fn* (false negative) represents the number of Spreaders classified as Checkers.

$$p = \frac{tp}{tp+fp} \tag{6.7}$$

$$r = \frac{tp}{tp+fn} \tag{6.8}$$

$$f_1 = 2 * \frac{p*r}{p+r} \tag{6.9}$$

Equations 6.7, 6.8, 6.9,, where p is the precision metric and r is the recall metric, demonstrate how these metrics can be used to assess the efficiency of Spreaders classification. Finding fake news spreaders is the main objective of this project because they pose the greatest social danger as opposed to fact checkers who act normally. In the binary case, recall and precision are better suited than accuracy to deal with this problem. We will explain the findings of each tested model using these underlying assumptions.

The first line of Table 6.6 lists the results of our model, which is described in Section 6.2, using only the textual data gathered from user timelines as input.

The RF Fake News Spreader Classifier, a random forest with 100 estimators, and a Gini split criterion with no maximum depth are used in parallel to produce the results in row two. The Random Forest ensemble method creates a large number of decision trees and outputs the class that represents the mean prediction of all the individual trees. In this instance, there are only two classes: fake news spreader and real news checker. The number of features listed and described in Table 222 is the source of our choice for the max_feature parameter, which is 11. Every user identified in the CoAID extended dataset has their Twitter account information collected. The boolean features (protected, verified, default profile, and default profile image) are mapped to 1 or 0, while the string type features (location and created at) are transformed with one hot encoding. Then, each feature is min-max scaled and translated separately until it falls within the predetermined range of zero to one. It's intriguing to note that its metrics are more similar to the best-performing ones, serving as an index for the vast amount of data that each Twitter user's social media graph features contain.

The final outcomes of the Giachanou et al.[80] model applied to our CoAID extended dataset are displayed on the third line of Table 6.6. Because their work explicitly searches Spreaders and Checkers taking user related features into consideration as well as news related features, we considered their model to be the previous state of the art in this field. From each user's LIWC dictionary, they extract personality traits and psychological signals.

The results of our concatenation of tweet embeddings and the tabular data containing information about Twitter users in the penultimate layer of the FNSC stacked neural network are reported in the last line by the Mixed Fake News Spreader Classifier. The additional data from the user's account does not raise the overall score in the current configuration. The findings shown in Table 6.6 and the comments that follow respond to the RQ1 listed at the beginning of this Chapter. We actually show that tweet encoding based on transformers and deep learning is successful in classifying fake news spreaders because it achieves results above 80% in precision, recall, and f1. Additionally, it is better in terms of solutions utilizing conventional machine learning, such as the RF Fake News Spreader Classifier previously mentioned.

Attribute	**Data Type**	**Twitter Attribute Description**
location	string	The user-defined location for this account's profile
protected	boolean	When true, indicates that this user has chosen to protect their tweets
verified	boolean	When true, indicates that the user has a verified account
followers count	integer	The number of followers this account currently has
friends count	integer	The number of users this account is following
listed count	integer	The number of public lists that this user is a member of
favourites count	integer	The number of tweets this user has liked in the account's lifetime
statuses count	integer	The number of tweets (including retweets) issued by the user
created at	string	The UTC datetime that the user account was created on Twitter
default profile	boolean	User has not altered the theme or background of his profile
default profile image	boolean	Not uploaded profile image

Table 6.7 Twitter account user information used for classification with the RF Fake News Spreader Classifier as described in Section 6.2.

6.4 Discussion

In Section 111, we use the precision, recall, and f1 data from Table 111 to demonstrate how our results advance the state of the art. f1 increases by 4%, recall increases by 6%, and precision increases by 3%. We also emphasized the significance of social media graph features and latent information found in written text. In spite of the fact that these two sources of data are useful for tasks involving the classification of fake news spreaders, their combined performance in a single deep learning architecture is not as good. Further research should be done in this area, according to this finding. In fact, we demonstrated that deep learning and sentence encoding based on transformers are the most efficient methods for categorizing disseminators of false information in the context of COVID-19 news, which answers our RQ1. Regarding data retrieval, there is still a crucial specification that needs to be made. Thanks to the Twitter Academy License[15], which the social media platform recently made available for authorized research projects, we were able to gather more tweets. This license enables us to collect information that is typically limited to an account with a standard developer license. Because of privacy concerns, we released our CoAID extended dataset with a mapped user id to protect the authors' anonymity. Given the important subject matter and the limited access we were granted, this requirement was even more important. A broader application with more general topics should be performed in order to increase the validity of this research project since the original collection of fake and real news as presented in the CoAID dataset is all related to COVID-19 topics. The answers to RQ2 include the expansion of the CoAID dataset to include the collection of Twitter timelines from more than 12k users, the phase of stance detection to determine whether a user supports the fake news he shares, and the labeling of users as fake news checkers or spreaders. The extended CoAID dataset we made available serves as a gold standard for categorizing fake news disseminators in the context of COVID-19 data. Finally, it's important to note that we do not want to implement a censorship process; rather, acting from a user-centered perspective, the next step is to launch awareness-raising campaigns aimed at those users. Parallel to this specification, we affirm that while the fact-checking process necessitates intensive human oversight, it is much simpler to scale and automate the detection of sensitive users as the target of guidelines suggestions.

[15] https://developer.twitter.com/en/solutions/academic-research

6.5 Conclusion

We looked at how dangerous it is for society when false information spreads from the viewpoint of the user. We created a linguistic model to categorize users as real news verifyers or fake news spreaders. In this project, we described a language model that analyzes user-written social media posts, converts them into high-dimensional arrays using a transformer-based encoder, and max pools them to produce a user-related high level embedding. The final layer of our FNSC (Fake News Spreader Classifier) stacked neural network then uses this embedding to carry out the classification task. In terms of f1 score, we outperformed Giachanou et al.'s [80] actual state-of-the-art by 4%. We provided an answer to the RQ1 by demonstrating that, even though the social media graph features have a significant influence on the classification of fake news spreaders, they are not as effective as the features that were extracted from the sole text. In fact, when processed in batches as user embeddings, transformer-based tweet encoding and deep learning are successful at classifying fake news spreaders in the context of COVID-19 news. The findings of this study establish a new classification criterion for the development of misinformation deterrence strategies. The development and publication of a gold standard for categorizing fake news disseminators in the context of COVID-19 news is the second major contribution of our work. We developed the extended CoAID dataset to respond to the RQ2 because there isn't a de facto standard for user-centered classification of fake news spreaders. To determine whether bots' semantics differ from those of actual users who spread false information, we plan to tackle the issue of bot detection in the field of misinformation dissemination in a subsequent work. In addition to the feature relating to text embeddings, we also want to develop a real-time analysis tool to track users who spread false information on social media by combining features from social media metrics, personality metrics, and sentiment. In order to further expand our CoAID extended dataset, we planned to expand the dataset to include topics besides COVID-19, which were collected in the original CoAID dataset. Our goal is to create automated tools and conversational agents that will support human efforts to counteract misinformation and to encourage good behavior in users who have previously spread false information or who resemble fake news spreaders.

