Demographic information retrieval from text for subject characterization and market segmentation

Daniel Escobar Grisales

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

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

Asesor
MSc. Juan Camilo Vásquez Correa

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

Cita	Escobar Grisales [1]
Referencia **Estilo IEEE (2020)**	[1] D. Escobar Grisales, "Demographic information retrieval from text for subject characterization and market segmentation", Tesis de maestría, Maestría en Ingeniería de Telecomunicaciones, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.

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

Biblioteca Carlos Gaviria Díaz

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

Universidad de Antioquia - www.udea.edu.co

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

Demographic Information Retrieval From Text For Subject Characterization And Market Segmentation

UNIVERSIDAD DE ANTIOQUIA

1 8 0 3

Research work for the Master's degree in Telecommunications.

Daniel Escobar Grisales

Director: Prof. Dr.-Ing. Juan Rafael Orozco Arroyave
Advisor: MSc. Juan Camilo Vásquez Correa

Faculty of Engineering

Department of Electronics and Telecommunications Engineering

University of Antioquia

Acknowledgments

First of all, I would like to thank my mother Amparo Grisales, my father Oswaldo Escobar, and my sister Diana Escobar. They are my main motivation in the different stages of my life and their support was very important to complete this work. I want to express my gratitude to my aunt Cecilia Grisales, my cousin Alexander Rojas and his wife Catalina Montaño, who welcomed me into their home, giving me their love and trust. I will always be deeply grateful to them. I also want to thank all my family, my aunts, my uncles, my cousins, especially Andres Rodriguez and Valentina Rodriguez who always support me and encourage me to give the best of me.

I want to express also my gratitude to Cristian Rios, Felipe Orlando Lopez, and Luis Felipe Parra, who not only helped me in different technical aspects of this work, but also they accompanied me in the complicated and happy moments of this master's degree. More than colleagues, I consider them part of my family. I would like to thank the pattern recognition team of GITA Lab. at the University of Antioquia: Paula Andrea Perez, Tomas Arias, Luis Felipe Gomez, Guberney Muñeton, Surley Berrio, and all team members. Thanks to my director, Prof. Dr.-Ing Juan Rafael Orozco Arroyave, who motivated me to start this master, guided me, and helped me in every part of the process. All my gratitude to him, who more than a director I consider a friend. I would also like to thank my advisor MSc. Juan Camilo Vásquez Correa, without his help it would not have been possible to complete this work, and whom I admire not only for his academic success but also for his ability to transmit all his knowledge.

Finally, thanks to the master's scholarship fund of the University of Antioquia and the Pratech Group for financing this research through the grant # PI 2019-24110.

Abstract

In recent years, the most important trends to improve customer services in the e-commerce industry are focused on customer customization and the use of automated dialogue systems to enhance the support experience. On one hand, demographic traits from a subject/customer such as gender, nationality, and age can help to strengthen marketing strategies or even improve customer empathy with the product or the advisor. On the other hand, the service of automated dialogue can help to improve the ability to serve multiple users. However, in order to improve the customer service support, the dialogue system should correctly recognize the customer requirements. This research work aims to improve customer services based on text data from the subject/customer, considering both scenarios, demographic trait recognition, and evaluation of effectiveness in conversations between humans and chatbots.

For demographic trait recognition, this work propose the use of recurrent and convolutional neural networks and a transfer learning strategy to recognize three demographic traits: gender, variety language according to nationality, and age. Models are tested in two different document types, Tweets (documents written in informal language) and call-center conversations (documents written in formal language). In documents in informal language, accuracies of up to 75% and 92% are achieved for the recognition of gender and language variety, respectively, and an unweighted average recall of up to 50% is achieved for age recognition. In documents in formal language, accuracies of up to 70%, 72%, and 68% are achieved for the recognition of gender, variety language, and age respectively. Results indicate that for the traits of gender and variety language it is possible to transfer the knowledge from a system trained on a specific type of expression to another, where the structure is completely different, and its amount of data is scarcer. In addition, the learning acquired by the models to recognize language va-

rieties in Spanish-speaking countries can be successfully used to fine-tune models to recognize more subtle language varieties, such as the ones within the same country.

For evaluation of effectiveness in conversations with chatbots, we propose a new methodology for automatic evaluation of chatbot effectiveness in real production environments. The analysis considers convolutional neural networks, using two parallel convolutional layers to evaluate questions and answers independently. This methodology is tested upon real conversations of chatbots that provide service to two different companies. The results are compared to baseline models based on classical techniques with different pre-trained word embedding models. According to our results, the proposed approach provides accuracies between 78% and 80%, which outperforms the best result of the baseline models by 2.9%.

Contents

Chapter 1

Introduction

1.1 Motivation.

E-commerce has been growing in the last years and has accelerated its growth during the COVID-19 pandemic. Recent studies published by the marketing research firm *eMarketers* show that the region with the highest growing in e-commerce in 2020 was Latin America with an increase of 36.7% followed by North America where the increase was of 31.8% [1]. This increase makes e-commerce attractive to many companies, which increase competition among each other to boost their sales in the online market. According to [2], the most important trends in recent years to improve e-commerce services are those related to customer personalization and the use of automated dialog systems to enhance the shopping experience.

An advanced customization capability can increase revenue by 25 percent [3]. Customer personalization not only improves customer empathy with the product but also helps to strengthen marketing strategies [4]. For instance, it is possible to segment and to personalize offers, thus products and services are exposed to the group of greatest interest. However, to deliver personalized experiences or to intelligently segment the market, marketers need to better understand their customers or prospects by retrieving specific information at each interaction. Although most of this information is collected explicitly through the registration process, this approach may be limited given that most potential customers of online stores are anonymous [5].

On the other hand, automated dialog systems such as chatbots can help to improve the ability to serve multiple users and to support services 24/7 [6]. However, chatbots, especially in languages such as Spanish, do not meet cus-

tomer expectations because the chatbot fails to correctly recognize the customer's requirements [7], [8]. As a result, many service providers stop using their chatbots due to negative feedback received from their customers [9].

Natural Language Processing (NLP) integrated with Machine Learning (ML) and Deep Learning (DL) can help to overcome the limitations to estimate customer demographic variables [5] and to evaluate the effectiveness of chatbots. This research work aims to extract customer information to improve and customize marketing strategies and the customer service offered by e-commerce. The objective is to design and to evaluate different models with the aim to: (1) predict subject/customer Demographic Traits (DTs) to improve marketing strategies and (2) evaluate the quality of customer service chatbots in production environments to improve their ability to address customer requirements and thereby improve their satisfaction with the service.

1.2 State of the art.

1.2.1 Methods to predict demographic traits.

Demographic Information Retrieval (DIR) consists in recognizing traits from a human being such as age, gender, personality, emotions, nationality, and others. Typically, the main aim is to create a user profile based on unstructured data. In e-commerce scenarios, this type of information provides advantages to companies in competitive environments because it allows to segment customers in order to offer personalized products and services, which strengths their marketing strategies [10], [11]. Text data from customers can be obtained via transliterations of voice recordings, chats, surveys, and social media. These text resources can be processed to automatically recognize gender, age or nationality of the users. Different studies have applied NLP techniques for gender, nationality or age recognition based on text resources, mainly from social media posts.

PAN-CLEF is a scientific event that has gained popularity for its competitions in multilingual tasks related to author profiling. Depending on the version, different challenges have been proposed, including the identification of personality [12], Profiling Hate Speech Spreaders on Twitter [13], gender [12]–[16], language variety [13], age group [12], [14], [16], [17], author diarization [16], and others. These challenges have contributed to the development

of new strategies to recognize DTs based on text from social media.

Term Frequency-Inverse Document Frequency (TF-IDF) is a classical method to extract features from text data widely used to resolve different NLP-tasks, including profiling. This feature represent each document based on the frequency of occurrence of the words in the document, weighted by its occurrence in all the documents in the corpus. In [18], [19] the authors used TF-IDF to extract features from tweets in the PAN17 corpus [13], which has labels for gender and language variety from different nationalities, including Argentina, Colombia, Venezuela, and others. By using a Support Vector Machine (SVM), the authors reported accuracies for gender classification and language variety identification around 81% and 94%, respectively, for Spanish language. In [20], the author extracted features from TF-IDF as well as specific information only available in social media posts such as the frequency of female- and male-emojis. The authors reported an accuracy of 83.2% in the PAN17 [13] corpus for gender recognition and 89.8% for language variety identification. This type of features have also been considered for age classification in [14], [21], obtaining accuracies of 43% in the PAN14 corpus. Although the high accuracy reported in [20], this type of methodology would not be accurate to model text data written in more formal scenarios such as customer reviews, product surveys, opinion posts, and customer service chats, which have a different structure compared to the texts data available in social media. Moreover, these types of features highly depend on the corpus, reducing generalization to other domains. For instance, some studies [22], [23] have concluded that females use emoticons more often than males, while another study [24], concluded the opposite.

In addition to TF-IDF, other works incorporate features such as Latent Semantic Analysis (LSA) and Second Order Atributes (SOA) [25]–[28]. LSA aims to define a new semantic space using TF-IDF and singular value decomposition. SOA is a supervised frequency based approach to estimate document vectors, Under this representation, each value in the document vector represents the relationship of each document with each target profile. In [29], the authors compare these features for gender and age classification in the PAN15 corpus. The authors show that results with LSA outperformed to SOA in all languages in both tasks. The combination of these features via an early fusion strategy improves the results for age classification by about 1% for English and Spanish. For gender classification the results only improve for English around 4%.

There are other NLP techniques that, just like LSA, aim to obtain representations of text by modeling its semantic content. Among the most widely used for author profiling is Word2Vec. This technique uses a neural network to learn word representations that maintain the semantic information of the word. In studies such as [30], the authors proposed a system to classify the gender of the persons who wrote 100,000 posts from Weibo (Chinese social network similar to Tweeter) based on Word2Vec. The system achieved an accuracy of 62.9%. The proposed method was compared with human judgments, whose accuracy was 60%. This fact evidences that the problem of recognizing gender in written texts is very hard even for human readers.

Word2Vec [31] and TF-IDF have been considered for age classification in studies such as [32]. The authors used this type of features to classify different demographic variables, including 3 age-groups: 18-20, 23-25 and 28-61. Classification was performed by logistic regression considering k-fold cross-validation strategy. The approach obtained accuracies of 60%, 56% and 61%, for each age group, respectively. Other features based on psycholinguistic dictionaries and Linguistic Inquiry and Word Count (LIWC) were considered, but as in previous works, features based on TF-IDF and Word2Vec achieved the best results.

Recently, Deep Neural Networks (DNN) such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been widely explored for various NLP tasks due to their high performance without need for engineered features [28], [33], [34]. CNN architectures have shown to be efficient in the author profiling tasks such as personality and author identification [35], [36]. In [37], the author used a methodology based on word- and sentence level embeddings and CNN for gender and variety language identification. Word2Vec and FastText, which is a variation of Word2Vec at character-level were considered. The model was evaluated in the VarDial corpus, which is composed of news articles in different varieties of Spanish. The proposed methodology was compared with a ML approach based on traditional features and SVM classifiers. Accuracies up to 73% and 92% were achieved for the CNN approach for gender and variety language identification, respectively. Results indicated that CNN models outperformed traditional ML models. In [28], the authors used different approaches to aggressiveness, occupation and location in the MEX-A3T database [38]. Among the different methodologies, the authors proposed a CNN with three different ways to initialize the weights for the embedding layer: 1) CNN-rand: In the model all

words are randomly initialized and then modified during training. 2) CNN-static: This model uses word embbeding vectors from a pre-trained model using FastText [39] to initialize the embedding layer and these weights are not modified during training. 3) CNN-NonStatic: is the same as previous one, but the embedding weights can change during training. The three CNNs show similar results for the three task, 61%, 55%, and 54% for aggressiveness, occupation, and location, respectively. The author concluded that the low accuracy could be due the lack of data. In addition, CNNs for text analysis have been used in task related with customer services. For instance in [40], the authors used CNNs to identify the customer churn based on the transcriptions of conversations in a call center from a telecomunication company. The embbedings were based on a pre-trained Word2Vec model.

RNNs have also been used to identify the author's demographic variables. In [41], the authors proposed a methodology based on Bidirectional Gated Recurrent Units (GRUs) and an attention mechanism for gender classification in the PAN17 corpus. The authors worked with a Word2Vec model as input for their DL architecture and reported accuracies of up to 75.3% and 85.2% for gender and language variety identification, respectively.

According to the reviewed literature, DIR based on text data has been mainly explored by using traditional ML techniques in social media scenarios, where the language is informal and the documents do not follow a formal structure [42]. There is a gap between models trained on formal and informal written language because a trained model with formal language data for a specific purpose will not achieve comparable results on an informal language scenario, or vice-versa [43]. Due to this reason, it is important to develop and evaluate trained models to estimate demographic variables in both types of languages: formal and informal. In addition, DTs recognition has been under-explored in documents related to e-commerce or customer service because, compared to data from social media, it is difficult to collect a large amount of such labeled data.

1.2.2 Methods to evaluate the quality of chatbots

A chatbot is a conversational software system that automatically interacts with a user/customer. It is designed to emulate the communication capabilities of a human being [44]. The service-oriented chatbot acts as an automated customer service representative, giving natural language answers [45].

Despite their great potential, many customer service chatbots do not meet customer expectations because the chatbot fails to correctly recognize the customer's requirements [7]. As a result, many service providers stop using their chatbots due to negative feedback received from their customers [9]. For this reason it is necessary to evaluate chatbot's effectiveness, i.e. their ability to recognize costumer's questions or requirements and to give a clear response or enable the required service.
Contact centers typically conduct surveys with a randomly selected and small group of customers to measure chatbot's effectiveness [46]. The selected group of customers is rather limited due to the high cost involved in the manual evaluation process. The number of evaluated conversations are typically not representative of the real state of the customer service process. Another popular and less expensive way to determine the effectiveness of a chatbot is through self-reported user satisfaction. Most of the works use the Likert scale to evaluate different aspects related to the conversation, such as effectiveness, quality, humanity, manner, and others [47]–[52]. In [47] the authors proposed a framework for chatbot evaluation based on the Grice's maxims, which consists of four conversational maxims originated from the natural language: quality, quantity, relation, and manner. A similar methodology is presented in [53] where the users were asked to rate chatbot's performance for Grice's maxims on a Likert scale. The authors used the chatbot conversation to test the correlation of human judgments with the Grice's maxims. Other methodology widely used to evaluate chatbots' performance is the PARAdigm for DIalogue System Evaluation (PARADISE), which estimates subjective factors by collecting user ratings through questionnaires. Some of the subjective factors include ease of use, clarity, naturalness, friendliness, robustness and willingness to use the system again [54]. The main drawback of using self-reported data to evaluate customer service and effectiveness in conversations with chatbots is that, typically only few users complete the questionnaire and those few completed questionnaires are generally influenced by different external factors [55], [56].

Other methodologies used to assess effectiveness of chatbot's conversations include the BiLingual Evaluation Understudy (BLEU) [57] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [58]. These methods were originally proposed for machine translation systems and are based on similarity measures between generated texts and expected responses. There exists also the Metric for Evaluation of Translation with Explicit Order-

ing (METEOR) [59], which is similar to BLEU but incorporates synonym matching. Several works are based on the aforementioned methods [60]–[66]; however, they have different limitations. For instance, they require reference corpora and are focused on token-level overlap between the reference and generated texts, hence a valid response to a certain statement in a conversation might has low token overlap with a reference response [67]. Additionally, the token-level overlap has shown poor correlation with human judgments, which limits its use in real-world applications [68]. There are also evaluation methods based on statistics about the conversation between customers and chatbots. For instance the number of times the chatbot was used [69], the number of times the user had to use help commands [70], total number of dialogues and their duration [67], and frequency of a keyword related to the customer's feeling [46], [70].

Recent studies are mainly based on word embeddings and use the cosine distance to measure similarity between a given response and the reference [68], [71]–[73]. The embeddings are typically generated by using an embedding layer of a neural network or are based on pre-trained models such as GLobal Vector for word representation (GLoVe) [74], Word2Vec [31], Bidirectional Encoder Representations from Transformers (BERT) [75], and BETO which is the Spanish version of BERT [76]. These embedding-based approaches are less studied in the topic of evaluating effectiveness in communications between chatbots and customers but they are capturing the attention of the research community thanks to their flexibility and good performance in other fields like sentiment analysis [77], mental health evaluation [78], [79], and others.

According to the reviewed literature, automated evaluation of chatbots effectiveness has been poorly studied with ML algoritms. Most of the automatic approaches are based on self-reported satisfaction or comparison with reference corpora. Few studies have addressed the problem using statistical learning strategies and no papers were found in the area of DL techniques to evaluate chatbots' effectiveness. Moreover, in production environments, the products and services offered by a company are constantly changing, leading to the need for to re-training chabots. Therefore it is necessary to propose an approach that does not define the quality of the chatbot by the similarity of the responses to a reference corpus.

1.3 Research problem

Given the different literature description shown in Section 1.2, the studies for the DTs recognition have mainly focused on documents from social media, and using traditional learning models. In e-comerece and customer/agent interaction environments, where the language has a more formal structure, it has been little explored due to the small amount of available data [32], [80]–[82]. Moreover, there are few papers that consider automated analysis methods in Spanish language.

In the studies related to the effectiveness analysis of dialog systems, chatbots design has been extensively studied, however, their automatic evaluation has only been considered by comparison with a reference corpus or by self-reported satisfaction tests. There are few approaches where automatic learning models are considered. In addition, the comparison with respect to a reference corpus exhibits low correlation with human judgments, leading to these methodologies have problems in production environments because the constant retraining of chatbots [83]–[85].

According to the previously described problems, the following research questions have been defined:

- ✓ How to recognize demographic traits of the author such as: gender, age, and language variety based on transliterations from call center conversations?.
- ✓ How to train a DL model from informal language documents to recognize DTs from documents with formal language?.
- ✓ How to automatically evaluate the effectiveness of a chatbot in a production environment without depending on a reference corpus and for this assessment has a high correlation with human evaluation?.

