UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA

UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LARISSA LUCENA VASCONCELOS

FEATURE EXTRACTION FROM TEXT FLOWS BASED ON SEMANTIC SIMILARITY FOR CLASSIFICATION TASKS: AN APPROACH INSPIRED BY AUDIO ANALYSIS

CAMPINA GRANDE – PB

2022

Federal University of Campina Grande

Electrical Engineering and Computer Center

Postgraduate Coordination in Computer Science

Feature Extraction from Text Flows Based on Semantic Similarity for Classification Tasks: an Approach Inspired by Audio Analysis

Larissa Lucena Vasconcelos

Thesis submitted to the Coordination of the Postgraduate Course in Computer Science at the Federal University of Campina Grande - Campus I as part of the necessary requirements to obtain the degree of Doctor in Computer Science.

Concentration Area: Computer Science

Research Line: Natural Language Processing

Claudio Elízio Calazans Campelo

(Supervisor)

Campina Grande, Paraíba, Brazil

V331f Vasconcelos, Larissa Lucena.

Feature extraction from text flows based on semantic similarity for classification tasks: an approach inspired by audio analysis / Larissa Lucena Vasconcelos. – Campina Grande, 2022.

104 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2022.

"Orientação: Prof. Dr. Claudio Elízio Calazans Campelo".

Referências.

1. Natural Language Processing. 2. Classification. 3. Semantic Similarity. 4. Feature Extraction. 5. Computer Science. I. Campelo, Claudio Elízio Calazans. II. Título.

CDU 004.43:81’322.2(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

LARISSA LUCENA VASCONCELOS

FEATURE EXTRACTION FROM SEMANTIC SIMILARITY FLOWS FOR TEXT CLASSIFICATION TASKS: AN APPROACH INSPIRED BY AUDIO ANALYSIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Doutor em Ciência da Computação.

Aprovada em: 18/03/2022

Prof. Dr. CLÁUDIO ELÍZIO CALAZANS CAMPELO, UFCG, Orientador

Profa. Dra. JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, UFCG, Examinadora Interna

Prof. Dr. LEANDRO BALBY MARINHO, UFCG, Examinador Interno

Prof. Dr. ADRIANO ALONSO VELOSO, UFMG, Examinador Externo

Prof. Dr. RINALDO JOSÉ DE LIMA, UFRPE, Examinador Externo

Documento assinado eletronicamente por **JOSEANA MACEDO FECHINE**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/04/2022, às 10:22, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **LEANDRO BALBY MARINHO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/04/2022, às 10:22, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **CLAUDIO ELIZIO CALAZANS CAMPELO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 19/04/2022, às 10:40, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Referência: Processo nº 23096.011087/2022-48 SEI nº 2289931

Resumo

A classificação de texto é um dos principais desafios investigados na pesquisa em Processamento de Linguagem Natural. Um melhor desempenho de um modelo de classificação depende de uma representação que possa extrair informações valiosas sobre os textos. O problema discutido nesta pesquisa de doutorado é como melhorar as representações de texto incorporando semântica para melhorar a eficácia dos modelos de classificação de texto. Visando não perder informações locais dos textos, uma forma de representá-los é por meio de fluxos, sequências de informações coletadas deles. Esta tese propõe uma abordagem que combina várias técnicas de representação de textos: a representação por fluxos, o poder dos *word embeddings* associado a léxicos por meio de semelhança semântica e a extração de *features* inspiradas na área de análise de áudio. A abordagem divide o texto em frases e calcula uma distância de similaridade semântica para um léxico em um *embedding space*. A sequência de distâncias compõe o fluxo do texto. Em seguida, o método realiza a extração de vinte e cinco *features* inspiradas na análise de áudio (*Audio-Like Features*). A adaptação de *features* da análise de áudio vem de uma semelhança entre um fluxo de texto e sinal digital, além do relacionamento existente entre texto, discurso falado e áudio. A avaliação experimental realizada compreende cinco tarefas de classificação de textos: Detecção de *Fake News* em Inglês e Português; Colunas de jornal versus notícias; Polaridade de Sentimentos envolvendo Resenhas de Filmes em Inglês e Resenhas de Livros em Português. Os experimentos compreenderam seis *datasets* e seis léxicos envolvendo os idiomas inglês e português. A eficácia da abordagem é comparada a fortes *baselines* que incorporam semântica na representação de texto: *Paragraph Vector* e *BERT*. O objetivo dos experimentos foi investigar se a abordagem proposta poderia competir com a eficácia dos métodos *baseline* ou melhorar sua eficácia quando associada a eles. A avaliação experimental demonstra que o método pode aumentar a eficácia da classificação dos métodos *baseline* em quatro dos cinco cenários. Na tarefa Detecção de *Fake News* em Português, a abordagem superou os *baselines* e obteve a melhor eficácia (PR-AUC = 0,98). As *features* propostas alcançaram melhores resultados em modelos de *Shallow Learning* comparado a *Deep Learning* em três tarefas. Nenhum subconjunto de *features* apareceu entre os mais impactantes em todas as tarefas de classificação, destacando a importância de todas as vinte e cinco *features*.