Chapter 7

Educational Chatbot to Support Question Answering on Slack

The results obtained so far show the potential of a linguistic model to assess certain user personal characteristics. As a last contribution to this research area we start the process to move the language processing toward language generation and user active interaction. This seminal work acts as a basement for future developments in which sentence generation is tuned in accordance to the user the chatbot is interacting with. We focus our work on the university courses because we have a directly accessible dataset of student/teacher textual interactions.

Chatbots for education open up new avenues for enhancing instructional strategies. Recent advancements in natural language processing and comprehension have made it possible to create virtual assistants that can comprehend queries presented as unstructured data. They use knowledge bases to generate an appropriate response. In real-world situations, such as in university courses, this work suggests an accurate solution that is simple to test. Actually, the educational chatbot's installation and training on already-existing digital workplaces are made easier by the use of RASA and Slack technologies. We assess the efficiency of our approach, which builds a stacked neural network to recognize intent with 0.99 accuracy.

The COVID-19 pandemic made digital tools even more necessary in the educational system. Online lectures led by distant lecturers were attended by students. In this situation, universities look for efficient digital tools to enable them to continue providing their services without losing contact with students. The deployment of virtual assistants into real-world scenarios has been encouraged by this context, demonstrating both their potential and their actual limitations. Regardless, research has shown that artificial intelligence is being rapidly adopted in the educational sector[122]. The chatbot system [123] is specifically one of the most well-liked AI tools in education. A chatbot is a virtual assistant that can interact with user input, process it, and provide a response. It also encourages the conversation to continue. There is the potential to assist teachers in responding to questions that are occasionally repeated or that have already been addressed in earlier iterations of the course in a situation where students must communicate with professors remotely and in tandem with the growing audience in courses. The best tool for this particular task is an educational chatbot. To prototype and develop the tools suggested by Cunningham et al. [124], there have already been suggested solutions and practical steps in the literature. Numerous studies conducted recently have shown that the use of chatbots in education has a positive impact on both the way that content is personalized and how well students learn[125], [126], [127]. Existing research suggests intriguing prototypes that call for the development of engineering skills and a deeper understanding of user experience in order to produce chatbots that are useful. In this work, we concentrate on the simplicity of installation of a solution on Slack, a popular digital workspace that permits user interaction. Additionally, we use RASA Open Source as a server side environment, which reduces the learning curve for configuring both the NLU (Natural Language Understanding) components of the communication pipeline and enables the teacher to expand the dataset of the educational chatbot by simply responding to the inquiry directly in the Slack channel. By responding to these research inquiries, we created this work.

- RQ1: Which dataset can be used in the educational system to train and test a chatbot?
- RQ2: Which natural language understanding model is most accurate at answering questions in a context of education?
- RQ3: Is it possible for a chatbot running on a popular chat platform to comprehend a student's intent during a conversation that goes through multiple turns?

The rest of this essay is organized as follows: in Section 7.1, we outline the suggested method for preparing and categorizing students' questions. We provide a preliminary experimental assessment of our method in Section 7.2. Finally, in Section 7.3, we wrap up by summarizing the results of the experiment and outlining further research.

7.1 Approach

The steps involved in data retrieval and manipulation, the decisions made when implementing the model, and the entire pipeline for understanding and processing natural language are all covered in the sections that follow. Last but not least, we outline the parameters that must be chosen for the client-server architecture and model optimization.

7.1.1 Dataset

The questions and responses were gathered from the Object Oriented Programming course at Politecnico di Torino in the spring semester of 2020, which was held during the COVID-19 pandemic. For reasons of privacy, all student information has been completely anonymized. The Slack web application supported remote assistance during this course. The course's workspace is divided into two main sections: theory and laboratories. Questions about frontal theory lessons are covered in the first channel, while exercises from the course's practical sessions are covered in the second. The experiment's chosen topic is inheritance, which has 53 subtopics and 300 questions that have been addressed. In Table 7.1, the intent labels that were generated from the subtopics and the responses are listed. This is our model's training data, and for the testing portion, we used the same topic questions from the 2021 version of the same course. In the supervised classification setting, the intent serves as the labels. Emojis, urls, and references to earlier questions in the channel with tags have all been deleted, and all handles citing users in the channel have also been removed from all of the questions. Finally, answers have been classified as actions while questions have been classified as intents. In accordance with a predetermined format[1], questions and answers are paired. The pairing format will be referred to as a "story" for the remainder of this essay.

[1] https://rasa.com/docs/rasa/stories/

Table 7.1 From the dataset gathered for the object-oriented programming course, a list of intents on the topic of inheritance was created. They originated from a set of 300 questions and corresponding pairs of answers.

Intent name		
Inheritance in general	Comparable	Comparator
Override	Polymorphism	Iterator-Iterable
Dynamic binding	Observer-Observable	Lambda Functions
Casting	Downcasting	Upcasting
Class/Attribute visibility	Methods Reference	Instance of keyword

This dataset responds to RQ1, our initial research question, and it also demonstrates how to construct a dataset for our model to process.

7.1.2 Question Answering Model

Our solution uses ConveRT (Conversational) as the neural network model. Symbols from the film Transformers). Henderson et al. [128] suggest this dual encoder built on a Transformer that is focused on conversational tasks. In contrast to Devlin et al.'s [121] BERT (Bidirectional Encoder Representations from Transformers), they create a model that is quicker and lighter to be memorized and trained. They achieve this result by combining 8-bit embedding quantization and quantization-aware training, sub-word level parameterization, and pruned self-attention along with a multi-context specialization that enables the long term memory mechanisms. The single-context ConveRT model, which serves as the foundation for the multi-context solution, is shown in Figure 7.1. Multi-turn conversations can actually only be handled by the multi-context solution. In our use case scenario, after receiving the initial response, students can interact with the chatbot and request additional clarification. In a similar vein, they can continue the conversation with additional questions. We choose to include the multi-context ConveRT in our solution because, as we demonstrate in the validation Section 7.2, it outperforms the BERT and SpaCy [2] ones and because it runs more quickly on our cloud-based solution.