1.4 Objectives

1.4.1 General objective

To design and implement a methodology based on NLP methods, DL and traditional learning techniques applied on Spanish texts from interactions between customers and call-center agents to model demographic information of customers and to do market segmentation.

1.4.2 Specific objectives

- ✓ To propose and implement methodologies for the feature extraction from texts, based on NLP methods, DL techniques and unsupervised learning algorithms
- ✓ To design and evaluate a system to DIR from transcriptions generated from interactions of customers with call center agents.
- ✓ To compare the results of this methodology by using in a standard database, in this case the database PAN17, which contains text of interactions of people with social networks (posts).
- ✓ To design and evaluate strategies for market segmentation based on information extracted from text.

1.5 Contribution of this study

This research work proposes different methodologies to retrieve information from text potentially useful to improve marketing and customer service strategies. This study considers two main scenarios: (1) the DTs recognition and (2) the evaluation of the effectiveness of chatbots in production environments.

For the DTs recognition, different corpora are studied, considering texts with informal structure (Tweeter data) and texts with formal structure (call-center conversations). Transfer Learning (TL) strategies are tested to improve results in call-center conversations using pre-trained social media models. Results indicate that the knowledge of the model for gender, language variety, and age prediction in documents with informal language allows to improve models to predict different DTs in documents with formal language. In addition, the knowledge learned by the model to classify an author by language variety (Argentina, Venezuela, Spain, Peru among others) helps to improve the results to classify authors from different Colombian regions, i.e, “Antioqueños” and “Bogotanos”.

A new approach based on parallel CNNs is proposed to evaluate effectiveness of chatbots in production enviroments. The proposed model also incorporates filters to extract features with multiple temporal resolutions. This methodology is tested upon real conversations of chatbots that provide

service to two different companies. The results are compared with baseline models based on classical techniques with different pre-trained word embedding models. According to our results, the parallel CNNs approach outperforms the best result of the baseline model by 2.9%. Based on this methodology a software prototype in Technology Readiness Level 6 is developed for the evaluation of conversations between chatbots and customers. To the best of our knowledge, this is the first study about automatic analysis models for chatbot effectiveness in production using DL techniques.

Chapter 2

Theorical background

This chapter consists of five sections: (1) Natural language processing, where the algorithms, models, and tools used to obtain a mathematical representation of data unstructured such as documents or texts are explained, (2) Machine learning and deep learning methods, which summarizes different algorithms used to build statistical models that allow to analyze and draw inferences from patterns in data, (3) Transfer learning, that introduces some strategies to exploit the knowledge acquired of a model pre-trained to improve the generalization capabilities in another model where typically the amount of data is most limited, (4) Validation strategies, to explain the validation strategies considered in this work, and (5) Performance metrics, where the metrics used to evaluate the models are explained.

2.1 Natural language processing methods

Natural Language Processing (NLP) is a set of tools, techniques and algorithms that allow a computer to process and understand unstructured data such as documents or texts. The main objective of NLP is to create a representation of text that adds structure to the unstructured natural language. This structure allows to analyze naturally written text from a syntactic, semantic or morphological point of view [86]. NLP analysis allows to perform multiple tasks such as: generating summaries of texts, identifying writing styles [87], intent classification [88], sentiment analysis [89], machine translation [90], information retrieval [91] and others.

In the beginning, NLP methods were designed like a system based on

rules, where a set of hand-coded methods were designed to give structure or to extract information from a text document. However, these systems can only be improved by increasing the complexity of the rules, which involves extensive handwork and expert knowledge for the design of each rule, making it system more and more difficult to scale. Other methods have emerged as a solution to these problems. These methods were based on statistical inference or ML to automatically learn the rules through the analysis of large corpora of typical samples of the problem to be modeled. Although these methods learn the rules automatically, they also need expert knowledge to generate a feature matrix, which is used as input of the ML algorithms. In the feature matrix each row represents a sample of the phenomenon and each column represents a feature. Finding the best features for a particular NLP problem requires specific prior knowledge of the problem, which is known as feature engineering.
The most recent NLP techniques include strategies based on neural networks or DL architectures. The main advantage of these new methods over the previously exiting ones, is that it does not required a stage of feature engineering. This fact allows that more complex NLP tasks to be developed without the need for expert knowledge on each problem. However, the major drawback of these methods is the number of samples required to train these architectures correctly. NLP techniques have constantly evolved and, despite the fact many methods based on ML have been replaced by neural networks, they continue to be relevant for contexts where interpretability and transparency are required or where there is insufficient data to train complex models.
In NLP methods, depending on the task and the type of samples, a preprocessing stage may be required to filter out noise generated by punctuation marks, capital letters or other text elements that do not provide useful information.

2.1.1 Pre-processing

This step consists of multiple sub-steps, such as: stemming, lemmatization, lower casing, and removal of punctuations, stopwords, frequent words, rare words, emojis, URLs, and others. However, depending on the problem only some sub-steps or none of them are used. For example, in NLP tasks like sentiment analysis, the meaning of the sentence or the frequency of the positive words might be important, thus it is useful to use stemming, because this

sub-step reduces inflected or derived words to their stem [92], which reduces noise, reduces vocabulary size and helps to detect words that are spelled in different ways but have the same meaning. A similar situation occurs when using lemmatization and lower casing. In other tasks, such as author profiling, the use of stemming could have a negative effect because the way like the author writes a specific word can be a pattern of the author's style and this information would be lost when changing the word using stemming [19]. Another factor to take into account to define the pre-processing, especially in NLP tasks focused on the retrieval of author DTs, is the source that generated the document. In a document generated by an Automatic Speech Recognition (ASR) system or by manual transliterations of spoken documents, the subject and who transcribes the text are different. In these type of problems, the NLP-system for author profiling aims to estimate the demographic variables of the subject who is speaking, based on the texts generated by a human or by a system. In this case, the sub-steps of lower casing, strip accents, and punctuation marks removal, can help to obtain more robust representations of the text, removing the writing style of the transcriber [93], [94]. Conversely, when the text generator is the original author, such as in social media posts, these sub-steps could cause that useful author-related information to be lost. The preprocessing stage in an NLP system is important because it reduces noise in the unstructured data, which facilitates the generation of more robust and simpler models [95]. However, pre-processing stages will depend on the phenomenon to be modeled and its relationship with the original source texts. Also this stage impact classification accuracy depending on the domain and language of the texts [96].
In this work, the URLs, mentions, hashtags, links, numbers, special characters, for example "rt", and HTML code, were removed from Twitter-based data, the emojis were not removed because in some works such as [19], [97] have shown to contain useful information to discriminate DTs such as gender. For transliterations from conversations in a customer service center a different pre-processing has been performed, the sub-steps of this pre-processing are as follows:

- ✓ Remove URLs, links and buttons.
- ✓ Remove punctuation.
- ✓ Remove emojis.

- ✓ Remove numbers.
- ✓ Lower casing
- ✓ Remove words with only one letter different to vowels or "y"
- ✓ strip accents

The pre-processing is more restrictive to data from customer service because these texts were manually transcribed based on conversations between customers and advisors, the text generator and the original author are different in this data. This is not the case for data from social media, where the documents were written by the author, therefore the writing style have useful information about the subject. For conversations between chatbots and customers, the same pre-processing stage is used, adding other sub-steps to remove content special of conversations via chat, such as special buttons and multimedia content.

After the pre-processing stage, documents are mapped into vectors that contain numerical values thus the computer can processes them. This vector representation can be built based on specific features of the document by using techniques like Term Frecuency-Inverse Document Frecuency (TF-IDF), Latent Semantic Analysis (LSA), and others. Nevertheless, recently the vector representation of a document depends of the vector representation of the words that compose it. This is because the progress of neural networks and advances in DL architectures have allowed to obtain vector representations of words that maintain some of their semantic properties [98]. These representations are known as word-embeddings and will be presented in Section 2.1.3.

2.1.2 Tokenization and word representations

Let's consider a set of documents $D = [d_1, d_2, d_3, ..., d_L]$, where each document could be a sentence, paragraph, phrase and others, in NLP this set is known as body. Tokenization refers to split a document into smaller units, such as individual words, character or terms. These units are called tokens. Generally, the small unit is the word but there are works in information retrieval based on text, where each token is a character [20]. All unique-tokens in the body form the vocabulary. However, generally only the V most

frequent words in the body are considered in the vocabulary to reduce the complexity of the methods and the noise caused by misspelled words.

To generate a vector representation of each token/word the simplest method is *One-Hot* encoding. A *One-Hot* vector represents each word in the vocabulary as a vector with only one element in 1 and the rest are 0. The position of the hot element (1) depends on the position of the word in the vocabulary. The dimension of the resulting vector depends on the number of words within the vocabulary (V).

Although this word representation is simple and easy to implement, it has several problems. First, it is not possible to infer any semantic relationship between two words given their *one-hot* vector. And second, the resulting vectors have a high dimensionality because they depend on the size of the vocabulary. The solution for these problems is the use of Word Embeddings, which is a type of mapping that allows words with a similar meaning to have a similar representation, with a smaller dimension than *One-Hot* vectors.

2.1.3 Word-embeddings

Word embeddings are real-valued representations of words produced by distributional semantic models [99]. Given a set of words $W = [w_1, w_2, w_3, ..., w_V]$, where V is the vocabulary size, a word-embedding is a point S_i in a numerical space S, where each point corresponds to a word in the vocabulary and the similarity between each pair of points is well defined [100]. When one word has a unique representation, the word-embedding is known as context independent because the representation is the same independently of the context of the word. When one word has multiple representations is because the representation depends on the context of the word i.e., words around the word of interest, and this word-embedding is known as context depend.

There are different methods and architectures to generate word embeddings and these can be either context-independent or context-dependent.

Context independent

One of the well known context-independent word embedding models is Word2Vec [101]. This model takes a large text corpus as input and produces a vector space, typically of several hundred dimensions. Word vectors are positioned in the vector space such that words sharing common context

in the corpus are geometrically close to each other [31]. There is a unique vector to represent each word in the corpus. For this reason it is known as a context-independent embedding because the representation of a word is the same regardless its context. This algorithm uses a neural network model to learn word relations. The original words are transformed to an *one-hot* representation to feed the network. Word embeddings based on Word2Vec can be obtained by following two strategies: Continuous Bag Of Words (CBOW) or Skip-Gram. CBOW takes the context of each word (one-hot encoded) as input and the network aims to predict the word based on the context. The number of context words is previously defined, typically a number between 3 and 7 is a good choice [102]. The neural network structure is shown in Figure 2.1, where $\mathbf{X}_k$ corresponds to c context words (*one-hot* encoded) of the k-th word in the vocabulary, v is the size of the vocabulary. The hidden layer contains d neurons and the output is a v-dimensional vector that corresponds to the target word. Skip-Gram is a similar strategy but in this case a word (*one-hot* encoded) is taken as input and the network aims to predict the context corresponding to the word. Figure 2.2 shows the neural network structure for the Skip-gram strategy.

These types of representations have gained popularity not only because each word-embedding keeps the semantic properties of the word, but also because it is a model trained in an unsupervised manner, i.e., the texts that are analyzed do not require prior labeling. This makes it possible to train models with a large amount of data based on freely accessible text resources without spending money on expensive hand-labeled databases.

In this work, two Word2Vec models were trained with dimensions of 100 and 300, namely 100-W2V and 300-W2V, respectively. Both models are trained with the Spanish WikiCorpus, which contains 120 million words [103]. In 100-W2V model CBOW strategy and 7 context words were used, while in 300-W2V model Skip-Gram strategy and 8 context words were considered. In this research work were considered both types of architectures because in some works show that CBOW have a performance slightly better than Skip-gram in learning words [104], [105]. Other works show that Skip-gram requires less training samples and works better for non-frequent words [106].

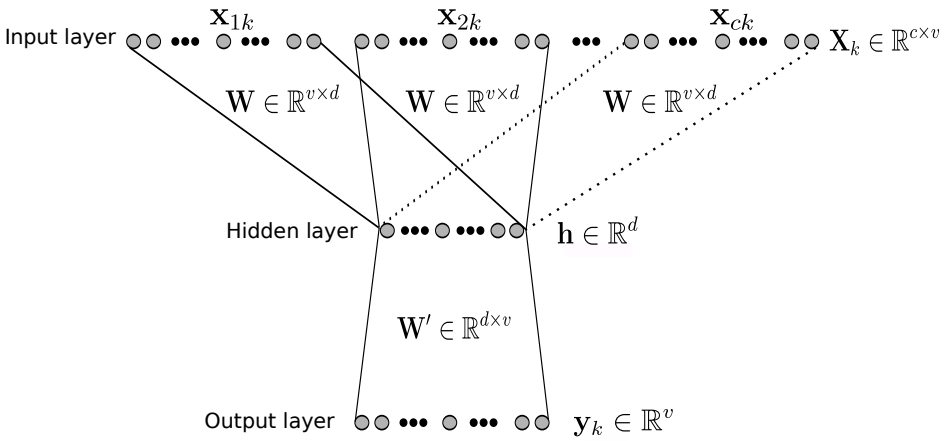


Figure 2.1. CBOW model. $\mathbf{X}_k$: context words (*one-hot* encoded) of the k-th word in the vocabulary; $\mathbf{y}_k$ *one-hot* encoding of the k-th word in the vocabulary; c: number of context words; v: number of words in the vocabulary; $\mathbf{W}$: weight matrix before the hidden layer, $\mathbf{W}'$: weight matrix after the hidden layer; and d: Word2Vec embedding dimension. Figure adapted from [107].

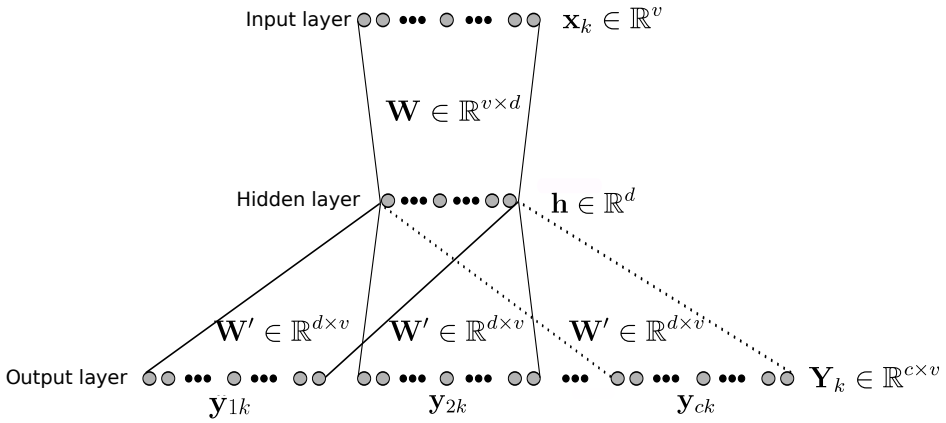


Figure 2.2. Skip-gram model. $\mathbf{Y}_k$: context words (*one-hot* encoded) of the k-th word in the vocabulary; $\mathbf{x}_k$ *one-hot* encoding of the k-th word in the vocabulary; c: number of context words; v: number of unique words in the vocabulary; $\mathbf{W}$: weight matrix before the hidden layer, $\mathbf{W}'$: weight matrix after the hidden layer; and d: Word2Vec embedding dimension. Figure adapted from [107].

Context dependent

In this type of embeddings the representation for each word is not unique because a representation depends on the context of the word. Recent developments of context-dependent embeddings [75], [108] show that systems based on such representations achieve good results in many different NLP tasks [109].
Bidirectional Encoder Representations from Transformers (BERT) [75] is one the most popular context-dependent embeddings. It is based on a Transformer architecture [110] originally, created for machine translation. This Transformer includes two separate mechanisms: an encoder that reads the text inputs and a decoder that produces a prediction for the task. The encoder is formed with a stack of layers that include self-attention and feed-forward connections. Decoders include all the elements present in the encoder with an additional encoder-decoder attention layer between the self-attention and the feed-forward layers [110]. Figure 2.3 shows the encoder and decoder of the Transformer architecture. To compute BERT embeddings only the transformer encoder is used, where the most important part is the multi-head attention mechanism. This mechanism consists of several attention layers running in parallel, which learns contextual relations among words (or sub-words) in a text. The attention function can be defined as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{2.1}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices, respectively. These matrices are built from the word-embeddings of sequences extracted from text. In BERT $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the same sequence, which is known as self-attention. The dot product between $\mathbf{Q}$ and $\mathbf{K}^T$ has information about the relationships among elements of $\mathbf{Q}$ and elements of $\mathbf{K}$. Then, this matrix is scaled by the $\sqrt{d_k}$, where d_k is the dimension of the word-embedding and a softmax function is applied to obtain weights, known as attention score. Finally, the output is computed as a weights sum of the values $\mathbf{V}$, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

In Word2Vec model, the author used one unsupervised task, to predict the word based on its context words or predict the context words based on the word, CBOW or SkipGram, respectively. Word2Vec has a directional approach which inherently limits context learning. To overcome this chal-

lenge and train the architecture with a specific task, BERT uses two training strategies: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

In the MLM task before feeding word sequences into BERT 15% of the words in each sequence are replaced with a [MASK] token. The model aims to predict the original value of the masked words, using the words non-masked in the sequence. In the NSP task, the model receives pairs of sentences as input A and B, where, 50% of the time B is the sentence that follows A and 50% of the time it is a random sentence from the corpus. The goal of the model is to recognize when B is the subsequent sentence of A in the original document.

Two context dependent models based on BERT are considered in this work. The first one is the BERT-Base, Multilingual Uncased pre-trained model, which was trained with the Multi-Genre Natural Language Inference (MultiNLI) corpus. The second model is BERT-Base trained with Spanish data from Wikipedia and all of the resources of the OPUS project [111]. This corpus has about 3 billion words and it is available online[1]. This model is commonly known as BETO [76]. The architecture of the BERT-Base model consists of 12 self-attention layers each one with 768 hidden units, for a total of 110M parameters. The last layer (768 units) is taken as the word-embedding representation. The source code to compute BERT and BETO embeddings is also available online[2] [112].