Abstract

Text classification is one of the mainly investigated challenges in Natural Language Processing research. The higher performance of a classification model depends on a representation that can extract valuable information about the texts. The problem discussed in this doctoral research is how to enhance text representations by incorporating semantics to improve the efficacy of text classification models. Aiming not to lose crucial local text information, a way to represent texts is through flows, sequences of information collected from texts. This thesis proposes an approach that combines various techniques to represent texts: the representation by flows, the power of the word embeddings text representation associated with lexicon information via semantic similarity distances, and the extraction of features inspired by well-established audio analysis features. The approach splits the text into sentences and calculates a semantic similarity metric to a lexicon on an embedding vector space. The sequence of semantic similarity metrics composes the text flow. Then, the method performs the twenty-five audio analysis features inspired (called Audio-Like Features) extraction. The features adaptation from audio analysis comes from a similitude between a text flow and a digital signal, in addition to the existing relationship between text, speech, and audio. The conducted experimental evaluation comprises five text classification tasks: Fake News Detection in English and Portuguese; Newspaper Columns versus News; Sentiment Polarity involving Movie Reviews in English and Book Reviews in Portuguese. The experiments comprised six datasets and six lexicons involving the English and Portuguese languages. The approach efficacy is compared to baselines that embed semantics in text representation: the strong Paragraph Vector and the BERT. The objective of the experiments was to investigate if the proposed approach could compete with the baselines methods efficacy or improve their effectiveness when associated with them. The experimental evaluation demonstrates that the method can enhance the baseline methods classification efficacy in four of the five scenarios. In the Fake News Detection in Portuguese task, the approach surpassed the baselines and obtained the best effectiveness (PR-AUC = 0.98). The proposed features achieved better results on shallow learning models than deep learning in three tasks. None subset of features appeared among the most impacting ones in all classification tasks, highlighting the importance of all the twenty-five features.

Acknowledgements

First of all, I want to thank God for allowing my course till the conclusion of this research, granting me the health, inspiration, and determination necessary for this accomplishment.

I especially thank my mother, Elza, my biggest inspiration and partner of all “adventures”, for all the support given and for always emphasizing the value of dedication in everything we propose to do. I also thank Eveline, my sister, and Isadora, niece, who complement this amazing family.

A special thanks to my supervisor, Cláudio Campelo, for all the patience, trust, partnership, and dedication showed to me and this research. Thank you for driving this guidance precisely the way I needed it.

I would also like to thank my colleagues at the Lacina laboratory for all the discussions, opinions, and contributions that enhanced this research.

I thank the Instituto Federal da Paraíba - IFPB, which provided my exclusive dedication to this work.

I would also like to thank every member of the Programa de Pós-Graduação em Ciência da Computação for all the necessary support.

Finally, I would like to thank all my family and friends who gave me any form of support during this journey.