After being given a question, ConveRT aims to choose the correct response. Either the question is present in our dataset or it is not. In any case, our model attempts to match the question based on similarity and automatically prints the

[2] https://spacy.io/models

paired response. If the student finds the response unsatisfactory, he has two options: rephrase the question or send it to the professor.

7.1.3 Model Hyperparameters

We use a configuration module and its parameters to fine-tune the ConveRT solutions for question-answer matching. The decisions made and their significance are discussed in the sections that follow.

- *Language* specifies the adoption of a specific idiom, we use the Italian language because the course in our experiment has been taught in Italian.
- *Tokenizer* explains how the words are translated into tokens before the encoding into embeddings. We select the ConveRT Tokenizer that is the default one given by the ConveRT model.
- *LexicalSyntacticFeaturizer* creates lexical and syntactic features for a user message to support entity extraction.
- *CountVectorsFeaturizer* creates a bag of words representing user messages, intents, and responses.
- *char wb* with *min ngram: 1* and *max ngram: 10* sets the smallest and largest ngram adopted by the tokenizer to produce tokens.
- *DIETClassifier* Dual Intent Entity Transformer (DIET) used for intent classification and entity extraction. DIET classifier is set with 300 *epochs*. One epoch is equal to one forward pass and one backward pass of all the training examples. The lower the number of epochs the faster the model is trained. DIET classifier is also tuned with the *constraint similarities* set on *True*. This parameter applies a sigmoid cross entropy loss overall similarity terms. This helps in keeping similarities between input and negative labels to smaller values. This should help in better generalization of the model to real world test sets.

There is yet another set of hyperparameters that must be set. They are referred to as policies. At each stage of a conversation, our educational chatbot uses policies to determine what to do.

- *Core fallback threshold: 0.1* helps ensure that low confidence messages are handled gracefully, giving the educational chatbot the option to either respond with a default message or attempt to disambiguate the user input.
- *TEDPolicy* Transformer Embedding Dialogue (TED) Policy is a multi-task architecture for next action prediction and entity recognition, with *max history: 5* it controls how much dialogue history the model looks at to decide which action to take next.

7.1.4 Client Server Architecture

We outline the architectural decisions we made when creating our educational chatbot in this section, along with instructions on how to set up the environment where it will be used. Between students and professors, we use the Slack application [3] as a channel of communication. The web client function is performed by it. It also houses our chatbot for education. This tool is widely used as a potent workspace to communicate with coworkers and students in both businesses and academic settings. Additionally, it supports integration with plugins and bots. On the server side of the communication, the natural language processing is handled concurrently using RASA Open Source.

We start the bot's development by building a Slack application. In order to create a new app, we go to https://api.slack.com/apps and select that option. Next, we decide on the application's name and the Slack channel where the bot will engage with users. The "OauthandPermission" section of the "Scopes" menus allows us to select the User Token Scopes and Bot Token Scopes that the bot will use to access the workspace. The scopes regulate the abilities and privileges of the bot and the users who use tokens. We install the app (our educational chatbot) on the workspace after configuring the tokens. Once the connection and installation are complete, we receive the Oauth Token. When managing connections on an app, the bot needs the Bot Token. . We develop a RASA project and specify the conversational capabilities of our context-aware conversational chatbot. Slots, stories, and policies are the three main tasks that must be configured.

[3] https://slack.com/

- *Slots*[4] are the assistant's memory. They store pieces of information that the bot needs to refer to later and can direct the conversation flow based on slot events.
- *Stories* are examples of conversations between a user and the bot. In stories different patterns of interaction are described.
- *Machine Learning Policies* help the chatbot to better predict the response in unseen conversation paths.

RASA X is the last element that makes up RASA. It is an instrument for CDD (Conversion Driven Development). All conversations are automatically saved, and the chatbot and its embedded NLU model are improved as a result.

Figure 7.2 outlines the system's entire pipeline of usage. In the Slack channel where the educational chatbot is set up, a student posts a question. The RASA server receives the question, which is then processed to determine the intent and, as a result, the best response. If the search is unsuccessful, the professor is contacted, who updates the dataset and broadens the chatbot's knowledge base by responding to the question. The solution will in any case be given to the student.

7.2 Results

We evaluate the efficiency and accuracy of our framework's metrics computation over the functionality of intent recognition. A multi-label supervised classification is the issue under evaluation. We compare the recall, precision, f1-score, and accuracy of the three main models, BERT, SpaCy, and ConveRT, as shown in equations 7.1, 7.2, 7.3 and 7.4. We use 80% of the dataset as a training set and 20% as a test set. In Table 7.2, the outcomes are listed.

$$recall = \frac{tp}{tp + fn} \tag{7.1}$$

$$precision = \frac{tp}{tp + fp} \tag{7.2}$$

[4] https://rasa.com/docs/rasa/domain#slot-types

$$f1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{7.3}$$

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{7.4}$$

Table 7.2 Evaluation metrics computed with a 80/20 train/test set setup for the task of intent recognition.

Model	Recall	Precision	f1-score	Accuracy
ConveRT	0.93	0.91	0.92	0.93
BERT	0.92	0.89	0.90	0.92
SpaCy	0.92	0.88	0.89	0.92

A fivefold cross validation method was used to test 174 conversations with 921 distinct actions. RASA provides in its suite the option to simulate user inputs that test various combinations of actions and conversational turns. Table 7.3 lists the findings from an analysis of the chatbot's performance over 174 attempts and 921 possible action combinations, measuring the percentage of correct conversations it was able to sustain.

Table 7.3 Evaluation metrics computed with a 5 fold cross validation setup for the task of correct conversation evolution. We tested 174 simulated conversations and their 921 paired actions.

Model	% of Correct Conversation	Precision	f1-score	Accuracy
ConveRT	0.999	0.999	0.999	0.999
BERT	0.977	0.998	0.997	0.997
SpaCy	0.960	0.994	0.993	0.994

The obtained results address both the task of intent recognition and the proper conversational decision-path choice in their answers to RQ2 and RQ3, respectively. We demonstrate that ConveRT is the model that performs the best in tasks with accuracy above 0.93 and f1-score above 0.92. These results were achieved using the Italian language because our dataset was in Italian. It is possible that using other languages, BERT or SpaCy could outperform the ConveRT model because in this case, the switch between various configurations is already set up.