Embedding layers

In DL architectures for NLP, an embedding layer is usually used to compute word-embeddings from one-hot vectors. Word-embeddings can be created by using a pre-trained models such as Word2Vec, however, it is also possible to use an embedding layer to learn word-embeddings with the most relevant information for a particular problem. This strategy is supervised, because the word-embeddings are computed based on the labels of each sample for the particular problem, in contrast to Word2Vec where each word is computed based on the context words (unsupervised). Similar to Word2Vec these word-embeddings are context independent, i.e., each word has a unique word-embedding regardless of the surrounding words. These word embed-

[1] https://github.com/josecannete/spanish-corpora
[2] https://github.com/PauPerezT/WEBERT

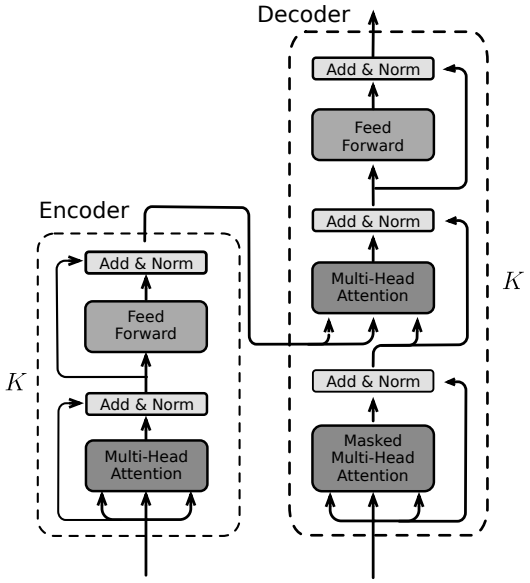


Figure 2.3. Topology of the Transformer architecture. K is the number of layers in the encoder and decoder. Figure is adapted from [110].

dings learn specific features that are relevant for a particular NLP task, such as, sentimental analysis or information retrieval. The drawback is that the number of parameters increases and the architecture may require more data to be properly trained.

These word-embeddings can be computed from scratch, where the embedding layer is initialized with random weights and the word-embeddings are estimated in the model itself. Another approach is to use a pre-trained model such as Word2Vec to initialize the weights and adjust the word-embeddings via re-training the embedding layer (Non-Static approach). The embedding layer can also be frozen to keep the weights from a pre-trained model, thus reducing the number of trainable parameters of the architecture (Static approach). There are works that show that learning word-embeddings based on the problem data can be useful to improve the accuracy of the models in different NLP tasks [113].

2.1.4 From word-embeddings to document representations

To obtain a vector representation for each document into the body, statistical functionals such as, mean, standar deviation, maximum, minimum, skewness and kurtosis are computed upon the embeddings of each word within the

document. Generally these funtionals are considered in each dimension of the embedding, which means that the vector dimension of each document will be given by $f \times d$, where f is the number of funtionals and d is the dimension of the word-embeddings.

Another numerical representation for documents based on their words is a matrix representation. In this type of representation it is necessary to limit the maximum number of the word in each document and do a zero-padding when is necessary. Typically, this representation is used in DL architectures such as CNNs. In classical patter analysis algorithms such as SVMs the document representation using statistical functionals is widely used.

In this work, for experiments of chatbot effectiveness evaluation, classical ML algorithms are used to generate the baseline models. In these experiments the text data are represented using four models: two context-independent (100-W2V and 300-W2V) and two context-dependent models (BERT and BETO). 6 statistical functionals: mean, standard deviation, maximum, minimum, skewness, and kurtosis are computed upon the word-embedding of each document to generate one vector representation per document.

2.2 Machine learning and deep learning methods

ML algorithms aim to build a model based on samples of the phenomenon, known as "trainig data", in order to make predictions or estimations on new data of the same phenomenon, known as "test data". Each sample is represented as a vector, where each dimension is a feature. The set of samples form the feature matrix, where each row is a sample. This matrix is the input for different algorithms, such as: Decision Trees (DT), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), SVMs, and others. In NLP tasks related to information retrieval, mainly those involving the prediction of demographic variables for author profiling, the SVM is the most used method, due to its generalization capability and its performance in problems with high dimensionality in comparison with other methods [13], [97], [98], [114].

2.2.1 Support vector machines

The main aim of an SVM is to maximize the separation between classes by finding a hyperplane with the largest distance to the nearest training data

point of any class by the concept of support vectors [115].

The decision function of the SVM is expressed according to

$$y_n \cdot (\mathbf{w}^\intercal \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n, \quad n = 1, 2, 3, \cdots, N \tag{2.2}$$

where the weight vector $\mathbf{w}$ and the bias value b define the separating hyperplane, ξ_n is a slack variable that penalizes the amount of errors allowed in the optimization process and N is the number of observations in the training data. $y_n \in \{-1, +1\}$ are the class labels and $\phi(\mathbf{x}_n)$ is a kernel function which maps the feature space $\mathbf{x}_n$ to a higher dimensional space, where the classes are linearly separable.

The optimization problem for finding the separating hyperplane is defined by Equation 2.3, where the hyperparameter C controls the offset between ξ_n and the margin width. The samples $\mathbf{x}_n$ that satisfy the condition of equality in the Equation 2.2 are the support vectors. Figure 2.4 shows a two-class SVM, where the maximum margin of separation is defined by the location of the support vectors.

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \tfrac{1}{2} ||\mathbf{w}||^2 + C \sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & y \cdot (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned} \tag{2.3}$$

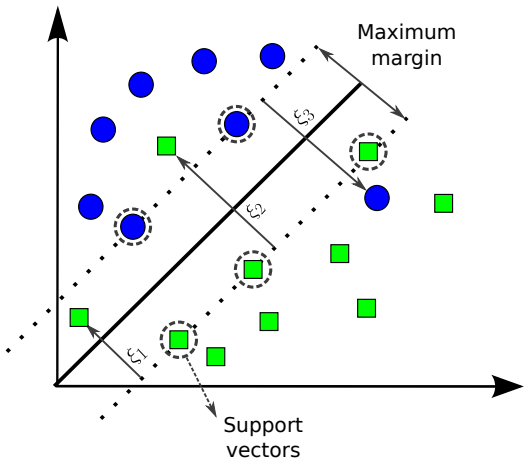


Figure 2.4. SVM with two classes. The maximum margin of separation is defined by the location of the support vectors.

The performance of classical ML methods such as SVM, is highly dependent on the feature matrix. Therefore, successful models based on classical ML algorithms require expert knowledge of the problem, looking for the samples to be represented by the features that best discriminate the classes according to the problem. The DL architectures allow to take a raw signal as input and the features are extracted automatically based on information of the data. The multiple layer of a deep representation learn the features that best modeled the phenomenon.

In NLP, DL architectures have been used to model different problems, such as, personality analysis, automatic translation, sentiment analysis, and others. Nevertheless, in the estimation of demographic variables or author profiling, the methods have been poorly used. The DL architectures widely used for NLP are the CNNs and RNNs.

2.2.2 Convolutional neural networks

CNNs integrate feature extraction and feature selection stages together with the pattern classification algorithm in a single architecture [40]. These networks contain a structure formed with convolutional filters and grouping layers, instead of the fully connected layers typically used in classical deep neural networks [116]. CNNs have been widely used in computer vision [117] and since recent years their applications have been extended to other domains/applications of NLP including machine translation [45], sentence/document classification [118], [119], generic text representations [120], [121], text-based sentiment analysis [122]–[124], and others. There are also works related to market analysis such as prediction of customer withdrawal based on transcriptions from call center conversations [40], [125].

CNN architectures for NLP are usually considered to extract sentence representations. Commonly, architectures include convolutional layers and max-pooling operations over all resulting feature maps. A sentence with l words can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{l \times d}$, where $\mathbf{X}$ is a word-embedding matrix composed of l word-embeddings with dimension d. In the convolutional layer the matrix is convoluted with some filter $W \in \mathbb{R}^{n \times d}$, where each filter has different size n but all filters have the same dimension d, which corresponds to the word-embedding dimension. The main idea of the CNN for text classification is to extract semantic features with multiple temporal resolutions through the convolution operation. Different filter sizes

correspond to different number of n in n-grams. Filters of $2 \times d$, $3 \times d$, $4 \times d$ are designed to map different semantic relationships including *bi*-gram, *tri*-gram, and *four*-gram, respectively. During the convolution process, each filter gradually moves down a word at a time along the sequence of words (i.e., vertically). Finally the max-pooling operation is applied after the convolutions. The final step is performed by a fully connected layer. The classification result of a sample is obtained by using a Softmax function applied to the output layer. Details of the architecture can be found in Figure 2.5 for a bi-class problem.

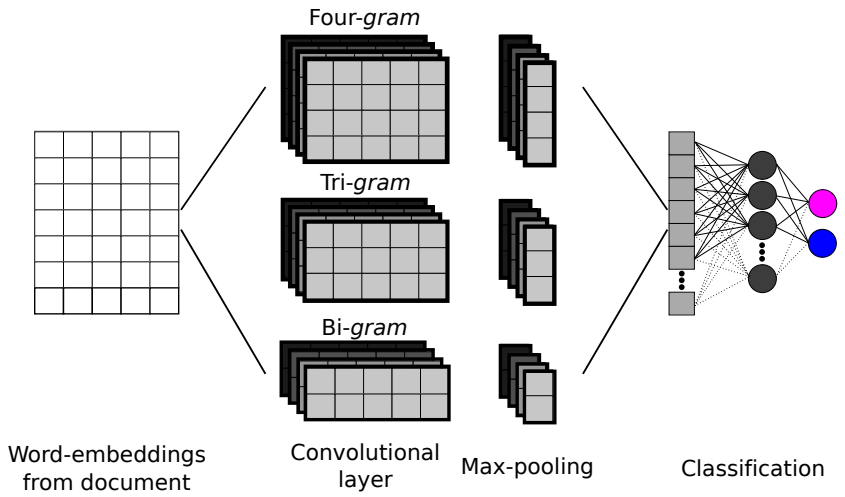


Figure 2.5. CNN architecture for NLP in Bi-class problems.

The methodology shown in Figure 2.5 is typically used when each sample of the phenomenon corresponds to a text or document from a unique source, for instance, texts from a book or an author. In this work, we proposed a novel architecture for problems, where each sample is composed of documents of two different sources, for instance, questions and answers in a conversation. This approach can be used in different problems, such as: sentiment analysis in conversations from WhatsApp, customer satisfaction in customer service, pedophilia analysis in social media conversations, and problems related to human-computer interaction systems like chatbot effectiveness analysis. The architecture is based on Parallel CNNs (P-CNNs) as shown in Figure 2.6. The input corresponds to word embeddings of questions and answers of each conversation. Then, two parallel convolutional layers are used in order to extract the corresponding feature vectors for questions and

answers, separately. Each convolutional layer has three parallel filters with different orders to exploit *bi*-gram, *tri*-gram, and *four*-gram relationships among words, and simultaneously allowing semantic features to be extracted with multiple temporal resolutions. Output vectors of this process have the same dimension because both convolutional layers have the same number of filters. These two vectors are concatenated before the fully connected layer to obtain a complete representation of the conversation. l_q and l_a in Figure 2.6 correspond to the maximum number of words in the questions and answers, respectively. The other dimension of the matrices corresponds to the size of the embedding that generally is generated using a embedding layer, which was previously explained. n_f is the number of filters in the Convolutional layer and n_d is the number of dense units in the dense layer. For this architecture the result of a sample is obtained by using a Sigmoid function applied to the output layer.
In this work, the embedding dimension was set to 100 to make it consistent with other studies with simple pre-trained models of Word2Vec, and with our baseline models. The best configuration of the parameters n_f and n_d is experimentally chosen based on performance evaluation considering the smallest possible number of parameters. More details about the optimization process are shown in Section 2.2.4

2.2.3 Recurrent neural networks

The main idea of RNNs is to model a sequence of feature vectors based on the assumption that the output depends on the input features at the present time-step and on the output at the previous time-step. Conventional RNNs exhibit a vanishing gradient problem, which appears when modeling long temporal sequences. Long Short Term Memory (LSTM) layers were proposed to solve this vanishing gradient problem by the inclusion of a *long-term* memory to produce paths where the gradient can flow for long duration sequences such as sentences of a Tweet, or the ones that appear in a conversation with a call-center agent [126].
The key in the LSTM is the cell state c_t, which gives the model longer memory of past events. The cell state can carry relevant information along the sequence. Thus, information from earlier time steps can arrive at later time steps, reducing the effects of short-term memory. LSTM can remove or add

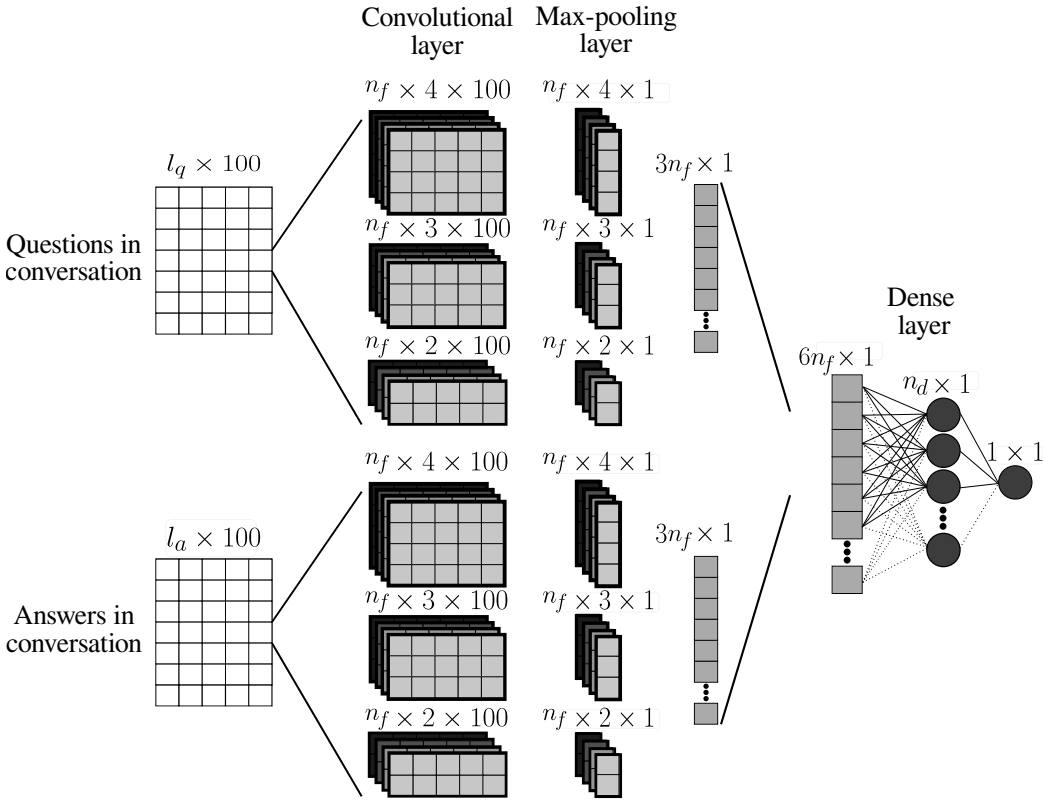


Figure 2.6. Proposed architecture (P-CNN). l_q and l_a are the number words in questions and answers, respectively, n_f is the number of filters in the Convolutional layer and n_d is the number of dense units in the dense layer.

information to the cell state, the information in the cell state is regulated by structures called gates. To decide what information is forgotten from the state cell, the forget gate f_t is used. Equation 2.4 defines the f_t, where information from the previous hidden state h_{t-1} and information from the current input x_t is passed through the Sigmoid function, the $\mathbf{W}_f$ is the weights matrix and b_f is the bias in the forget gate. The output of this gate is a value between 0 and 1, where values closer to 0 means to forget and values closer to 1 means to keep.

$$f_t = \sigma(\mathbf{W}_f[h_{t-1}, x_t] + b_f). \tag{2.4}$$

New information is add to the cell state using the input gate i_t and the vector of new candidates $\tilde{c}$. Similar to the forget gate, the input gate decides which values is updated, as shown in the Equation 2.5. The new candidates $\tilde{c}$ are computed based on Equation 2.6, The $tanh$ layer is used so that the output is between -1 and 1, which helps to regulate the network [3]. The old cell state is updated using Equation 2.7.

$$i_t = \sigma(\mathbf{W}_i[h_{t-1}, x_t] + b_i). \tag{2.5}$$

$$\tilde{c} = tanh(\mathbf{W}_c[h_{t-1}, x_t] + b_c). \tag{2.6}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}, \tag{2.7}$$

Finally, the output gate (Equation 2.8) decides what parts of the cell state is used to define the new hidden state. New hidden state is defined by Equation 2.9, where, $*$ denotes a point-wise (Hadamard) multiplication operator, as in Equation 2.7. All the gates, cells and activation have the same dimension.

$$o_t = \sigma(\mathbf{W}_o[h_{t-1}, x_t] + b_o). \tag{2.8}$$

$$h_t = o_t * tanh(c_t), \tag{2.9}$$

[3] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

LSTM has a *causal* structure, i.e., the output at the present time step only contains information from the past. However, many applications require information from the future [127]. Bidirectional LSTMs are created to address such a requirement by combing a layer that processes the input sequence forward through time with an additional layer that moves backwards the input sequence. A scheme of the Bidirectional LSTM is shown in Figure 2.7. Words from the data are represented using a word-embedding layer. The input to the Bi-LSTM layer consists of l d-dimensional words-embedding vectors, where l is the length of the sequence. The final decision is made at the output layer by using Softmax activation function.