Contents

List of Symbols

ALF - *Audio-Like Feature*

ANN - *Artificial Neural Network*

BERT - *Bidirectional Encoder Representations from Transformers*

Bi-GRUs - *Bidirectional Gated Recurrent Units*

Bi-LSTM - *Bidirectional Long Short Term Memory*

BoW - *Bag-of-Words*

CBOW - *Continuous Bag-of-Words*

CNN - *Convolutional Neural Network*

CRF - *Conditional Random Fields*

D2V - *Paragraph Vector*

DL - *Deep Learning*

ELMo - *Embeddings from Language Models*

FFN - *Feed-Forward Network*

GLoVe - *Global Vectors for Word Representation*

GPT-2 - *Generative Pre-trained Transformer 2*

LFSP - *Last Frame Sentence Padding*

LSTM - *Long Short Term Memory*

NLP - *Natural Language Processing*

PV-DBOW - *Paragraph Vector - Distributed Bag of Words*

PV-DM - *Paragraph Vector - Distributed Memory*

RF - *Random Forest*

RNN - *Recurrent Neural Network*

SL - *Shallow Learning*

SS - *Semantic Similarity*

STFT - *Short-Time Fourier Transform*

SVM - *Support Vector Machine*

W2V - *Word2Vec*

WMD - *Word Mover's Distance*

XGBoost - *Extreme Gradient Boosting*

List of Figures

List of Tables

Chapter 1

Introduction

The constant growth of the amount and variety of digital text documents leads to the necessity of a consistent evolution in text mining techniques to continue promoting quality knowledge discovery [32; 2; 1]. Fake news detection, spam filtering, scientific articles organization, and sentiment analysis are examples in the vast field of text mining applications [48; 96; 52]. Frequently, text mining applications are structured as text classification tasks, aiming at creating a classification model capable of designating known class labels to text documents [102].

Text classification is one of the most discussed and studied challenges in Natural Language Processing (NLP) research. This kind of task involves creating a general classification model supported by previously labeled texts. Then, the created model is used to predict the class of unlabeled texts. Building a classification model requires a structured form of representing the texts. The superior performance of a classification model depends on a representation that can extract valuable common information about the texts [1; 38; 53].

A prevalent text representation model is the Bag-of-Words (BoW). This representation relies on the occurrence of words (from a known vocabulary) present in a text. Each text representation consists of a vector formed by a measure of each known word's presence [41]. The BoW representation (in its pure form) does neither express words order nor text syntax [102]. Despite being a simple representation, BoW is a model that achieves good results in several tasks, even challenging to surpass in different scenarios.

Nevertheless, some scenarios may demand a semantically more elaborate text representation to enhance the classification model performance since semantics is a powerful tool to recognize different contexts even when a similar vocabulary is used [2]. Word embeddings are a widespread representation that embeds semantics, in which vectors of various dimensions can express context information of words by adding a dependency between the words to the representation: the closer they are in a given context, the more they depend on each other [41]. Word2Vec [72], Paragraph Vector [60], Glove [82] and FastText [18] are examples of a particular and popular type of word embeddings, called static word embeddings. These models generate context-independent embeddings by representing each word (or a longer piece of text, e.g., in the "Paragraph Vector" approach) through a single vector, regardless of the context in which it occurs. Thus, static word embeddings are not able to represent polysemy.

Recently, contextualized word representations have been proposed, which are text representation models that consider the context in which the word occurs to represent it. Therefore, the same word in different contexts will have different representations [27]. As examples of contextualized word representations, it is possible to list: ELMo [83], BERT [28], and GPT-2 [88].

Another way of embedding semantics on text representation is recurring to lexicons. In the domain of NLP, lexicons are sets of terms or expressions that reflect a specific semantic context of a language. For example, the presence of terms from an argumentative lexicon in a piece of text may indicate that the author wanted to convey an argumentative tone [10].