7.3 Conclusion

This work introduces a new chatbot system to assist with question-answering in education, particularly helpful for courses with a large number of students and in remote teaching due to pandemics. We offer a pedagogical chatbot that can be easily installed on Slack. The ConveRT Natural Language Understanding model outperformed the other baseline models in both intent recognition and making the right conversational decisions, demonstrating the server side of our framework's accuracy of 0.99. This model was developed in RASA Open Source and Rasa X. We tested the model on questions created in 2021 and trained it using the course Slack channel, achieving high accuracy in response identification. The system could be used for other subjects and in teaching courses, and after it has been implemented, a survey on real-time interaction during a semester of study could be conducted. These are additional improvements to this work. Adoption of multimedia content and spoken conversation interactions are upcoming projects in this field.

Fig. 7.1 ConveRT dual-encoder model with a single context architecture. At various network layers (e.g. the final rx, or the hx) to other tasks like value extraction or intent detection. It should be noted that the model employs two distinct feed-forward network (FFN) layers: 1) feed-forward 1 is the traditional FFN layer also utilized by Vaswani et al. [74], and 2) the feed-forward 2 network, which has three fully connected nonlinear feed-forward layers and a linear layer that maps to the final encodings hx and hy, does not share parameters with the feed-forward 1 network.

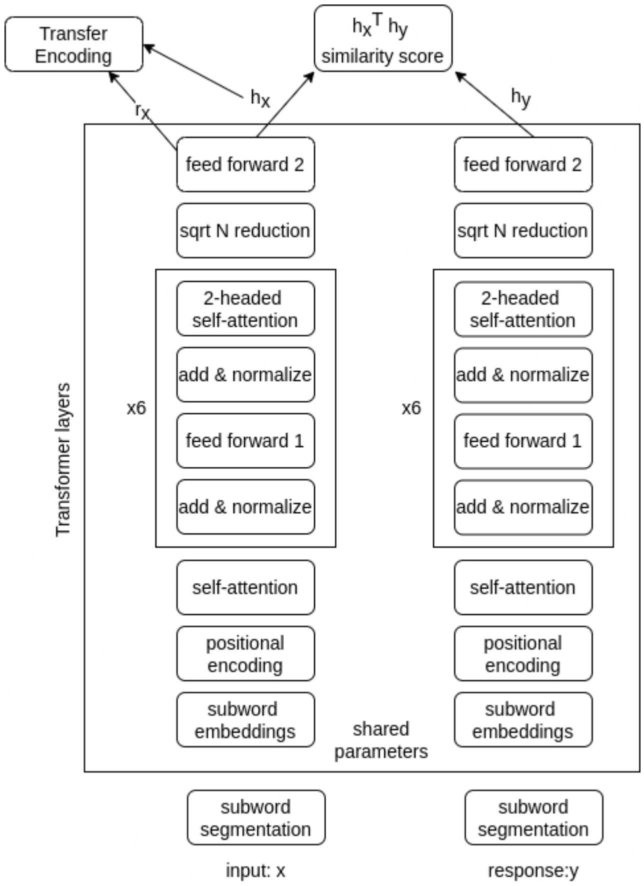
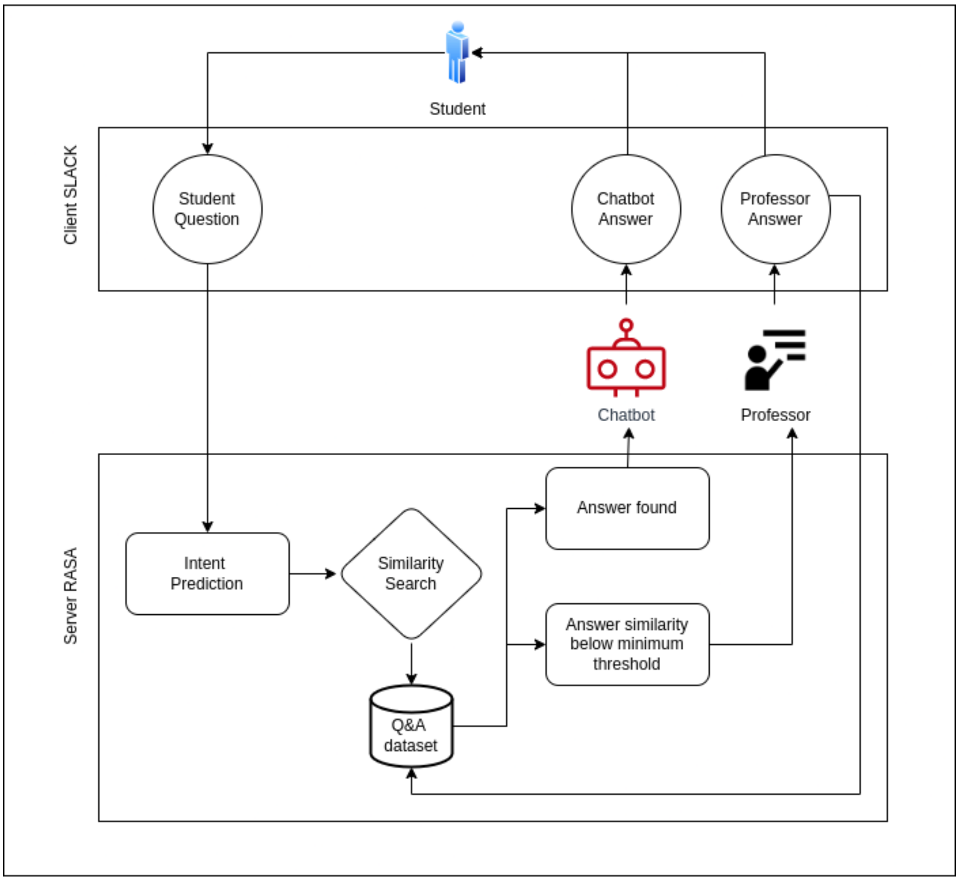


Fig. 7.2 Block diagram illustrating the steps taken from the student's initial question on the Slack course channel through the RASA server's intent recognition and search for an answer. If this stage is successful, the chatbot will post the response on Slack or will send the student's question to the professor on Slack. The interaction is saved in the update dataset once it has been satisfactorily answered for use in the future.



Chapter 8

Conclusions

In conclusion, this thesis has presented a comprehensive study on the application of natural language processing and machine learning techniques for various tasks related to social media analysis. We proposed novel and effective solutions for the estimation of personality traits, mining micro-influencers, automated classification of fake news spreaders, and development of an educational chatbot. The research presented in this thesis has demonstrated the potential of these techniques for understanding and predicting user behavior on social media platforms, identifying micro-influencers, and identifying fake news spreaders. Additionally, it also demonstrates the potential of chatbot as an educational tool.