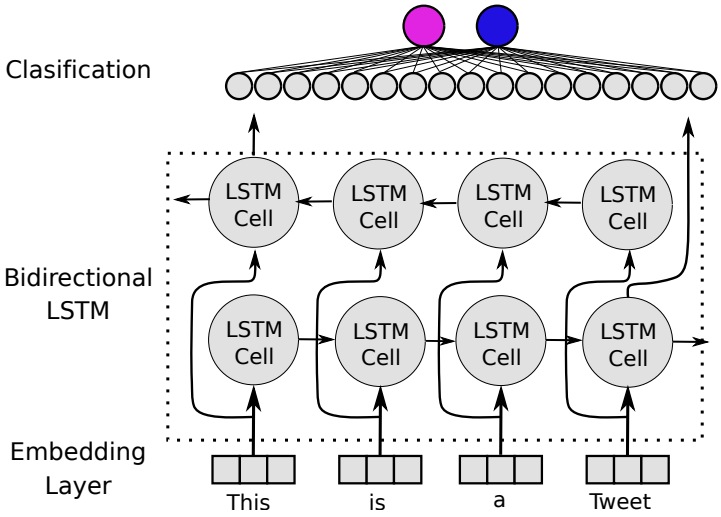


Figure 2.7. Bi-LSTM architecture for NLP.

2.2.4 Training

The architectures considered in this work are implemented in Tensorflow 2.0, and are trained with a sparse categorical cross-entropy loss function when the activation function of the last layer is Softmax and Binary cross-entropy when is Sigmoid, in both cases using an Adam optimizer [128]. An early stopping strategy is used to stop training when validation loss does not improve after 10 epochs and the model with the best validation loss is saved. The embedding dimension d is set to 100 for all experiments with DL and the embedding layer is trained from scratch. The vocabulary is defined with the v most frequent words in the body, a word is included in the

vocabulary only if its occurrence frequency in the documents from the body is greater than 5% of the total number of documents that form the body, otherwise, the word is not considered in the analysis because its frequency of occurrence is too low. This helps to reduce the number of embedding layer parameters. v is different for each experiment because it depends on the number of unique words present in the training sets of each experiment. Hyper-parameters such as: n_f and n_d in the architectures CNN-based and the number of cells in the architectures LSTM-based are optimized using a grid-search with values $\in \{16, 32, 64, 128, 256\}$, the dropout rate is varied $\in \{0.1, 0.3, 0.5, 0.8\}$ and the learning rate $\in 1e^{-4}, 1e^{-3}$. Hyper-parameters are chosen upon the validation accuracy and the simplest model.

2.3 Transfer learning

Classical ML and DL algorithms have been designed to work in isolation. These algorithms are trained to solve a specific problem. When the feature-space, the problem, or the domain change the model is rebuilt from scratch. TL aims overcoming the isolated learning utilizing knowledge acquired for one task to solve other related task. In [116], TL is defined as *"Situation where what has been learned in one setting is exploited to improve generalization in another setting"*.

Although it is not an exclusive concept for DL, TL is widely used for build new models based on complex DL architectures. This is because most models which solve complex problems with a high accuracy need a high amount of data, and obtaining these amounts of labeled data for supervised models can be difficult. Therefore, target models can be trained with less data using the gained knowledge (features or weights) from a scope model, improving the capability of generalization of the target model. The core idea of TL is shown in Figure 2.8. In [129], for the context of DL, where feature extraction and classification steps are integrated together, two popular strategies are defined, which are explained as follows.

2.3.1 Pre-trained models as feature extractors

DL models are layered architectures that learn different features in each layer. In the final of the architecture a fully connected layer is typically used to obtain the final prediction in a supervised manner. This layered

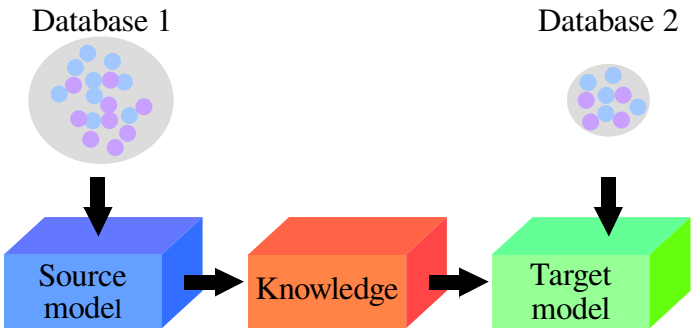


Figure 2.8. General methodology of TL.

architecture allows to remove the output layer (final layer) of a pre-trained model, in order to use the model as fixed feature extractor for other problem. Also, it is even possible to remove other layers in order to obtain the features of an intermediate layer that provides more useful information for the new problem. The aim is to take advantage of the pre-trained model weights to extract features, but not to update the model layer weights during training with new data for the new problem. Example of this strategy in NLP is the use of pre-trained word-embedding models such as Word2vec and BERT. In these cases, the word-embeddings are computed based on DL models trained with a high amount of resources (hardware) and data. The knowledge of these pre-trained model is used to compute word-embeddings that preserve semantic information that can be useful in different NLP tasks.

2.3.2 Fine tuning pre-trained models

In this technique not only some layers are removed, but also some parameters are adjusted with the new data. In DL models, the initial layers extract general characteristics and gradually the deeper layers learn more specific characteristics of the problem. Some layers can be frozen (fix weights), which allows to reduce the amount of parameter to fine-tune in the target model. The idea is to take the general knowledge of the problem and use this as a starting point to obtain better performance with less training time and less amount of data.

For example in DL architecture for NLP, such as the CNN or Bi-LSTM shown above, the embedding layer is the one with more parameters to learn because they depend on the size of the vocabulary v and the dimension of the embedding d. If the vocabulary of two problems is similar and there is

a source model trained for a specific task, then a pre-trained model can be used to generate a new model for a new task related with the task of the pre-trained model, where less data are available. The embedding layer can be frozen to reduce the number of trainable parameters and the output layer can be replaced by a layer according to the new problem.

2.4 Validation strategies

Validation strategies are used to optimize and evaluate the performance of the ML or DL model, which can be considered as close to the true performance of the model in real environments. The idea of the validation is to divide the data, in a training set and test set, where the training set is used to estimate all parameters of the model and the test set is used to measure the model performance. The most important in the validation is not to use the test data to optimize the model. Typically, the validation strategy depends on the amount of available data. There are two main validation strategies, Hold-out and Cross-validation.

2.4.1 Hold-out

This strategy is widely used when a large amount of data are available and the models have high computational cost. The aim of this strategy is to use three different sets (1) the training set to be used to learn the parameters of the model, (2) the development set, which is used to optimize the parameters of the system, and (3) the test set, which is used for the final evaluation of the performance of the model. Generally, the data are divided in training and test set using a 70-30% or 80-20% split, respectively. Then, the training set is split in training and development set using other partition (again 70-30% or 80-20%). The data for each set are chosen randomly. Figure 2.9 shows the hold-out strategy. This validation strategy can be done several times taking different training and test sets, similar to a validation strategy known as *Random Subsampling*. However in the DL models usually the training and test set are chosen once, because the computational cost is high depending of the amount of data and the complexity of the model.

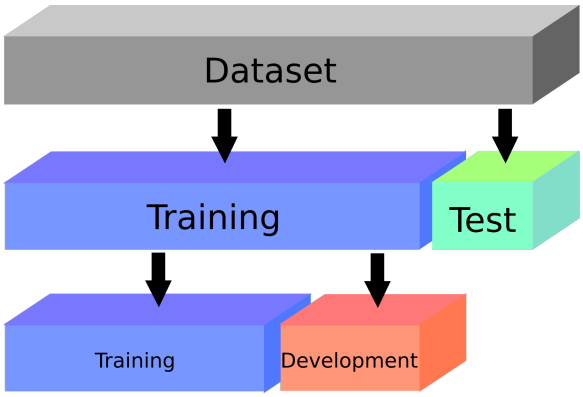


Figure 2.9. Validation strategy Hold-out

2.4.2 Cross-validation (k-folds)

This technique is widely used when there is not enough available data. In this strategy the data are divided into k subsets or folds, where k is an integer number. $k-1$ folds are used as training set and the remaining fold is used as test set. This process is repeated k times, with each fold available. Finally the performance of the model is computed using the average of the performance metric of each estimation. When $k = P$, where P is the number of samples in the complete database, this cross validation is known as Leave-One-Out Cross-Validation (LOOCV).

A common practice is to randomize the database and then split it into k folds, repeating the k-fold cross validation several times in order to consider folds with different samples. Figure 2.10 shows the k-fold cross validation with M iterations and $k = 5$. The advantage of this validation strategy is that all data are used for training and testing. The disadvantage is that this strategy has a higher computational cost compared to the Hold-out strategy.

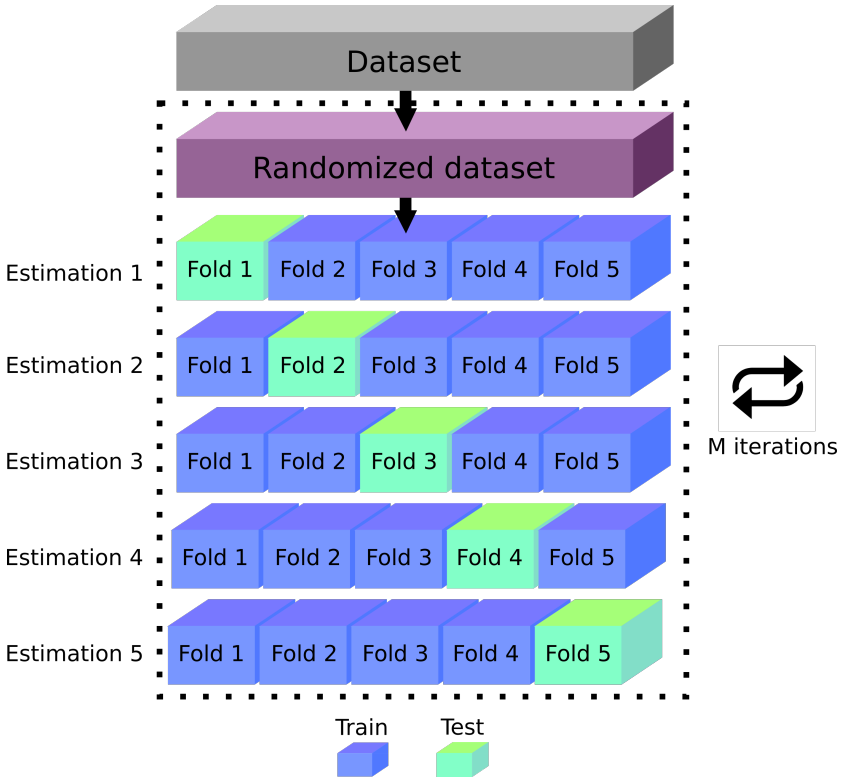


Figure 2.10. k-fold cross validation with $k = 5$ for M iterations.

2.5 Performance metrics

Performance metrics are used to evaluate and quantify the behavior and generalization capability of a model. These metrics are computed based on the outcomes of a model, comparing the real labels with the prediction labels by the model. In this research work, different metrics are considered for bi-class and multi-class classification problems.

Confusion matrix

A confusion matrix is an $n \times n$ matrix, where n is the number of classes in the classification problem. In this matrix are compared original labels with the ones predicted by the model. Table 2.1 shows the structure of the confusion matrix for bi-class classification problem ($n = 2$), where: True Positive (TP) is the number (or percentage) of samples correctly classified in the positive class (1); False Positive (FP) is the number (or percentage) of

samples misclassified in the positive class; True Negative (TN) is the number (or percentage) of samples correctly classified in the negative class (0); False Negative (FN) is the number (or percentage) of samples misclassified in the negative class.

Table 2.1. Confusion matrix for bi-class models.

Predicted			
	Positive 1	TP	FP
	Negative 0	FN	TN
		1 Positive	0 Negative

True

Table 2.1 shows the structure of the confusion matrix for a multi-class classification problem with three classes (0, 1, and 2). TP_0 is the number of TP samples taking class 0 as the positive class and the rest as negative class. E_{01} is the number of samples from class 0 that were incorrectly classified as class 1. Thus, the FN in the class 0 (FN_0) is $E_{01} + E_{02}$, and FP in the class 0 (FP_0) is $E_{10} + E_{20}$. In confusion matrix with dimension $n \times n$ there are n correct classifications a $n^2 - n$ possible errors. In both confusion matrices (bi-class and multi-class) the main diagonal indicates the success rate of the model while the rest indicates the error rate.

Table 2.2. Confusion matrix for multi-class problems

Predicted				
	0	TP_0	E_{10}	E_{20}
	1	E_{01}	TP_1	E_{21}
	2	E_{02}	E_{12}	TP_2
		0	1	2

True

Different performance metrics are derived of the confusion matrix, such as: accuracy, sensitivity, specificity, precision, and F1-Score.

Accuracy

Accuracy (Acc) is a performance metric widely used in different applications. For bi-class problems, this metric is defined as:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{2.10}$$

Acc is the ratio of the number of correct predictions (TP + TN) to the total number of examples (TP + FN + FP + TN). In multi-class classification problem the Acc is equal to the arithmetic mean of the Acc in each class. Equation 2.11 shows the Acc for a multi-class problem, where the N is the number of classes and Acc_i is the Acc when the class i is considered as the positive class and the rest of the classes are considered as the negative class.

$$\text{Acc} = \frac{1}{N} \sum_{i}^{N} Acc_i \tag{2.11}$$

This metric does not consider the number of samples in each class, therefore, so in unbalanced problems the accuracy can be high even if the model does not discriminate classes correctly. To overcome these limitations other performance metrics, such as the Unweighted Average Recall, have been proposed.

Unweighted Average Recall

Unweighted Average Recall (UAR) is also known as balance accuracy and is used when the problem is unbalanced. This metric is defined as:

$$\text{UAR} = \frac{1}{2} \left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right) \tag{2.12}$$

In balanced problems, the number of samples in the positive class (TP + FN) and the negative class (TN + FP) is the same, therefore the Acc = UAR. According to Equation 2.16, UAR is equal to the arithmetic mean of sensitivity and specificity.

Sensitivity or recall

Sensitivity (Sen) is the ratio between the number of samples correctly classified in the positive class and the number of samples in the positive class.

$$\text{Sen} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.13}$$

This metric indicates the system's capability to recognize positive class. This performance metric is also known as recall.

Specificity

Specificity (Spe) indicates the system's capability to recognize negative class. This metric is defines as:

$$\text{Spe} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.14}$$

Precision

Precision is the ratio between the number of samples correctly classified in the positive class (TP) and the number of samples predicted as positives (TP + FP). This metric indicates the quality of a positive prediction made by the model. Pre is also known as positive predictive value.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.15}$$

F1-Score

This metric is defined as the harmonic mean of the precision and recall. For multi-class classification problems the F-Score is the arithmetic mean of the F-Score computed in each class, similar to the accuracy in multi-class problems (Equation 2.11).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.16}$$

Receiver operating characteristic curve

Receiver Operating Characteristic (ROC) curve is a graphical representation of the binary classifier performance. This curve is the proportion of True Positive Rate (TPR) known as the Sensitivity and the False Positive Rate (FPR) define as (1 − Specificity), when decision threshold of a model is varied. Figure 2.11B shows the ROC curve. This curve is defined in the x-axis by FPR, and in the y-axis by TPR.

Figure 2.11A shows Gaussian distribution of classification scores, the decision threshold is the score at which a sample is classified as positive. Different areas can be identified in the graphic. Blue area corresponds to TN, red area corresponds TP and the intersection of both areas corresponds to the error of the model (FN + FP). The ROC curve is built based on the Gaussian distribution of the classification scores of a model. Each point of the ROC curve corresponds to the TPR and FPR for a specific decision threshold.

Area Under Curve (AUC) is a performance measure, which indicate the capability of the model to distinguish between classes. AUC is the area under curve ROC, which varies between 0 and 1. Values close to 1 means that the model distinguish better both classes, values close to 0.5 mean that the model predicts as well as chance. When, in Figure 2.11A, FN + FP = 0 the AUC i equal to 1.

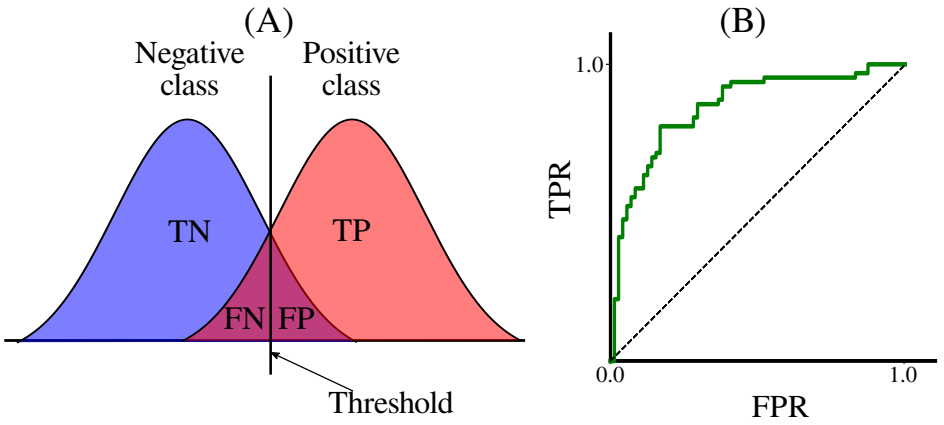


Figure 2.11. A: Gaussian distribution of classification scores. B: ROC curve. TPR True Positive Rate; FPR: False Positive Rate

Cohen's kappa coefficient

Cohen's kappa coefficient (κ) is used for measuring the agreement between two evaluators. Its origins are in the field of psychology but is also used as a performance metric in multi-class problems. In ML the Cohen's kappa coefficient measures the degree of agreement between the true values and the predicted values. This coefficient is defined as:

$$\kappa = \frac{p_o - p_\epsilon}{1 - p_\epsilon}, \tag{2.17}$$

where p_o is the empirical probability of agreement on the label assigned to any sample, and p_ϵ is the expected agreement when both evaluators assign labels at random [130]. κ can vary from -1 to $+1$, where according to [131] ≤ 0.4, $0.4 \leq \kappa \geq 0.75$, $\kappa > 0.75$ indicates a agreement, poor, good and excellent, respectively.

Chapter 3

Databases

Different corpora from social media and customer service centers are considered. In the experiments for DTs recognition are considered social media data from the PAN15 and PAN17 corpora, which are widely used in works related to author profiling and recognition of demographic variables. Additional data in a more formal language are considered for the DTs recognition scenario using a corpus composed of conversations between customers and advisors.
In the experiments where the effectiveness of chatbots is evaluated, conversations between customers and chatbots from two different Colombian companies are included. All corpora in this work are collected in real environments, i.e., tweets are from real authors, and conversations are transliterations of real interactions of customers.

3.1 Data for demographic trait recognition

3.1.1 PAN15

This corpus was collected for the 3rd Author Profiling Task PAN-CLEF 2015 competition [12]. The data include tweets in four languages: English, Spanish, Italian, and Dutch. The corpus has annotations about age, gender and the Big Five personality traits (openness, conscientiousness, extroversion, agreeableness, and emotional stability). In this work, we consider only the Spanish data of the corpus to compare the task of DTs recognition in informal and formal language using the same language. This corpus includes four groups of ages distributed as follows: 18-24, 25-34, 35-49, and 50-XX. The

corpus is divided into training and test set. The train and test set include data from 92 and 79 subjects, respectively. Table 3.1 shows the distribution of the data for each class and the average number of tweets per subject. The data is unbalanced among the classes, mainly because the number of subject older than 50 years old using Twitter is lower than in other groups of age.

Table 3.1. Distribution of Twitter users in terms of age classes

	18-24		**25-34**		**35-49**		**50-XX**	
	Subjects	Tweets	Subjects	Tweets	Subjects	Tweets	Subjects	Tweets
Train	21	100.0 ± 0.0	42	99.1 ± 4.1	22	98.6 ± 5.4	7	100.0 ± 0.0
Test	14	99.9 ± 0.3	44	97.5 ± 9.6	15	96.1 ± 14.2	6	100.0 ± 0.0

3.1.2 PAN17

This corpus was collected for the 5th Author Profiling Task PAN-CLEF 2017 competition [13]. The data include tweets in four languages: Arabic, English, Portuguese, and Spanish, for this study only Spanish data are considered. In this database, there are variants of Spanish from seven countries: Argentina, Chile, Colombia, Mexico, Peru, Spain and Venezuela. The training set is composed by tweets from 600 subjects from each country (300 female). Since each subject has 100 Tweets, there is a total of 4200 subjects and 420000 Tweets in the corpus. The test set comprises data from 400 subjects from each country (200 female) for a total of 2800 subjects and 280000 Tweets. This corpus is balanced in terms of LV and gender. For comparison with previous studies, we kept the original train and test sets. The training set was randomly divided into 80% for training and 20% to optimize the hyper-parameters of the models (development set). All data distribution was performed subject independent to avoid subject specific bias and to guarantee a better generalization capability of the models.

3.1.3 Conversations between customer and advisor

Conversations between customer and advisors of a pension administration company were collected. Texts are manually generated by a group of linguistic experts based on the audio signals from the customers. Similarly, the labels for age, gender and perceived dialect is assigned based on the audio

recordings processed by the linguists. Formal language is typically used by the customers when asking for a service, making a request, asking about certificates, and other questions about the service provided by the company. The average number of words per conversation is 602, with a standard deviation of 554. This corpus is drastically unbalanced. There are Colombian dialects and age groups with very few or invalid samples. For this reason, 3 sub-databases were built, depending on the DT: gender, age, and dialect. These three sub-databases have 42 common customers. The gender-corpus is composed of 220 samples (110 female), where the average number of words per conversation is 560 with a standard deviation of 479. The age corpus has two classes, young and adult, each one with 32 samples, where the average number of words per conversation is 571 with a standard deviation of 519. Finally, the dialect corpus has two classes according to internal Colombian dialects. The classes in the dialect corpus are "Antioqueño", which refers to the dialect from Antioquia-Colombia, and "Bogotano", which refers to the dialect from the center of the country. In the dialect corpus, each class has 80 samples, where the average number of words per conversation is 569 with a standard deviation of 401. A description of the sub-databases according to the DTs: gender, dialect, and age is shown in Table 3.2.

Table 3.2. The number of subjects in each sub-databases according to the DT. Invalid: subjects with an invalid label of the DT. Bog: number of subjects labeled as "Bogotanos". Ant: number of subjects labeled as "Antioqueños"

	Gender			Dialect			Age		
	Female	Male	Invalid	Bog	Ant	Invalid	Young	Adult	Invalid
Gender-corpus	**110**	**110**	0	71	70	79	24	177	19
Dialect-corpus	97	57	6	**80**	**80**	0	21	126	13
Age-corpus	40	23	1	28	19	17	**32**	**32**	0

3.2 Data to evaluate chatbot effectiveness (customer vs chatbot)

Two databases are considered to test the different methodologies for evaluation of effectiveness in conversations between humans and chatbots. Both contain conversations between chatbots and customers of two different companies in Colombia. A summary of the metadata for the two databases is provided in Table 3.3. Each conversation was labeled by a group of linguistic experts according to the effectiveness in the service/responses provided by the chatbot (i.e., effective vs. ineffective). Both databases were balanced according to effectiveness, conversations were randomly removed from the class with the most samples until the classes were balanced. Each corpus was designed with a specific semantic content because the companies offer different products and their chatbots were trained independently. The average number of words in the answers of the chatbots are different because their response structure is different. These databases allow to validate whether the approach proposed is general and suitable to be used in different types of chatbots trained for different markets, regardless of the semantic content or structure of their responses.

Table 3.3. Summary of information included in the two databases for customer vs chatbot.

	Chatbot DB1		**Chatbot DB2**	
	Effective	Ineffective	Effective	Ineffective
# Conv.	1768	1768	830	830
# Interactions per Conv.	5.53	5.29	2.93	5.76
# Word in questions	15.67	26.56	10.58	28.63
# Word in answers	230.35	248.60	61.97	138.46

3.2.1 Chatbot DB1

Conversations were collected from a chatbot trained to provide service to customers of a pension administration company. Most customers who interact with the chatbot are looking for information about an already purchased

service, account status, membership certificates, how to cancel a service, and others. This corpus consists of 3536 conversations between the chatbot and customers. Effective conversations have an average of 5.53 interactions (questions + answers) while the ineffective ones have an average of 5.29 interactions.

3.2.2 Chatbot DB2

This corpus contains 1660 conversations collected from a chatbot that provides customer service to a Telco company. The requirements of this chatbot are related to technical support or contracting telecommunication services. The chatbot should provide information about service plans, coverage, technical assistance, and others. In this case the chatbot answers have a simpler structure, therefore uses less words in its answers compared to DB1 database. The average number of words in effective and ineffective conversations are 2.93 and 5.76, respectively.

Chapter 4

Experiments and results

This chapter shows different experiments where information of a subject/customer based on text data is extracted. Experiments are divided according to two scenarios: DTs recognition and chatbot effectiveness evaluation. The aim in first scenario is to recognize three DTs: gender, Language Variety (LV) and age based on text with formal and informal language in order to customize marketing strategies. The second scenario aims to evaluate chatbot effectiveness to improve the automated customer services.

4.1 Demographic traits recognition

Figure 4.1 shows the methodologies to DTs recognition using two approaches, short and long texts. The short text approach consists in evaluating of short sequences of texts; thus, the architectures are trained and the DT (gender, LV or age) of the individual is computed based on the average classification scores of all short texts from the same individual. Note that for PAN17 and PAN15 corpus, each tweet is a short text, while for call-center transliterations each conversation is divided into chunks with 60 words, like in [41]. The long text approach consists in evaluating long texts. In this case, the complete text data from the individual is entered to the network at the same time. For the PAN17 and PAN15 corpus all Tweets per individual are concatenated, and for the call-center conversations we consider the complete transliteration of each conversation. Long text strategy is only evaluated using the CNN-based approach because longer segments produced vanishing gradient problems in the Bi-LSTM network.
The approaches are shown in Figure 4.1. Each approach is tested in texts with

informal and formal language. In analysis for text with informal language are used two benchmark corpus: PAN17, which has labels for gender and LV, and PAN15, which has labels for age. In analysis for text with formal language are used the three sub-databases from call-center conversation corpus.

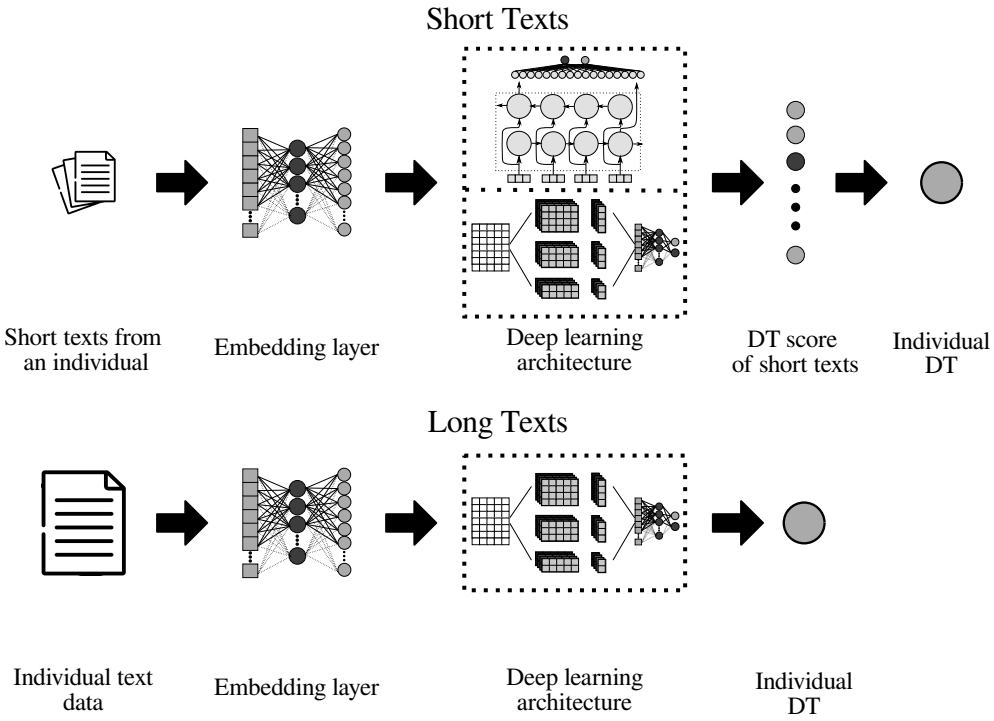


Figure 4.1. General methodologies to demographic traits recognition using short and long text.

Additionally, In this work is presented a cluster analysis based on k-means in order to perform a customer segmentation strategy, using the prediction scores of our neural networks. This analysis is focused especially on the Colombian DTs recognition for inter-country assessment using PAN17 corpus, and intra-country using the call-center conversations corpus. The number of clusters is chosen by using the elbow method and Kneedle algorithm [132].

4.1.1 Informal structured language (PAN17 and PAN15 corpus)

In experiments on age recognition, the train set is unbalanced, especially in the age group of people over 50 years of age, as shown in Table 3.1. Therefore, in this experiment was implemented a data augmentation strategy such that

all classes had 42 subjects. This number is chosen because is the number of subjects in the age group "25-34", which is the group with the largest number of subjects, it is shown in Table 3.1. In the data augmentation strategy random Tweets of the age group are taken, these Tweets are translated to other languages using the library deep-translator of Python [1]. The language is chosen randomly among: English, Chinese (traditional), French, Japanese, Korean, Afrikaans, Albanian, Czech, German, and Greek. Once the Tweet is translated into another language, it is translated back into Spanish. This data augmentation strategy is known as back translation and it is often used to generate more training data in NLP problems. Each new subject created with the data augmentation strategy is composed of 100 Tweets. The test set is not affected by the data augmentation strategy, because it is only used to train and optimize the model.

Table 4.1. Results of the gender, Language Variety (LV), and age classification in the PAN17 and PAN15 database. All values are given in %. **Acc**: Accuracy, **UAR**: Unweighted Average Recall

Gender (PAN17)						
	Acc per Tweet	Acc per Subject	UAR per Subject	Precision per Subject	Recall per Subject	F1-score per Subject
Short Bi-LSTM	60.5	71.3	71.3	69.6	72.0	70.8
Short CNN	61.1	71.4	71.4	81.1	68.0	73.9
Long CNN	-	**75.9**	**75.9**	**75.6**	**76.1**	**75.8**
LV (PAN17)						
	Acc per Tweet	Acc per Subject	UAR per Subject	F1-score per Subject	κ score per Subject	
Short Bi-LSTM	44.1	83.3	83.3	83.4	80.5	
Short CNN	48.5	89.8	89.8	89.8	88.0	
Long CNN	-	**92.3**	**92.3**	**92.3**	**91.0**	
Age (PAN15)						
	Acc per Tweet	Acc per Subject	UAR per Subject	F1-score per Subject	κ score per Subject	
Short Bi-LSTM	38.6	73.4	50.6	68.4	49.7	
Short CNN	47.2	73.4	50.3	69.1	49.2	
Long CNN	-	68.4	50.5	65.2	45.5	

The results obtained for the PAN17 and PAN15 corpus considering only

[1] https://github.com/nidhaloff/deep-translator

Spanish data are shown in Table 4.1. The analysis based on long texts to classify gender according to accuracy per subject shows an improvement of 4% compared to the one based on short texts. The improvement when classifying LV is about 3% using long texts with CNN. In the experiment on age recognition, the three models show a similar result, according to UAR. In this experiment, UAR is a more reliable measure of performance than accuracy, because class imbalance in the test set of PAN15.

Figure 4.2 shows the confusion matrix of each DT using the models generated with CNN using long texts. In the confusion matrix for LV recognition, the model recognizes all classes with accuracies over 90%, except in the LV from Venezuela, where 2.8% of the samples are classified as LV from Colombia, which makes sense because of the geographic proximity of the two countries. The recognition of LV from Spain and Mexico presents the best performance. In the confusion matrix for age recognition, the model is able to correctly recognize 84% and 71% of the samples in the age groups "25-34" and "18-24", respectively. The samples in the age group "35-50" are mainly confounded with samples in the age group "25-34". None of the 6 samples in the age group over 50 years achieve to be recognized.

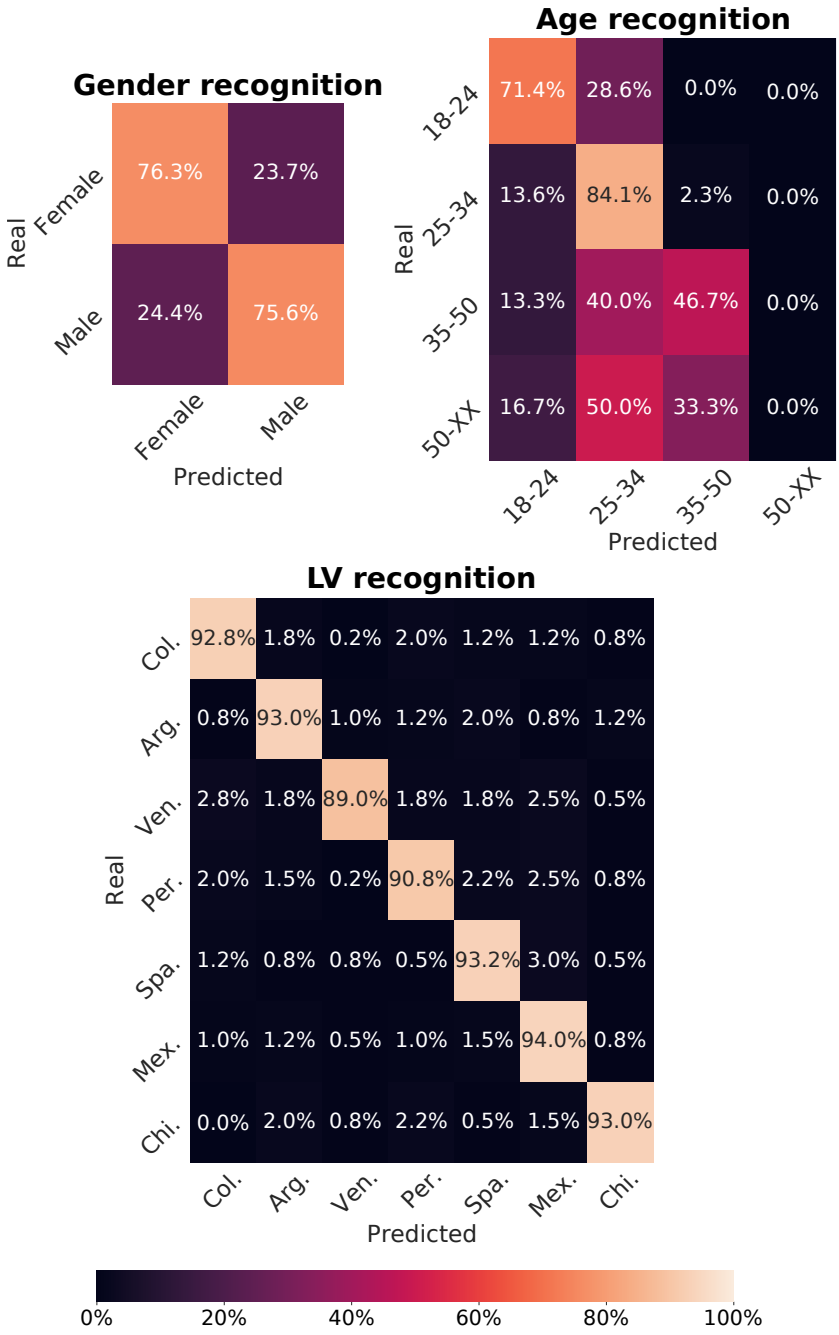


Figure 4.2. Normalized confusion matrix for each demographic trait, using the models generated with CNN using long text approach.