The proposed solutions were thoroughly evaluated using real-world data and benchmark datasets, and the results demonstrate the effectiveness of the proposed methods. The proposed model for personality traits estimation achieved superior performance compared to existing state-of-the-art models, and the MIMIC model for micro-influencer identification achieved high precision and recall. The automated classifier for fake news spreaders also performed well in terms of precision and recall. The educational chatbot was evaluated in a real-world classroom setting, and the results showed that it can effectively assist students in their learning process and improve their understanding of the subject matter.

As for future work, there are several directions that can be pursued based on the research presented in this thesis. One area of future research is to further improve the performance of the proposed models by incorporating more data and features. Additionally, the proposed models can be fine-tuned for specific domains and languages, to improve their performance in real-world applications. Another area of future research is to develop more advanced methods for identifying fake news spreaders, such as using deep learning techniques and incorporating external sources of information. Moreover, more experiments can be done to evaluate the effectiveness of the educational chatbot in different domains and settings. Furthermore, the research can be extended to other social media platforms to evaluate the robustness of the proposed methods.

In addition, it is important to consider the ethical and societal implications of these techniques and the potential impact they may have on individuals and society as a whole. Therefore, future research should also focus on exploring the ethical implications of these techniques and their potential impact on individuals and society, as well as the development of guidelines for their responsible use.

Overall, this thesis has contributed to the field of social media analysis by proposing novel and effective solutions for various tasks, and has provided a foundation for future research in this area. The research presented in this thesis has the potential to improve the targeting of marketing campaigns, identify misinformation on social media platforms, and help students to learn more effectively.

8.1 Discussion on different personality models

The thesis work on personality detection can certainly be adapted to other models beyond the Five Factor Model (FFM) and the basic human values framework by Schwartz, such as the Myers-Briggs Type Indicator (MBTI) or other personality frameworks.

The FFM and Schwartz's values framework are widely used models for understanding personality traits and values, respectively. However, there are several alternative models and theories that offer different perspectives on personality.

Some examples include:

Myers-Briggs Type Indicator (MBTI): The MBTI is a popular model that categorizes individuals into one of 16 personality types based on four dimensions: extraversion/introversion, sensing/intuition, thinking/feeling, and judging/perceiving. This model focuses on individual differences in preferences and cognitive styles.

Holland’s Six Personality Types: Developed by John L. Holland, this model categorizes individuals into one of six personality types based on their interests and preferences: realistic, investigative, artistic, social, enterprising, and conventional. It is often used in career counseling and vocational psychology.

Enneagram: The Enneagram is a personality model that categorizes individuals into one of nine types based on their core motivations, fears, and desires. It offers insights into individuals’ patterns of thinking, feeling, and behaving, and provides a framework for personal growth and self-awareness.

HEXACO Model: The HEXACO model expands upon the FFM by including an additional factor called Honesty-Humility. It categorizes individuals into six dimensions: honesty-humility, emotionality, extraversion, agreeableness, conscientiousness, and openness to experience.

When adapting a thesis work on personality detection to different models, you would need to modify the feature representation and classification techniques to align with the constructs and dimensions of the new model. For example, if working with the MBTI, you would focus on the four MBTI dimensions and develop features that capture relevant indicators of each dimension. Similarly, for other models, you would need to identify the core constructs and design features that capture those constructs.

Keep in mind that adapting the thesis work to different models may require additional data collection or annotation efforts to ensure you have labeled data that aligns with the new model’s categories or dimensions. Additionally, you may need to explore different classification algorithms or modify existing algorithms to suit the specific requirements of the new model.

By adapting the thesis work to different personality models, you can contribute to the understanding of diverse perspectives on personality and provide valuable insights into how different models can be used for personality detection and analysis.

References

[1] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, 1(1):arXiv:1810.04805, Oct 2018.

[3] Giulio Carducci, Giuseppe Rizzo, Diego Monti, Enrico Palumbo, and Maurizio Morisio. Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning. *Information*, 9(5):127, may 2018.

[4] Daniele Quercia, Michal Kosinski, David Stillwell, and Jon 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*, pages 180–185. IEEE, 2011.

[5] Firoj Alam, Evgeny A. Stepanov, and Giuseppe Riccardi. Personality traits recognition on social network - facebook. *International AAAI Conference on Web and Social Media*, 2013.

[6] P. Raghavan C. D. Manning and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[7] Chaudhary S.; Sing R.; Hasan S. T.; Kaur I. A comparative study of different classifiers for myers-brigg personality prediction model. *IRJET*, 5:1410–1413, 2018.

[8] Katharine C Briggs. *Myers-Briggs Type Indicator*. Consulting Psychologists Press Palo Alto, CA, 1976.

[9] Di Xue, Lifa Wu, Zheng Hong, Shize Guo, Liang Gao, Zhiyong Wu, Xiao-Feng Zhong, and Jianshan Sun. Deep learning-based personality recognition from text posts of online social networks. *Applied Intelligence*, 48:4232–4246, 2018.

[10] Fei Liu, Scott Nowson, and Julien Perez. A language-independent and compositional model for personality trait recognition from short texts. In *EACL*, 2016.

[11] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, Mar 2017.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, 1(1), Jan 2013.

[13] Fabio Celli. Unsupervised personality recognition for social network sites. In *ICDS 2012*, 2012.

[14] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500, 2007.

[15] Eleanna Kafeza, Andreas Kanavos, Christos Makris, and Pantelis Vikatos. T-pice: Twitter personality based influential communities extraction system. In *Proceedings - 2014 IEEE International Congress on Big Data, BigData Congress 2014*, 07 2014.

[16] Xiangguo Sun, Bo Liu, Qing Meng, Jiuxin Cao, Junzhou Luo, and Hongzhi Yin. Group-level personality detection based on text generated networks. *World Wide Web*, Sep 2019.

[17] Shuotian Bai, Tingshao Zhu, and Li Cheng. Big-Five Personality Prediction Based on User Behaviors at Social Network Sites. *arXiv e-prints*, page arXiv:1204.4809, Apr 2012.

[18] L. C. Lukito, A. Erwin, J. Purnama, and W. Danoekoesoemo. Social media user personality classification using computational linguistic. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–6, Oct 2016.

[19] Francisco Iacobelli and Aron Culotta. Too neurotic, not too friendly: Structured personality classification on textual data. In *AAAI Workshop - Technical Report*, 01 2013.

[20] James W Pennebaker and Laura A King. Linguistic styles: Language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296, 1999.