4.1.2 Formal structured language (call-center conversations corpus)

In experiments to DTs recognition for call-center conversations, two approaches are tested: (1) training the network only using the data from the corresponding corpus, and (2) training the model via TL by using a pre-trained model generated with the PAN15 or PAN17 corpus (depending on DT). For the transfer learning experiment, the best model for each approach (short and long text) is fine-tuned but freezing the embedding layer to keep the tokenizer and a larger vocabulary. Experiments without freezing the embedding layer were also performed but the results were not satisfactory. The motivation for using TL is to test whether the knowledge learned by a model trained with text data in informal language is useful to improve DTs classification systems based on text with formal language, where it is generally more difficult to collect a large amount of labeled data.

The results observed for transliterations from call center conversations are shown in Table 4.2. This table includes experiments based on short and long texts with and without applying TL strategy. The results for this corpus are obtained following a 10-fold cross-validation strategy due to the small size of the corpus. For gender and LV recognition, the highest accuracy is obtained here also with the long texts, in the same way as the results obtained with PAN17 corpus. In addition, note that, for both DTs (gender and LV), the accuracy improves by up to 13% when the TL strategy is applied. In the experiment for age recognition, the best accuracy is obtained with the CNN approach using short texts and training the model from scratch. Moreover, in experiments for age recognition, the TL strategy does not improve results in the target model.

The best models according to the accuracy per subject are based on CNN, for gender and LV with the long text approach with TL and for age recognition with the short text approach without TL. The best models are explored in Figure 4.3. Models show a similar result in the three DTs, however, the model for dialect recognition indicates a slightly better performance. In the experiment for age recognition, the distribution of the prediction scores shows the instability of the model for both classes, because the predicted scores of many sample are very close to the decision threshold (0.5).

Table 4.2. Results of the gender, Language Variety (LV) and Age classification in the call-center conversations data. All values are given in %.**TL**: Transfer Learning, **Acc**: Accuracy

		Acc per text	Acc per subject	Precision	Recall	F1-Score
Gender						
Short Bi-LSTM	no TL	52.7 ± 6.43	54.2 ± 10.1	65.3 ± 29.2	53.3 ± 22.7	55.2 ± 19.8
	TL	51.6 ± 5.07	56.4 ± 12.1	55.0 ± 13.8	56.7 ± 13.2	55.2 ± 11.9
Short CNN	no TL	57.9 ± 9.20	65.9 ± 12.7	52.0 ± 22.2	70.4 ± 17.2	57.8 ± 19.9
	TL	58.3 ± 6.48	62.9 ± 14.9	61.1 ± 17.9	64.6 ± 15.2	61.1 ± 15.7
Long CNN	no TL	-	56.9 ± 9.50	48.0 ± 27.0	54.9 ± 37.8	47.1 ± 28.1
	TL	-	**70.8 ± 11.6**	**74.5 ± 15.1**	**66.1 ± 14.9**	**69.2 ± 13.0**
Dialects						
Short Bi-LSTM	no TL	51.8 ± 12.1	55.8 ± 16.6	52.8 ± 27.5	71.5 ± 36.8	55.6 ± 25.9
	TL	57.3 ± 5.74	63.0 ± 11.1	62.9 ± 17.8	68.2 ± 22.8	62.3 ± 16.2
Short CNN	no TL	60.4 ± 7.90	66.2 ± 12.1	62.8 ± 19.7	75.4 ± 20.5	66.4 ± 16.7
	TL	59.8 ± 6.50	67.8 ± 12.0	68.2 ± 17.0	73.3 ± 22.2	67.7 ± 15.6
Long CNN	no TL	-	59.4 ± 14.5	54.4 ± 35.3	51.7 ± 35.3	48.1 ± 28.4
	TL	-	**72.8 ± 11.7**	**70.5 ± 15.5**	**75.9 ± 19.1**	**72.1 ± 15.2**
Age						
Short Bi-LSTM	no TL	54.3 ± 19.3	59.6 ± 25.8	62.3 ± 44.3	46.7 ± 37.1	50.0 ± 36.8
	TL	54.3 ± 13.3	54.8 ± 21.9	49.4 ± 40.9	44.3 ± 36.3	42.1 ± 32.2
Short CNN	no TL	**61.6 ± 17.6**	**68.8 ± 23.3**	**62.9 ± 32.6**	**62.9 ± 32.5**	**65.6 ± 29.7**
	TL	57.8 ± 16.6	65.4 ± 25.5	64.3 ± 35.4	64.3 ± 36.0	59.8 ± 32.0
Long CNN	no TL	-	50.7 ± 15.6	47.9 ± 31.0	64.4 ± 41.5	47.2 ± 28.2
	TL	-	52.4 ± 14.0	46.4 ± 21.9	63.6 ± 33.9	51.3 ± 24.8

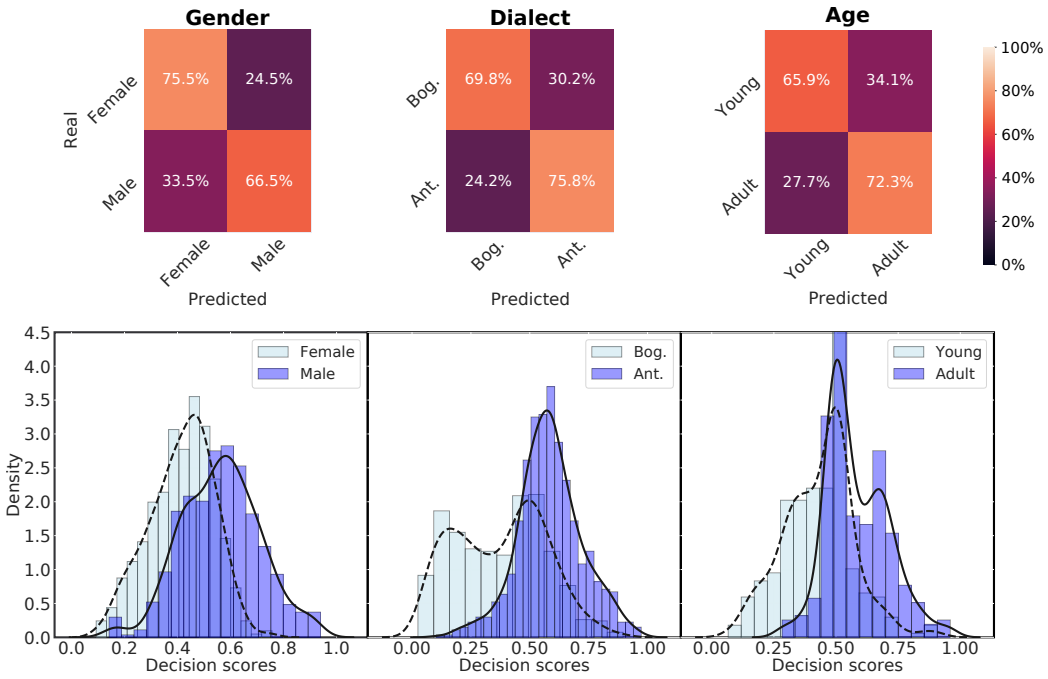


Figure 4.3. Normalized confusion matrix and distributions for the prediction scores from the best models for each demographic trait. For gender and LV recognition, the models based on CNN with long texts are used. For age recognition, the model based on CNN with short test is used.

4.1.3 Analysis of recognized Colombian DTs for user segmentation (inter vs intra country)

In this experiment are compared the demographic trait recognition inter and intra country. For analysis inter-country, PAN17 corpus is used, which has labels from gender and LV from each subject. For analysis intra-country are considered the customers with common samples in sub-databases of gender and dialect, there are a total of 130 subjects, distributed as 72 "Bogotanos" and 58 "Antioqueños", and 77 female and 53 male users. In this experiment is not considered the age because for analysis inter-country the PAN15 corpus does not have labels from LV according to the nationality.

Figure 4.4 shows the results of a cluster analysis using all samples of the test set from PAN17 corpus. In this experiment, the best resulting model from the previous experiments is used in order to perform user segmentation strategies. In this experiment are plotted the gender score in the horizontal axis vs. the probability of being classified as Colombian in the vertical axis. These data

are obtained at the output of the CNNs after the Softmax activation function. The results indicate the presence of three clusters, where 95.2% of subjects in cluster 1 are Colombian, while 97% of the subjects in clusters 2 and 3 are non-Colombian. Regarding gender, clusters 2 is mainly formed by female subjects (75.5%) while cluster 3 is formed by 75.2% male subjects. Cluster 1 does not have a dominant gender. In addition, note that the Colombian dialect recognition based on text is more accurate compared to gender recognition, although for non-Colombian samples each cluster is composed of at least 75% of persons with a specific gender.

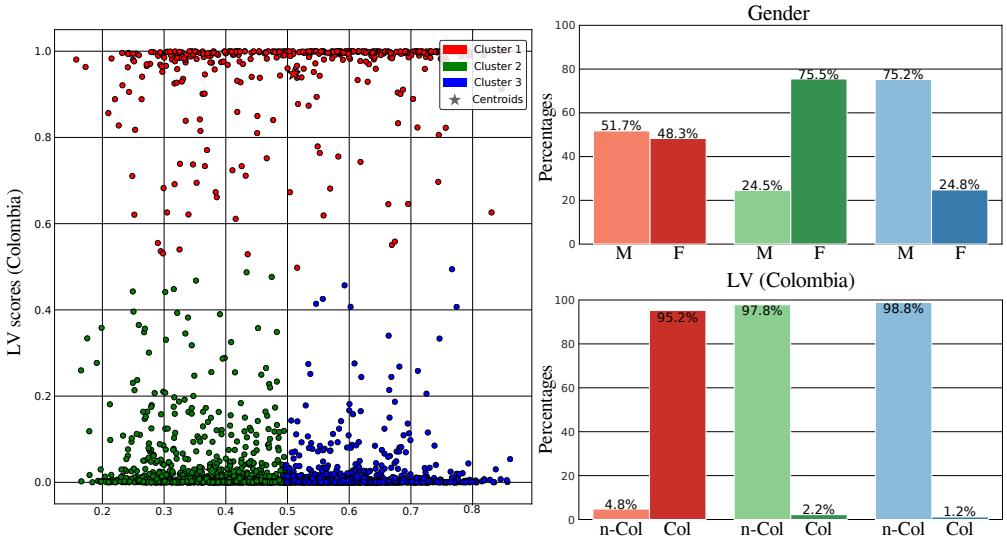


Figure 4.4. Results of k-means experiments using the scores of the best models for Colombian DTs recognition inter-country. M: Male, F: Female, LV: Language Variety, n-Col: non-Colombian, Col: Colombian.

Results of the intra-country analysis are shown in Figure 4.5. The best models of the experiments to gender and dialect recognition in call-center conversations were used. According to Figure 4.5, cluster 1 is composed mainly of subjects from Bogotá ("Bogotanos"), cluster 2 of subjects from Antioquia ("Antioqueños") and cluster 3 is slightly balanced in terms of dialect with a larger number of subjects from Bogotá. Regarding gender, only cluster 3 has a larger percentage of female subjects. The other two clusters are not gender discriminative.

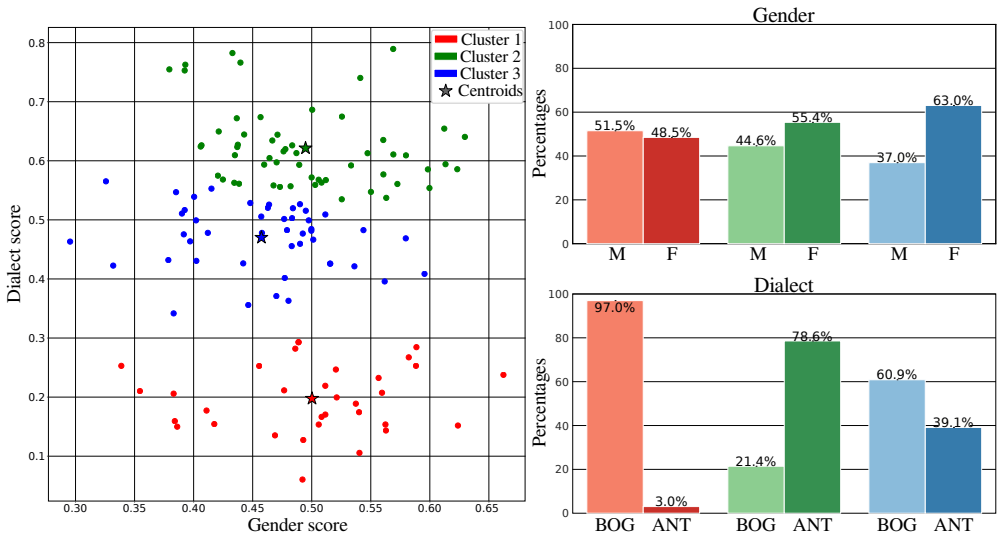


Figure 4.5. Results of k-means experiments using the scores of the best models to DTs recognition intra-country. M: Male, F: Female, BOG: Bogotano, ANT: Antioqueño.

4.1.4 Discussion

Results in experiments for DTs recognition indicate that architectures based on CNN exhibit higher accuracies than those based on RNN. This result could been because CNNs aim to model spatial information while RNNs aim to model sequential information, usually, CNNs obtain better results in tasks related to key-phrase recognition and RNNs in tasks where it is important to model the meaning of the document [133]. We believe for demographic traits recognition it is important to detect some key-phrases that provide information about the author, however this is something that should be validated in a future work. The approaches with long text appear to be successful to recognize DTs such as gender and LV, while approaches with short texts are effective to recognize age. The back translation strategy show a low accuracy in the age group "50-XX", where 83% of the training samples were created via data augmentation strategy and the model does not achieve to recognize any sample of the test set. However, in the age group "18-24", where 50% of the training samples were created via data augmentation strategy the model achieves to recognize 71.4% of the samples in the test set. Therefore, the data augmentation strategy is useful, when at least 50% of the training samples

of the model corresponds to texts form real authors.
In experiments with call-center conversations, the learning acquired by models in informal language is useful to improve the accuracy of the models for gender and LV recognition in documents with formal language. Even, the models to recognize LV in Spanish-speaking countries can be successfully used to fine-tune models to recognize more subtle LVs, such as the ones within the same country. TL strategy is not successful in age recognition likely because the model to age recognition in informal language (source model) is not accurate in adult age groups such as the age groups "35-50" and "50-XX". Hence, the learning acquired by the model in informal language does not help to improve the model in formal language, where it is necessary to discriminate between young and adult customers.
In experiments to compare the DTs recognition inter vs intra country, the subjects tend to be grouped according to their LV. This is better observed in the inter-country analysis, but it also occurs in the intra-country analysis. This can be explained because the differences in dialects within the same country are much subtle than the ones observed among different countries that share the same native language. In addition, for the inter-country scenario, gender-dependent clusters are created. Conversely, for the intra-country analysis, the clusters are more gender-balanced although in some clusters there is a slight tendency for a specific gender to appear.

4.2 Evaluation of effectiveness in chatbots

We evaluated the proposed P-CNN architecture upon the two databases described in Section 3. Results obtained with the proposed approach are compared with respect to those obtained with the baseline models. The datasets are divided into 70% for train and 30% for test making sure of class-balance in each subset. Optimal parameters are found using hold-out validation strategy. This validation is repeated 10 times with a random selection of train and test subsets, i.e., independent experiments.

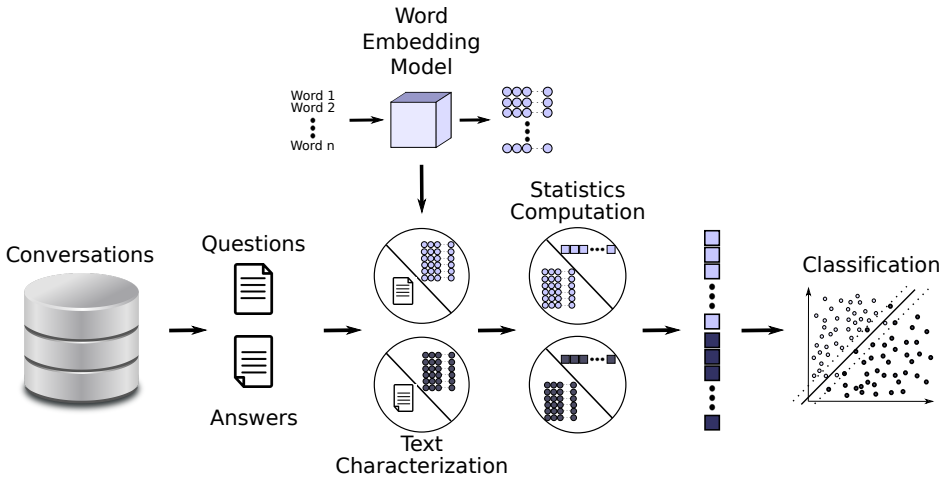


Figure 4.6. General methodology to evaluate the baseline models.

4.2.1 Evaluation of the baseline models

Context-dependent and context-independent embeddings are considered to evaluate the baseline approaches following the methodology depicted in Figure 4.6.

Results are reported in Table 4.3 in terms of Acc, Sen, Spe, AUC. Note that among the classical word embeddings 100-W2V is the one that yields better results in the two databases with accuracies of 76.0% and 79.8% for DB1 and DB2 databases, respectively. It is also worth that Spe is always higher in the two databases, indicating that the models are better to detect conversations where the chatbot was not able to provide a good service to the customer or was not unable to understand what the customer was asking for. This is actually a good characteristic because QoS areas in the companies are mainly focused on detecting problematic cases, such as those where the company or the service provider was not able to make the customer happy or satisfied. Figure 4.7 shows the results more compactly through the ROC curves obtained from experiments with each database separately and for each pre-trained word-embedding model.

Table 4.3. Results obtained with classical word embeddings.

	Database 1				Database 2			
	100-W2V	300-W2V	BERT	BETO	100-W2V	300-W2V	BERT	BETO
Acc. (%)	**76.04 ± 0.21**	73.68 ± 0.27	74.73 ± 1.39	71.95 ± 1.13	**79.80 ± 1.64**	78.13 ± 0.62	79.40 ± 0.79	63.15 ± 0.17
Sen. (%)	70.95 ± 0.27	67.19 ± 0.78	57.16 ± 4.57	55.33 ± 4.24	75.65 ± 0.41	76.01 ± 0.52	71.16 ± 2.69	49.36 ± 1.85
Spe. (%)	81.13 ± 0.19	80.17 ± 0.76	92.30 ± 1.84	88.57 ± 2.00	84.02 ± 0.37	80.24 ± 0.94	87.18 ± 1.97	76.95 ± 1.68
AUC	0.86	0.85	0.84	0.81	0.88	0.85	0.87	0.64
C; γ	10; 0.0001	1; 0.0001	1; 0.001	10; 1	1; 0.001	1; 0.0001	10; 0.0001	1; 1

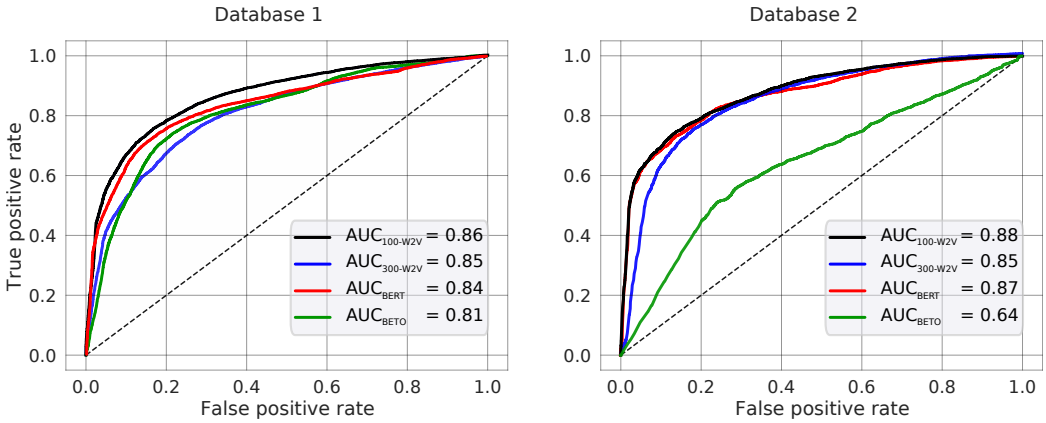
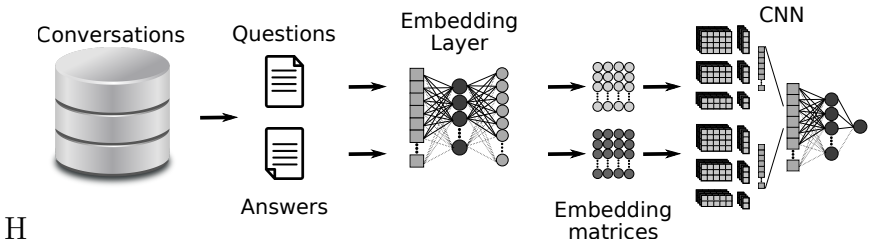


Figure 4.7. ROC curves obtained for database 1 (left) and database 2 (right) with classical word embeddings.

4.2.2 Evaluation of the parallel CNN

In this experiment each database is evaluated independently following the methodology shown in Figure 4.8. Results are reported in Table 4.4 in terms of Acc, Sen, Spe, and AUC.



H

Figure 4.8. General methodology of the proposed approach.

Table 4.4. Results obtained with the proposed CNN.

	Database 1	**Database 2**
Acc. (%)	79.0 ± 0.5	80.2 ± 1.5
Sens. (%)	74.6 ± 0.9	73.6 ± 3.3
Spec. (%)	83.4 ± 0.8	86.5 ± 1.9
AUC.	0.86	0.88

Accuracies of 79.0% and 80.2% are obtained for DB1 and DB2 databases, respectively. Note that the proposed approach yields better results in the two databases than those obtained with the baseline models (see Table 4.3). The main advantage of the proposed approach is that no pre-trained models were used to generate the different embeddings to build the input matrices. The embeddings are generated directly by the network through an embedding layer. Note that as in the case of the results with the baseline models, the performance is similar in both databases. Besides, specificity is also higher in the two cases, indicating that the proposed approach is more accurate to detect ineffective conversations. The ROC curves for these experiments are shown in Figure 4.9. Note that performance is similar in both cases, which likely indicates that the proposed method is equally suitable for the two databases considered in the chatbot evaluation scenario.

4.2.3 Comparison between the proposed approach and the baseline models

Besides classification experiments, Mann U Whitney tests are performed to evaluate the significance of the classification scores to discriminate the two classes: effective vs. ineffective conversations. Four cases are explored and results are summarized in Figure 4.10.

For the CNN approach the scores from the activation function in the output layer are used to perform the tests. The average of the scores was subtracted, thus they are zero-centered. In the baseline the distances of each sample to the optimal hyper-plane of the SVM are used as scores to perform the tests. Only the results of the W2V-100 embedding model are considered.

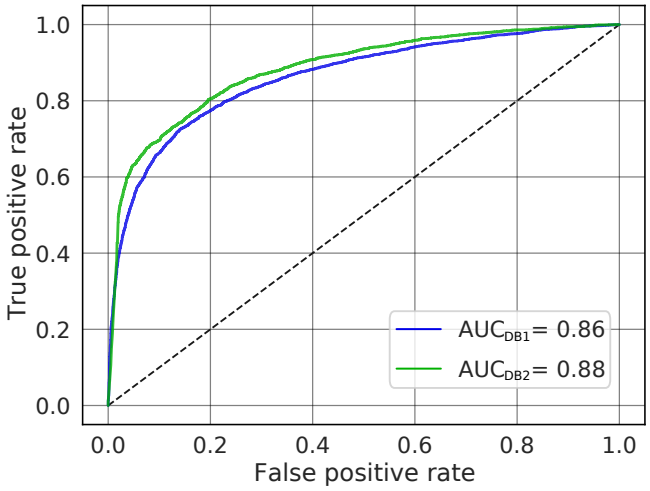


Figure 4.9. ROC curves obtained for database 1 (DB1) and database 2 (DB2) with the proposed approach.

According to the statistical tests, there are significant differences between the distribution of the two classes in both approaches for both databases (p-value $\ll 0.001$). Hence, in principle, the two methods (word embeddings and the proposed P-CNN) are equally suitable to classify between effective vs. ineffective conversations. Nevertheless, the classification results exhibited an improvement of 2.9% and 0.38% for DB1 and DB2 databases, respectively by using the proposed P-CNN in comparison with the baseline models.

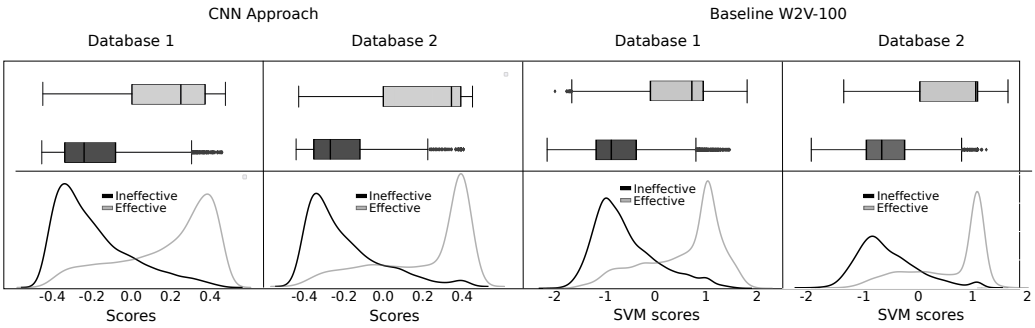


Figure 4.10. Distribution and box-plots of the scores obtained from the CNN approach and the model.

4.2.4 Discussion

The methodology proposed in this study aims to discriminate between effective and ineffective conversations. The proposed methodology allows word embeddings to be trained within the model itself, therefore specific terms commonly used within the company's or market's language have a representation according to the semantics of conversations and do not depend on the semantic of an external corpus. In addition, the vocabulary of the trained model is limited to the one used in the conversations of the training database. Therefore, if the model is tested in another environment, its performance will depend on the similarity of the semantic fields of the new environment with the vocabulary of the training database. This may cause a negative effect on its generalization capability; however, in general terms its advantages surpass this drawback, especially considering that the typical application scenario of this technology is when a company already has a chatbot in production and needs to evaluate the effectiveness of its service. In these cases, a training set can be generated to feed the P-CNN and create the evaluation model.
Regarding the experiments with the baseline models, we expected to obtain better results with modern pre-trained models such as BERT or BETO. However the evidence shows that a simpler model is sufficient to address the problem presented in this paper. This could be due to the specific application addressed in this work because in the two cases the customer knew that (s)he was interacting with a chatbot, thus more concrete and precise language with few context words was used.

It is noteworthy that the classification of conversations using the proposed model does not evaluate the quality of service perceived by the customer. The proposed system in this study is trained with information labeled by a human expert in linguistics based on whether the customer requirements were correctly addressed (effective conversation) or not. This means that the result of the system is in line with the evaluation that a human would assign to a given conversation. Hence, the result can be used to assess the effectiveness of a given chatbot in order to detect ineffective conversations, and then to re-train the chatbot to improve its capability to effectively provide service to the customers. Note that a model aligned with customer's feedback would be sensitive to capture customer satisfaction and not chatbot effectiveness. For instance, a conversation can be correctly addressed by the chatbot but the service that the chatbot provides can make the customer not to feel sat-

isfactorily served.
The main application of the system proposed in this work is to evaluate and to improve the chatbot's performance. On the one hand, the percentage of ineffective conversations detected by the model in a given time period can be useful to determine whether the chatbot's training was correct or if it is necessary to update some rules. On the other hand, the conversations classified as ineffective can be analyzed to determine the source of the chatbot's failure, and to update the rules.
A limitation of the proposed methodology is its computational complexity, which does not allow the real-time evaluation of conversations. However, it is possible to make evaluations by time intervals, for instance the number of ineffective conversations per hour, per day or per week, depending on the traffic of conversations processed by the chatbot.
One of the most relevant results of this research work has been the implementation of a software prototype in Technology Readiness Level 6 (TRL-6) to evaluate conversations between chatbots and customers. This prototype has different modalities, train, re-train and evaluate. In train and re-train mode, the system generate new models based on labeled conversations. In evaluation mode, the system takes a set of conversation unlabeled and a model generated by the system in the train or re-train mode, The evaluation mode generate a JSON report with the prediction of each conversation, and the percentage of effective and ineffective conversations.

Chapter 5

Conclusions

This research work proposes an automated approach to retrieve information from text that can be useful to customize and improve marketing and customer service strategies. This study covered two main scenarios: (1) automatic recognition of DTs from users/customers based on text data and (2) the evaluation of the effectiveness of chatbots in production environments.
For the first case, this work aimed to recognize three DTs from users: gender, LV, and age. The methodology is addressed both informal text data from social media posts, and based on formal text data collected from call center conversations. Different DL models are considered including CNNs and LSTMs. In addition, a TL approach was considered, where source models are pre-trained with data collected from social networks, and then fine-tuned with the call center conversations data, which have a more formal structure than the social media posts used for pre-training.
Results in the DTs recognition in text with informal language indicate that it is possible to classify the gender, LV and age of a subject based on his/her social media posts with accuracies of up to 75% and 92% for gender and LV recognition, respectively, and an UAR of up to 50% for age recognition. In documents with formal language, we obtained accuracies of up to 70%, 72% and 68%, for gender, LV in the same country, and age recognition, respectively. The use of a TL strategy improved the accuracy for gender and LV recognition. For age recognition the models trained from scratch are more accurate, which could be due to the fact that the source model for age apparently only learns correctly the age groups "18-24" and "25-34".
The results obtained in this work suggest that for the task of DTs recognition in informal and formal scenarios, architectures based on CNN exhibit higher

accuracies than those based on RNN. Furthermore, the results indicate that the approaches with long texts are useful to recognize gender and LV, and those with short texts are effective to recognize age. Experiments with call center conversations show that the learning acquired by the models to recognize LV in Spanish-speaking countries can be successfully used to fine-tune models to recognize more subtle LVs, such as the ones within the same country. In some cases, TL strategies generalize better than others where the neural networks are trained from scratch. TL could be a suitable strategy for companies or sectors where it is not possible to create large datasets from scratch. These results could be very positive since it is possible to benefit from large amounts of text data that are available in other domains like social networks.

For the second scenario, the effectiveness of chatbots during conversations with customers of two different companies is automatically evaluated in terms of whether the service requested by the customer was effectively addressed by the chatbot. Classical word-embedding approaches like Word2Vec and BERT are used as baseline and their performance is compared with respect to a novel approach, based on parallel CNNs with multiple temporal resolutions. Questions from customers and answers from the chatbots are modeled independently by two parallel convolutional layers. Each layer is composed of three filters, considering multiple temporal resolutions. *Bi*-gram, *tri*-gram and *four*-gram relationships among the words are considered simultaneously to extract the feature vector of the question and the answer.

The results indicate that the proposed approach achieves higher accuraccies than those obtained with baseline models in the two databases. Observed improvements range between 0.38 and 2.9 percentage points depending on the database. The main advantage of the proposed approach is that it does not depend on pre-trained models which are typically created with millions of words that are not necessarily related with the context of the given task (i.e., the target corpus). The proposed method allows to create specific models per each context, generating more accurate systems adapted to particular needs. These experiments and its results are a step forward in the automatic chatbot effectiveness evaluation and will allow companies to improve their QoS monitoring process. By using this approach it will not be necessary to use self-reported satisfaction surveys, instead, the service provided by the chatbot can be accurately and automatically evaluated.

Methodologies implemented in this work for DTs recognition can be extended

to other applications related to subject information retrieval such as personality, education level, and others, which would allow the building of more complete and specific subject/customer profiles, in order to make the customer feel more familiar with the agent. In addition, positive results obtained with TL strategies motivate the development of similar methodologies in other important tasks for the industry such as document analysis, sentiment analysis, and among others.

The system proposed for automatic evaluation of effectiveness use all collected conversations to measure the chatbot effectiveness and the system is able to accurately identify ineffective conversations, thus they can be analyzed properly to update the chatbot rules in a retraining process. Other challenges like naturalness and personality for the chatbots' language might be studied in future research.

Appendix A
Conferences & Publications

The following publications were derived from the development of this research work.

A.1 Journals

- ✓ **D. Escobar-Grisales**, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave. "Recognition of demographic traits in informal and formal language scenarios vis transfer learning." *TecnoLógicas*, under review.
- ✓ **D. Escobar-Grisales**, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave. "Evaluation of effectiveness in conversations between humans and chatbots using parallel convolutional neural networks with multiple temporal resolutions." *Multimedia Tools and Applications*, under review.
- ✓ J. R. Orozco-Arroyave, J. C. Vásquez-Correa, P. Klumpp, P. A. Pérez-Toro, **D. Escobar-Grisales**, N. Roth, C. D. Rios-Urrego, et al., "Apkinson: the smartphone application for telemonitoring Parkinson's patients through speech, gait and hands movement." *Neurodegenerative Disease Management*, 10(3), 2020, pp. 137-157.
- ✓ **D. Escobar-Grisales**, J. C. Vásquez-Correa, J. F Vargas-Bonilla, and J. R. Orozco-Arroyave. "Identity verification in virtual education using biometric analysis based on keystroke dynamics." *TecnoLógicas*, 2020, vol. 23, no 47, p. 193-207.

A.2 Book Chapters

✓ **D. Escobar-Grisales**, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave. "Gender Recognition in Informal and Formal Language Scenarios via Transfer Learning." *Workshop on Engineering Applications (WEA 2021)*. Springer, Cham, 2021

A.3 Conferences

✓ **D. Escobar-Grisales**, C. D. Rios-Urrego, D. A López-Santander, J. D. Gallo-Aristizabal, J. C. Vásquez-Correa, E. Nöth, and J. R. Orozco-Arroyave. "Colombian Dialect Recognition Based on Information Extracted From Speech and Text Signals." in *Proceeding IEEE Automatic Speech Recognition and Understanding ASRU*, 2021, Accepted.

✓ J. C. Vásquez-Correa, T. Arias-Vergara, P. Klumpp, M. Strauss, A. Küderle, N. Roth, S. Bayerl, N. Garcıa-Ospina, P. A. Pérez-Toro, L. F. Parra-Gallego, C. D. Rios-Urrego, **D. Escobar-Grisales**, J. R. Orozco-Arroyave, B. Eskofier, and E. Nöth, "Apkinson: a Mobile Solution for Multimodal Assessment of Patients with Parkinson's Disease". in *Proceedings Interspeech*, 2019, (pp. 964-965).

List of Figures

Bibliography

[1] E. Cramer-Flood, *Global ecommerce update 2021: Worldwide ecommerce will approach $5 trillion this year*.

[2] BIGCOMMERCE, *14 ecommerce trends leading the way*, 2020. [Online]. Available: https://www.bigcommerce.com/articles/ecommerce/ecommerce-trends/#14-ecommerce-trends-leading-the-way.

[3] M. Abraham, J. Gonzales, R. Archacki, and S. Fanfarillo, "The next level of personalization in retail," *Boston Consulting Group (BCG)*, 2019.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent systems*, vol. 28, no. 2, pp. 15–21, 2013.

[5] D. Fernandez-Lanvin, J. de Andres-Suarez, M. Gonzalez-Rodriguez, and B. Pariente-Martinez, "The dimension of age and gender as user model demographic factors for automatic personalization in e-commerce sites," *Computer Standards & Interfaces*, vol. 59, pp. 1–9, 2018.

[6] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "Superagent: A customer service chatbot for e-commerce websites," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 97–102.

[7] M. S. B. Mimoun, I. Poncin, and M. Garnier, "Case study—embodied virtual agents: An analysis on reasons for failure," *Journal of Retailing and Consumer services*, vol. 19, no. 6, pp. 605–612, 2012.

[8] A. Janssen, L. Grützner, and M. H. Breitner, "Why do chatbots fail? a critical success factors analysis," in *Forty-Second International Conference on Information Systems*, 2021.