[21] Shlomo Argamon, Sushant Dhawle, Moshe Koppel, and James W. Pennebaker. Lexical predictors of personality type. In *In proceedings of the joint annual meeting of the Interface and the Classificaton society of North America*, Cincinnati, OH, USA, 2005. In Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America.

[22] James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577, 2003.

[23] Jon Oberlander and Alastair J Gill. Language with character: A stratified corpus comparison of individual differences in e-mail communication. *Discourse Processes*, 42(3):239–270, 2006.

[24] Upendra Kumar, Aishwarya N. Reganti, Tushar Maheshwari, Tanmoy Chakroborty, Björn Gambäck, and Amitava Das. Inducing personalities and values from language use in social network communities. *Information Systems Frontiers*, 20(6):1219–1240, 12 2018.

[25] James W. Pennebaker, Matthias R. Mehl, and Kate G. Niederhoffer. Psychological aspects of natural language use: our words, our selves. *Annual Review of Psychology*, 54(1):547–577, 2003.

[26] James W. Pennebaker and Laura A. King. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6):1296–1312, 1999.

[27] Max Weisbuch, Zorana Ivcevic, and Nalini Ambady. On being liked on the web and in the “real world”: Consistency in first impressions across personal webpages and spontaneous behavior. *Journal of Experimental Social Psychology*, 45(3):573–576, 2009.

[28] M. Su, C. Wu, and Y. Zheng. Exploiting turn-taking temporal evolution for personality trait perception in dyadic conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):733–744, April 2016.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[30] Isabel Anger and Christian Kittl. Measuring influence on twitter. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, New York, NY, USA, 2011. ACM Press.

[31] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone’s an influencer: Quantifying influence on Twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74, New York, NY, USA, 2011. ACM Press.

[32] Carolina Bigonha, Thiago N. C. Cardoso, Mirella M. Moro, Marcos A. Goncalves, and Virgilio A. F. Almeida. Sentiment-based influence detection on Twitter. *Journal of the Brazilian Computer Society*, 18(3):169–183, 2011.

[33] Christine Kiss and Martin Bichler. Identification of influencers - measuring influence in customer networks. *Decis. Support Syst.*, 46(1):233–253, December 2008.

[34] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[35] Easley David and Kleinber Jon. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World.* Cambridge University Press, Cambridge, 2010.

[36] Alpaslan Burak Eliacik and Nadia Erdogan. Influential user weighted sentiment analysis on topic based microblogging community. *Expert Syst. Appl.*, 92(C):403–418, February 2018.

[37] Jilin Chen, Gary Hsieh, Jalal U. Mahmud, and Jeffrey Nichols. Understanding individuals' personal values from social media word use. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work &#38; Social Computing*, CSCW '14, pages 405–414, New York, NY, USA, 2014. ACM.

[38] Amit Mandelbaum and Adi Shalev. Word embeddings and their use in sentence classification tasks. *CoRR*, abs/1610.08229, 2016.

[39] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[40] Roy Schwartz, Roi Reichart, and Ari Rappoport. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China, July 2015. Association for Computational Linguistics.

[41] Shazia Tabassum, Fabiola Pereira, Sofia Fernandes, and João Gama. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8:e1256, 04 2018.

[42] Gregory Albery, Lucinda Kirkpatrick, Josh Firth, and Shweta Bansal. Unifying spatial and social network analysis in disease ecology. *The Journal of animal ecology*, 90, 09 2020.

[43] Mehran Badin Dahesh, Gholamali Tabarsa, Mostafa Zandieh, and Mohammadreza Hamidizadeh. Reviewing the intellectual structure and evolution of the innovation systems approach: A social network analysis. *Technology in Society*, 63:101399, 11 2020.

[44] Man Hung, Evelyn Lauren, Eric Hon, Wendy Birmingham, Julie Xu, Sharon Su, Shirley Hon, Jungweon Park, Peter Dang, and Martin Lipsky. Social network analysis of covid-19 sentiments: Application of artificial intelligence. *Journal of Medical Internet Research*, 22:e22590, 08 2020.

[45] Jooho Kim and Makarand Hastak. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, 38:86–96, 02 2018.

[46] Seungil Yum. Social network analysis for coronavirus (covid-19) in the united states. *Social Science Quarterly*, 101(4):1642–1647, 2020.

[47] Fabián Riquelme and Pablo González-Cantergiani. Measuring user influence on twitter: A survey. *Journal of Information Processing and Management*, 52, 04 2016.

[48] Stefan Räbiger and Myra Spiliopoulou. A framework for validating the merit of properties that predict the influence of a twitter user. *Expert Systems with Applications*, pages –, 01 2014.

[49] Linyuan Lu, Duanbing Chen, Xiaolong Ren, Qian-Ming Zhang, Yi-Cheng Zhang, and Tao Zhou. Vital nodes identification in complex networks. *ArXiv*, abs/1607.01134, 2016.

[50] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 591–600, New York, NY, USA, 2010. Association for Computing Machinery.

[51] Xin Chen. Critical nodes identification in complex systems. *Complex & Intelligent Systems*, 1, 02 2016.

[52] Changjun Fan, Zeng li, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence*, 2:1–8, 06 2020.

[53] Iris Roelens, Philippe Baecke, and D.F. Benoit. Identifying influencers in a social network: The value of real referral data. *Decision Support Systems*, 91, 07 2016.

[54] Tian Gan, Shaokun Wang, Meng Liu, Xuemeng Song, Yiyang Yao, and Liqiang Nie. Seeking micro-influencers for brand promotion. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1933–1941, New York, NY, USA, 2019. Association for Computing Machinery.

[55] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling, 2016.

[56] Benyamin Bashari and Ehsan Fazl-Ersi. Influential post identification on instagram through caption and hashtag analysis. *Measurement and Control*, 53:002029401987748, 01 2020.

[57] Cheng Zheng, Qin Zhang, Sean Young, and Wei Wang. On-demand influencer discovery on social media. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2337–2340, New York, NY, USA, 2020. Association for Computing Machinery.

[58] Yun-Bei Zhuang, Zhi-Hong Li, and Yun-Jing Zhuang. Identification of influencers in online social networks: measuring influence considering multidimensional factors exploration. *Heliyon*, 7:e06472, 04 2021.

[59] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, May 2017.

[60] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[61] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018.

[62] Massimo Stella, Emilio Ferrara, and Manlio De Domenico. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, 115(49):12435–12440, 2018.

[63] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378, 2019.

[64] Andrew Guess, Jonathan Nagler, and Joshua Tucker. Less than you think: Prevalence and predictors of fake news dissemination on facebook. *Science advances*, 5(1):eaau4586, 2019.

[65] Gordon Pennycook and David G Rand. Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proceedings of the National Academy of Sciences*, 116(7):2521–2526, 2019.

[66] Chengcheng Shao, Pik-Mai Hui, Lei Wang, Xinwen Jiang, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. Anatomy of an online misinformation network. *PloS one*, 13(4):e0196087, 2018.

[67] Swapnil Dhamal. Effectiveness of diffusing information through a social network in multiple phases. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[68] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250, 2010.

[69] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[70] Jing Zhang, Jie Tang, Juanzi Li, Yang Liu, and Chunxiao Xing. Who influenced you? predicting retweet via social influence locality. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):1–26, 2015.

[71] Jing Guo, Peng Zhang, Chuan Zhou, Yanan Cao, and Li Guo. Personalized influence maximization on social networks. In *Proceedings of the 22nd ACM international conference on Information and Knowledge Management*, pages 199–208, 2013.

[72] Osama Mansour and Nasrine Olson. Interpersonal influence in viral social media: A study of refugee stories on virality. In *Proceedings of the 8th International Conference on Communities and Technologies*, pages 183–192, 2017.

[73] Maria Teresa Borges-Tiago, Flavio Tiago, and Carla Cosme. Exploring users' motivations to participate in viral communication on social media. *Journal of Business Research*, 101:574–582, 2019.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[75] Stefan Stieglitz and Linh Dang-Xuan. Political communication and influence through microblogging–an empirical analysis of sentiment in twitter messages and retweet behavior. In *2012 45th Hawaii international conference on system sciences*, pages 3500–3509. IEEE, 2012.

[76] Shan Jiang and Christo Wilson. Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW), nov 2018.

[77] Laura Burbach, Patrick Halbach, Martina Ziefle, and André Calero Valdez. Who shares fake news in online social networks? In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pages 234–242, 2019.

[78] Craig Ross, Emily S Orr, Mia Sisic, Jaime M Arseneault, Mary G Simmering, and R Robert Orr. Personality and motivations associated with facebook use. *Computers in human behavior*, 25(2):578–586, 2009.

[79] Jannica Heinström. Five personality dimensions and their influence on information behaviour. *Information research*, 9(1):9–1, 2003.

[80] Anastasia Giachanou, Esteban A Ríssola, Bilal Ghanem, Fabio Crestani, and Paolo Rosso. The role of personality and linguistic patterns in discriminating between fake news spreaders and fact checkers. In *International Conference on Applications of Natural Language to Information Systems*, pages 181–192. Springer, 2020.

[81] James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The development and psychometric properties of liwc2007. 2007. *URL: http://liwc. net/index. php [accessed 2015-09-14][WebCite Cache ID 6bX6QdwIO]*, 2015.

[82] Yang Li, Amirmohammad Kazemeini, Yash Mehta, and Erik Cambria. Multitask learning for emotion and personality traits detection. *Neurocomputing*, 493:340–350, 2022.

[83] Sahraoui Dhelim, Nyothiri Aung, Mohammed Amine Bouras, Huansheng Ning, and Erik Cambria. A survey on personality-aware recommendation systems. *Artif. Intell. Rev.*, 55(3):2409–2454, mar 2022.

[84] Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839, Marseille, France, June 2022. European Language Resources Association.

[85] Sanjay Kumar, Akshi Kumar, Abhishek Mallik, and Sakshi Dhall. Opinion leader detection in asian social networks using modified spider monkey optimization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5), may 2023.

[86] Akshi Kumar, Saurabh Raj Sangwan, Adarsh Kumar Singh, and Gandharv Wadhwa. Hybrid deep learning model for sarcasm detection in indian indigenous language using word-emoji embeddings. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5), may 2023.

[87] Kumar Sanjay Kumar Akshi, Aggarwal Nipun. Sira: a model for propagation and rumor control with epidemic spreading and immunization for healthcare 5.0. *Soft Computing*, 27:4307 – 4320, 2023.

[88] Sanjay Kumar, Akshi Kumar, Abhishek Mallik, and Rishi Ranjan Singh. Optnet-fake: Fake news detection in socio-cyber platforms using grasshopper optimization and deep neural network. *IEEE Transactions on Computational Social Systems*, pages 1–10, 2023.

[89] Ingo Frommholz, Haider M Al-Khateeb, Martin Potthast, Zinnar Ghasem, Mitul Shukla, and Emma Short. On textual analysis and machine learning for cyberstalking detection. *Datenbank-Spektrum*, 16(2):127–135, 2016.

[90] Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.

[91] Andrew Neal, Gillian Yeo, Annette Koy, and Tania Xiao. Predicting the form and direction of work role performance from the big 5 model of personality traits. *Journal of Organizational Behavior*, 33(2):175–192, 2012.

[92] Robert R. McCrae and Paul T. Costa. Empirical and theoretical status of the five-factor model of personality traits. In *The SAGE Handbook of Personality Theory and Assessment: Volume 1 — Personality Theories and Models*, pages 273–294. SAGE Publications Ltd, Los Angeles, CA, USA, 2008.

[93] Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. Workshop on computational personality recognition: Shared task. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(2):2–5, Aug. 2021.

[94] Ernest C Tupes and Raymond E Christal. Recurrent personality factors based on trait ratings. *Journal of personality*, 60(2):225–251, 1992.

[95] John M Digman. Personality structure: Emergence of the five-factor model. *Annual review of psychology*, 41(1):417–440, 1990.

[96] Lewis R Goldberg. The structure of phenotypic personality traits. *American psychologist*, 48(1):26, 1993.

[97] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[98] James M. Joyce. *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[99] Francisco Manuel Rangel Pardo, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. Overview of the 3rd author profiling task at pan 2015. In *CLEF 2015 Evaluation Labs and Workshop Working Notes Papers*, pages 1–8, 2015.

[100] Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, and Benno Stein. Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. In *Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al.*, pages 750–784, 2016.

[101] Shalom H. Schwartz. An overview of the schwartz theory of basic values. *Online Readings in Psychology and Culture*, 2(1), 2012.

[102] Karen Freberg, Kristin Graham, Karen McGaughey, and Laura A. Freberg. Who are the social media influencers? a study of public perceptions of personality. *Public Relations Review*, 37(1):90 – 92, 2011.