[9] J. Feine, S. Morana, and U. Gnewuch, "Measuring service encounter satisfaction with customer service chatbots using sentiment analysis," in *Proceedings of the 14th International Conference on Wirtschaftsinformatik (WI2019)*, 2019.

[10] O. Dogan and B. Oztaysi, "Gender prediction from classified indoor customer paths by fuzzy c-medoids clustering," in *International Conference on Intelligent and Fuzzy Systems (INFUS)*, Springer, 2019, pp. 160–169.

[11] R. Hirt, N. Kühl, and G. Satzger, "Cognitive computing for customer profiling: Meta classification for gender prediction," *Electronic Markets*, vol. 29, no. 1, pp. 93–106, 2019.

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

[13] F. Rangel, P. Rosso, M. Potthast, and B. Stein, "Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter," *Working notes papers of the CLEF*, pp. 1613–0073, 2017.

[14] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, and G. Inches, "Overview of the author profiling task at pan 2013," in *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, CELCT, 2013, pp. 352–365.

[15] F. Rangel, P. Rosso, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, W. Daelemans, *et al.*, "Overview of the 2nd author profiling task at pan 2014," in *CEUR Workshop Proceedings*, CEUR Workshop Proceedings, vol. 1180, 2014, pp. 898–927.

[16] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein, "Overview of the 4th author profiling task at pan 2016: Cross-genre evaluations," *Working Notes Papers of the CLEF*, vol. 2016, pp. 750–784, 2016.

[17] E. Stamatatos, W. Daelemans, B. Verhoeven, P. Juola, A. López-López, M. Potthast, and B. Stein, "Overview of the author identification task at pan 2014.," *CLEF (Working Notes)*, vol. 1180, pp. 877–897, 2014.

[18] I. Markov, H. Gómez-Adorno, and G. Sidorov, "Language-and subtask-dependent feature selection and classifier parameter tuning for author profiling," in *Notebook for PAN at Conference and Labs of the Evaluation Forum (CLEF)*, 2017.

[19] M. Martinc, I. Skrjanec, K. Zupan, and S. Pollak, "Pan 2017: Author profiling-gender and language variety prediction.," in *CLEF (Working Notes)*, 2017.

[20] A. Basile, G. Dwyer, M. Medvedeva, J. Rawee, H. Haagsma, and M. Nissim, "N-gram: New groningen author-profiling model," *arXiv preprint arXiv:1707.03764*, 2017.

[21] M. Potthast, T. Gollub, F. Rangel, P. Rosso, E. Stamatatos, and B. Stein, "Improving the reproducibility of pan's shared tasks," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2014, pp. 268–299.

[22] M. L. Newman, C. J. Groom, L. D. Handelman, and J. W. Pennebaker, "Gender differences in language use: An analysis of 14,000 text samples," *Discourse Processes*, vol. 45, no. 3, pp. 211–236, 2008.

[23] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in twitter," in *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, 2010, pp. 37–44.

[24] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, *et al.*, "Personality, gender, and age in the language of social media: The open-vocabulary approach," *PloS one*, vol. 8, no. 9, e73791, 2013.

[25] S. Daneshvar and D. Inkpen, "Gender identification in twitter using n-grams and lsa," in *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*, 2018.

[26] A. Bacciu, M. La Morgia, A. Mei, E. N. Nemmi, V. Neri, and J. Stefa, "Bot and gender detection of twitter accounts using distortion and lsa.," in *CLEF (Working Notes)*, 2019.

[27] A. P. López-Monroy, M. Montes-y-Gómez, H. J. Escalante, and L. V. Pineda, "Using intra-profile information for author profiling.," in *CLEF (Working Notes)*, 2014, pp. 1116–1120.

[28] M. E. Aragón and A. P. López-Monroy, "Author profiling and aggressiveness detection in spanish tweets: Mex-a3t 2018.," in *IberEval@ SEPLN*, 2018, pp. 134–139.

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

[30] W. Li and M. Dickinson, "Gender prediction for chinese social media data.," in *Conference Recent Advances in Natural Language Processing (RANLP)*, 2017, pp. 438–445.

[31] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (ANPIS)*, 2013, pp. 3111–3119.

[32] F. Hsieh, R. Dias, and I. Paraboni, "Author profiling from facebook corpora," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

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

[34] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 655–665. DOI: 10.3115/v1/P14-1062. [Online]. Available: https://www.aclweb.org/anthology/P14-1062.

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

[36] S. Ruder, P. Ghaffari, and J. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *ArXiv*, vol. abs/1609.06686, 2016.

[37] H. Gómez-Adorno, R. Fuentes-Alba, I. Markov, G. Sidorov, and A. Gelbukh, "A convolutional neural network approach for gender and language variety identification," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 4845–4855, 2019.

[38] M. Á. Álvarez-Carmona, E. Guzmán-Falcón, M. Montes-y Gómez, H. J. Escalante, L. Villasenor-Pineda, V. Reyes-Meza, and A. Rico-Sulayes, "Overview of mex-a3t at ibereval 2018: Authorship and aggressiveness analysis in mexican spanish tweets," in *Notebook papers of 3rd sepln workshop on evaluation of human language technologies for iberian languages (ibereval), seville, spain*, vol. 6, 2018.

[39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv:1607.01759*, 2016.

[40] J. Zhong and W. Li, "Predicting customer churn in the telecommunication industry by analyzing phone call transcripts with convolutional neural networks," in *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence (ICAI*, 2019, pp. 55–59.

[41] D. Kodiyan *et al.*, "Author profiling with bidirectional rnns using attention with grus.," in *Conference and Labs of the Evaluation Forum (CLEF)*, RWTH Aachen, vol. 1866, 2017.

[42] M. González Bermúdez, "An analysis of twitter corpora and the differences between formal and colloquial tweets," in *Proceedings of the Tweet Translation Workshop 2015*, CEUR-WS. org, 2015, pp. 1–7.

[43] J. Gu and Z. Yu, "Data annealing for informal language understanding tasks," *arXiv:2004.13833*, 2020.

[44] M. Nuruzzaman and O. K. Hussain, "A survey on chatbot implementation in customer service industry through deep neural networks," in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, IEEE, 2018, pp. 54–61.

[45] M.-C. Jenkins, R. Churchill, S. Cox, and D. Smith, "Analysis of user interaction with service oriented chatbot systems," in *International Conference on Human-Computer Interaction (HCI)*, Springer, 2007, pp. 76–83.

[46] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, 2009, pp. 1387–1396.

[47] C. Chakrabarti and G. F. Luger, "A framework for simulating and evaluating artificial chatter bot conversations," in *The Twenty-Sixth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2013.

[48] V. Hung, M. Elvir, A. Gonzalez, and R. DeMara, "Towards a method for evaluating naturalness in conversational dialog systems," in *2009 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, IEEE, 2009, pp. 1236–1241.

[49] R. Zhao, O. J. Romero, and A. Rudnicky, "Sogo: A social intelligent negotiation dialogue system," in *Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA)*, 2018, pp. 239–246.

[50] B. Heller, M. Proctor, D. Mah, L. Jewell, and B. Cheung, "Freudbot: An investigation of chatbot technology in distance education," in *EdMedia+ Innovate Learning*, Association for the Advancement of Computing in Education (AACE), 2005, pp. 3913–3918.

[51] R. P. Schumaker, M. Ginsburg, H. Chen, and Y. Liu, "An evaluation of the chat and knowledge delivery components of a low-level dialog system: The az-alice experiment," *Decision Support Systems*, vol. 42, no. 4, pp. 2236–2246, 2007.

[52] D. Peras, "Chatbot evaluation metrics," *Economic and Social Development: Book of Proceedings*, pp. 89–97, 2018.

[53] P. Jwalapuram, "Evaluating dialogs based on grice's maxims," in *Proceedings of the Student Research Workshop associated with Recent Advances in Natural Language Processing (RANLP)*, 2017, pp. 17–24.

[54] J. Cahn, "Chatbot: Architecture, design, & development," *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*, 2017.

[55] R. L. Oliver, *Satisfaction: A behavioral perspective on the consumer: A behavioral perspective on the consumer*. Routledge, 2014.

[56] Y. Xiang, Y. Zhang, X. Zhou, X. Wang, and Y. Qin, "Problematic situation analysis and automatic recognition for chinese online conversational system," in *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 2014, pp. 43–51.

[57] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[58] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out Association for Computational Linguistics (ACL)*, 2004, pp. 74–81.

[59] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[60] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv:1503.02364*, 2015.

[61] Y. Song, R. Yan, X. Li, D. Zhao, and M. Zhang, "Two are better than one: An ensemble of retrieval-and generation-based dialog systems," *arXiv:1610.07149*, 2016.

[62] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," *arXiv:1506.06714*, 2015.

[63] K. Yao, B. Peng, G. Zweig, and K.-F. Wong, "An attentional neural conversation model with improved specificity," in *Proceedings of the 53th annual meeting of the Association for Computational Linguistics (ACL)*, 2016.

[64] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2015)*, 2015.

[65] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1, 2016, pp. 994–1003.

[66] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[67] A. Venkatesh, C. Khatri, A. Ram, F. Guo, R. Gabriel, A. Nagar, R. Prasad, M. Cheng, B. Hedayatnia, A. Metallinou, *et al.*, "On evaluating and comparing conversational agents," *arXiv:1801.03625*, vol. 4, pp. 60–68, 2018.

[68] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[69] J. Jia, "Csiec: A computer assisted english learning chatbot based on textual knowledge and reasoning," *Knowledge-Based Systems*, vol. 22, no. 4, pp. 249–255, 2009.

[70] J. Pereira and O. Dıéaz, "A quality analysis of facebook messenger's most popular chatbots," in *Proceedings of the 33rd annual ACM Symposium on Applied Computing (SAC)*, 2018, pp. 2144–2150.

[71] X. Gu, K. Cho, J.-W. Ha, and S. Kim, "Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder," in *International Conference on Learning Representations (ICLR)*, 2018.

[72] J. Sedoc, D. Ippolito, A. Kirubarajan, J. Thirani, L. Ungar, and C. Callison-Burch, "Chateval: A tool for chatbot evaluation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 60–65.

[73] Z. Xu, B. Liu, B. Wang, C.-J. Sun, X. Wang, Z. Wang, and C. Qi, "Neural response generation via gan with an approximate embedding layer," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP*, 2017, pp. 617–626.

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

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

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

[77] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-based sentiment analysis using bert," in *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa)*, 2019, pp. 187–196.

[78] P. A. Pérez-Toro, J. C. Vásquez-Corre, T. Arias-Vergara, P. Klumpp, J. R. Orozco-Arroyave, and E. Nöth, "Acoustic and linguistic analyses to assess early-onset and genetic alzheimer's disease," *IEEE International Conference on Acoustics, Speech and Signal Processing, in press*, 2021.

[79] M. Rodrigues Makiuchi, T. Warnita, K. Uto, and K. Shinoda, "Multimodal fusion of bert-cnn and gated cnn representations for depression detection," in *Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop (AVEC)*, 2019, pp. 55–63.

[80] M. Conway, M. Hu, and W. W. Chapman, "Recent advances in using natural language processing to address public health research questions using social media and consumergenerated data," *Yearbook of medical informatics*, vol. 28, no. 01, pp. 208–217, 2019.

[81] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107 134, 2021.

[82] A. Pinto, H. Gonçalo Oliveira, and A. Oliveira Alves, "Comparing the performance of different nlp toolkits in formal and social media text," in *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[83] H. Jiang, B. Dai, M. Yang, W. Wei, and T. Zhao, "Towards automatic evaluation of dialog systems: A model-free off-policy evaluation approach," in *EMNLP*, 2021.

[84] A. Ghandeharioun, J. H. Shen, N. Jaques, C. Ferguson, N. Jones, A. Lapedriza, and R. Picard, "Approximating interactive human evaluation with self-play for open-domain dialog systems," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[85] K. Georgila, C. Gordon, H. Choi, J. Boberg, H. Jeon, and D. Traum, "Toward low-cost automated evaluation metrics for internet of things dialogues," in *9th International Workshop on Spoken Dialogue System Technology*, Springer, 2019, pp. 161–175.

[86] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*. Chapman and Hall/CRC, 2010.

[87] C. Lu, Y. Bu, J. Wang, Y. Ding, V. Torvik, M. Schnaars, and C. Zhang, "Examining scientific writing styles from the perspective of linguistic complexity," *Journal of the Association for Information Science and Technology*, vol. 70, no. 5, pp. 462–475, 2019.

[88] L. Meng and M. Huang, "Dialogue intent classification with long short-term memory networks," in *National CCF Conference on Natural Language Processing and Chinese Computing*, Springer, 2017, pp. 42–50.

[89] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy, "Emotion recognition in conversation: Research challenges, datasets, and recent advances," *IEEE Access*, vol. 7, pp. 100 943–100 953, 2019.

[90] K. Imamura and E. Sumita, "Recycling a pre-trained bert encoder for neural machine translation," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 23–31.

[91] D. Qiu, H. Jiang, and S. Chen, "Fuzzy information retrieval based on continuous bag-of-words model," *Symmetry*, vol. 12, no. 2, p. 225, 2020.

[92] K. M. Alomari, H. M. ElSherif, and K. Shaalan, "Arabic tweets sentimental analysis using machine learning," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, 2017, pp. 602–610.

[93] G. Orellana, B. Arias, M. Orellana, V. Saquicela, F. Baculima, and N. Piedra, "A study on the impact of pre-processing techniques in spanish and english text classification over short and large text documents," in *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, IEEE, 2018, pp. 277–283.

[94] A. M. Ebrahimi and A. A. Barforoush, "Preprocessing role in analyzing tweets towards requirement engineering," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, IEEE, 2019, pp. 1905–1911.

[95] N. Babanejad, A. Agrawal, A. An, and M. Papagelis, "A comprehensive analysis of preprocessing for word representation learning in affective tasks," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 5799–5810.

[96] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, 2014.

[97] F. Rangel, P. Rosso, A. Charfi, W. Zaghouani, B. Ghanem, and J. Snchez-Junquera, "Overview of the track on author profiling and deception detection in arabic," in *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS. org, Kolkata, India*, 2019.

[98] R. Bayot and T. Gonçalves, "Multilingual author profiling using word embedding averages and svms," in *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, IEEE, 2016, pp. 382–386.

[99] A. Bakarov, "A survey of word embeddings evaluation methods," *arXiv:1801.09536*, 2018.

[100] J. Yan, *Text representation.* 2009.

[101] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus," *Data Science and Engineering*, vol. 4, no. 2, pp. 157–175, 2019.

[102] H. Caselles-Dupré, F. Lesaint, and J. Royo-Letelier, "Word2vec applied to recommendation: Hyperparameters matter," in *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*, 2018, pp. 352–356.

[103] S. Reese, G. Boleda Torrent, M. Cuadros Oller, L. Padró, and G. Rigau Claramunt, "Word-sense disambiguated multilingual wikipedia corpus," in *7th International Conference on Language Resources and Evaluation (LREC)*, 2010.

[104] A. Alwehaibi and K. Roy, "Comparison of pre-trained word vectors for arabic text classification using deep learning approach," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2018, pp. 1471–1474.

[105] S. Bhoir, T. Ghorpade, and V. Mane, "Comparative analysis of different word embedding models," in *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, IEEE, 2017, pp. 1–4.

[106] N. Sayer, "Google code archive-long-term storage for google code project hosting," *XP055260798, Retrieved from the Internet [retrieved on 20160323]*, 2014.

[107] X. Rong, "Word2vec parameter learning explained," *arXiv 1411.2738*, 2014.

[108] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv:1802.05365*, 2018.

[109] A. Miaschi and F. Dell'Orletta, "Contextual and non-contextual word embeddings: An in-depth linguistic investigation," in *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP)*, 2020, pp. 110–119.

[110] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems (NIPS*, 2017, pp. 5998–6008.

[111] J. Tiedemann, "Parallel data, tools and interfaces in opus.," in *Lrec*, vol. 2012, 2012, pp. 2214–2218.

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

[113] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" *arXiv preprint arXiv:1804.06323*, 2018.

[114] F. Rangel, P. Rosso, M. Montes-y-Gómez, M. Potthast, and B. Stein, "Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter," *Working Notes Papers of the CLEF*, pp. 1–38, 2018.

[115] J. R. Orozco-Arroyave *et al.*, "Automatic detection of hypernasal speech of children with cleft lip and palate from spanish vowels and words using classical measures and nonlinear analysis," *Revista Facultad de Ingenieríéa Universidad de Antioquia*, no. 80, pp. 109–123, 2016.

[116] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1.

[117] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[118] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[119] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," *arXiv:1805.04174*, 2018.

[120] Z. Gan, Y. Pu, R. Henao, C. Li, X. He, and L. Carin, "Learning generic sentence representations using convolutional neural networks," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2390–2400.

[121] S. Tang, H. Jin, C. Fang, Z. Wang, and V. R. de Sa, "Exploring asymmetric encoder-decoder structure for context-based sentence representation learning," 2018.

[122] L. Luo, "Network text sentiment analysis method combining lda text representation and gru-cnn," *Personal and Ubiquitous Computing*, vol. 23, no. 3-4, pp. 405–412, 2019.

[123] S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models," *rXiv:1904.04206*, 2019.

[124] H. Kim and Y.-S. Jeong, "Sentiment classification using convolutional neural networks," *Applied Sciences*, vol. 9, no. 11, p. 2347, 2019.

[125] M. Gridach, H. Haddad, and H. Mulki, "Churn identification in microblogs using convolutional neural networks with structured logical knowledge," in *Proceedings of the 3rd Workshop on Noisy User-generated Text (WNUT)*, 2017, pp. 21–30.

[126] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavvaf, and E. A. Fox, "Natural language processing advancements by deep learning: A survey," *arXiv:2003.01200*, 2020.

[127] D. W. Otter *et al.*, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. on Neural Networks and Learning Systems (TNNLS)*, 2020.

[128] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[129] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[130] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computational Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.

[131] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical methods for rates and proportions*. john wiley & sons, 2013.

[132] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st international conference on distributed computing systems workshops*, IEEE, 2011, pp. 166–171.

[133] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.