[103] Monika Ewa Rakoczy, Amel Bouzeghoub, Alda Lopes Gancarski, and Katarzyna Wegrzyn-Wolska. In the search of quality influence on a small scale – micro-influencers discovery. In Hervé Panetto, Christophe Debruyne, Henderik A. Proper, Claudio Agostino Ardagna, Dumitru Roman, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, pages 138–153, Cham, 2018. Springer International Publishing.

[104] Oliver P. John, Alois Angleitner, and Fritz Ostendorf. The lexical approach to personality: A historical review of trait taxonomic research. *European Journal of Personality*, 2(3):171–203, 1988.

[105] Farhad Rahmanov, Muslum Mursalov, and Anna Rosokhata. Consumer behavior in digital era: impact of covid 19. *Marketing and Management of Innovations*, 5:243–251, 01 2021.

[106] Ratih Purbasari, Zaenal Muttaqin, and Deasy S. Sari. Digital entrepreneurship in pandemic covid 19 era: The digital entrepreneurial ecosystem framework. *Review of Integrative Business and Economics Research*, 10:114–135, 2021. Name - Capital IQ; Copyright - Copyright Society of Interdisciplinary Business Research 2021.

[107] Haudi Haudi, Erna Rahadjeng, Ruby Santamoko, Riyan Sisiawan Putra, Dwi Purwoko, Dewi Nurjannah, Intan Koho, Hadion Wijoyo, Ade Siagian, Yoyok Cahyono, and Agus Purwanto. The role of e-marketing and e-crm on e-loyalty of indonesian companies during covid pandemic and digital era. *Uncertain Supply Chain Management*, 20:1–12, 04 2022.

[108] Evelina Francisco, Nadira Fardos, Aakash Bhatt, and Gulhan Bizel. Impact of the covid-19 pandemic on instagram and influencer marketing. *International Journal of Marketing Studies*, 13:20, 05 2021.

[109] Jewon Lyu Maureen Lehto Brewster. Exploring the parasocial impact of nano, micro and macro influencers. In *Exploring the Parasocial Impact of Nano, Micro and Macro Influencers*, volume 77. Iowa State University Digital Press, 12 2020.

[110] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1096–1103, 04 2020.

[111] Ron Mokady, Amir Hertz, and Amit Bermano. Clipcap: Clip prefix for image captioning. ., 11 2021.

[112] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable

visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[113] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November 2020. Association for Computational Linguistics.

[114] Toan Ha, Stephen Schensul, Judy Lewis, and Stacey Brown. Early assessment of knowledge, attitudes, anxiety and behavioral adaptations of connecticut residents to covid-19. *medRxiv*, 2020.

[115] Quattrociocchi Cinelli, Valensise Galeazzi, Schmidt Brugnoli, Zollo Zola, and Scala. The covid-19 social media infodemic. *Scientific Reports*, 10(JOUR), oct 2020.

[116] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6086–6093, Marseille, France, may 2020. European Language Resources Association.

[117] Md Rafiqul Islam, Shaowu Liu, Xianzhi Wang, and Guandong Xu. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, 10(1), sep 2020.

[118] Maria Glenski, Tim Weninger, and Svitlana Volkova. Identifying and understanding user reactions to deceptive and trusted social news sources. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 176–181, Melbourne, Australia, jul 2018. Association for Computational Linguistics.

[119] Limeng Cui and Dongwon Lee. Coaid: Covid-19 healthcare misinformation dataset, 2020.

[120] Ahmet Aker, Leon Derczynski, and Kalina Bontcheva. Simple open stance classification for rumour analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 31–39, Varna, Bulgaria, sep 2017. INCOMA Ltd.

[121] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[122] Sofie Roos. Chatbots in education: A passing trend or a valuable pedagogical tool?, 2018.

[123] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Python-bot: A chatbot for teaching python programming. *Engineering Letters*, 29(1), 2020.

[124] Sam Cunningham-Nelson, Wageeh Boles, Luke Trouton, and Emily Margerison. A review of chatbots in education: practical steps forward. In *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate*, pages 299–306. Engineers Australia, 2019.

[125] Fabio Clarizia, Francesco Colace, Marco Lombardi, Francesco Pascale, and Domenico Santaniello. Chatbot: An education support system for student. In *International Symposium on Cyberspace Safety and Security*, pages 291–302. Springer, 2018.

[126] Bhavika R Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh. Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1525–1530. IEEE, 2017.

[127] Sharob Sinha, Shyanka Basak, Yajushi Dey, and Anupam Mondal. An educational chatbot for answering queries. In *Emerging Technology in Modelling and Graphics*, pages 55–60. Springer, 2020.

[128] Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online, November 2020. Association for Computational Linguistics.

Publication List

The studies discussed as part of the present dissertation, in which I have been either the main author or a co-author, have been published in the following conference proceedings or journals.

- Simone Leonardi, Diego Monti, Giuseppe Rizzo, and Maurizio Morisio. 2020. Mining micro-influencers from social media posts. In Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20). Association for Computing Machinery, New York, NY, USA, 867–874. doi: 10.1145/3341105.3373954
- Leonardi, Simone, Diego Monti, Giuseppe Rizzo, and Maurizio Morisio. 2020. "Multilingual Transformer-Based Personality Traits Estimation" Information 11, no. 4: 179. doi: 10.3390/info11040179
- Leonardi, Simone, Giuseppe Rizzo, and Maurizio Morisio. 2021. "Automated Classification of Fake News Spreaders to Break the Misinformation Chain" Information 12, no. 6: 248. doi: 10.3390/info12060248
- Simone Leonardi, Marco Torchiano (2022) Educational Chatbot to Support Question Answering on Slack, In: Methodologies and Intelligent Systems for Technology Enhanced Learning, 12th International Conference, Im Lecture Notes in Networks and Systems, vol. 580, pp. 20-25, 2022, doi:10.1007/978-3-031-20617-7
- Simone Leonardi and Luca Ardito (2022), MIMIC: a Multi Input Micro-Influencers Classifier, In Proceedings of ICSMADM 2022: 16. International Conference on Social Media Analysis and Data Mining, vol. 1, pp. 168-175, 2022, https://attachments.waset.org/22/ebooks/may-2022-in-barcelona-2022-05-25-11-51-04.pdf

Furthermore, in the context of my Ph.D. career, I have collaborated as a coauthor to the realization of the following studies.

- L. Ardito, R. Coppola, S. Leonardi, M. Morisio and U. Buy, Automated Test Selection for Android Apps Based on APK and Activity Classification, in IEEE Access, vol. 8, pp. 187648-187670, 2020, doi: 10.1109/ACCESS.2020.3029735.