Recent researches have been associating the power of the word embeddings representation with the additional information that lexicons promote to achieve a more accurate text representation model [10; 110; 15; 51; 36]. The widely used pre-trained word embeddings contain semantics for a general context. Associating these pre-trained word embeddings to lexicon information can give a more precise representation under a particular context. A technique to perform this association is based on semantic similarity. It consists in calculating semantic distances between the text terms and the lexical terms (using, for example, Manhattan Distance, Shortest Path Distance, Cosine Distance, or Word Mover's Distance - WMD [57]). The smaller the distance, the semantically closer is the text to the lexicon.

A text can be represented by applying such a semantic similarity technique at different levels of granularity. A possible way is to compute each text's word or sentence similarity and then calculate a summary metric (as an average, median or maximum) to represent the whole text [51]. As Aker et al. [4] discussed, this type of representation can lead to the loss of important information, especially for long texts. Different kinds of texts can present singularities in the semantic context of specific text parts that could be decisive in improving the classification task efficacy, and summary metrics probably would ignore them. Another form of representation consists in dividing the text into smaller units and computing the semantic similarity for each unit. In this case, the text units' semantic similarity sequence constitutes the text representation. We define this representation as *Text Flow*, following Mao and Lebanon's [69] definition of flow: a sequence of information collected from the words, sentences, or paragraphs of the text.

As Mao and Lebanon's [69], some other studies [107; 37] represent texts as flows and use the flows to perform classification tasks. However, their flows are not based on semantic similarity as in this thesis. As Filatova [33] and Seo and Jeon [97], other works obtain good results by extracting relevant features from the flows before performing the classification task.

Despite all the efforts mentioned above, representing natural language texts by incorporating semantics aiming to improve even more classification models efficacy is an open challenge for the NLP research community.

In the light of the prior considerations, the problem discussed in this doctoral research is how to enhance text representations by incorporating semantics to improve the efficacy of text classification models.

1.1 Objectives and Research Questions

The main objective of this doctoral research is to propose a method of representing texts by flows of information that incorporate semantics and, then, extracting features from the flows to enhance text classification efficacy.

Thus, to achieve the main objective, three specific objectives were established:

- To define a method of structuring texts as flows of information.
- To define a technique of incorporating semantics to the flow representation.

- To propose an innovative approach to extract features from the flow to improve text classification efficacy.

Given the objectives presented above, the research questions (RQ) of this doctoral study are:

- RQ1: Is a classifier model using the proposed approach features competitive to strong baseline methods in terms of efficacy?
- RQ2: Can the association of the proposed method with strong baseline methods improve the baselines' efficacy on text classification tasks?
- RQ3: Is there a subset of features from the proposed approach that could perform as well as or better than the entire set of features in all considered classification tasks?
- RQ4: The method achieves better efficacy when used to feed shallow or deep learning classification algorithms?

1.2 Overview of the Proposed Approach

The proposed approach represents texts by flows that incorporate lexicon information and then extracts features inspired by audio analysis from the flows. Singularly, it combines the text representation by flows, the power of the word embeddings text representation associated with lexicon information via semantic similarity distances, and the extraction of features inspired by well-established audio analysis features.

First, to represent the text as a flow, the method split the text into sentences. In this way, the approach does not lose crucial local information for differentiating two kinds of texts.

A WMD from each sentence to a lexicon on an embedding vector space is calculated to compose the text flow. Therefore, besides incorporating the specific context information from the lexicon into the representation, the approach avoids depending only on finding (and counting) known vocabulary terms.

The text flow can be approximated to a digital signal [63] if we consider each sentence as a point in "time" and each sentence WMD as the "signal" (flow) amplitude (Figure 1.1).

The NLP area comprises not only research on written language (texts) but also research on spoken language – or speech. Some examples of research involving speech are Automatic Speech Recognition (ASR) and Speech-to-Text (STT). The knowledge resulting from speech analysis research can be applied to a sort of modern applications, such as Personal Digital Assistance, like Siri, and Alexa [112].

Speech is a digital signal, specifically an audio signal, and, therefore, can be investigated by audio analysis [87; 86]. Considering the text flow as an approximation of a digital signal and this linkage of text-speech-audio, we decided to analyse the text flows inspired by the audio analysis, extracting features adapted from deep-seated audio analysis features.

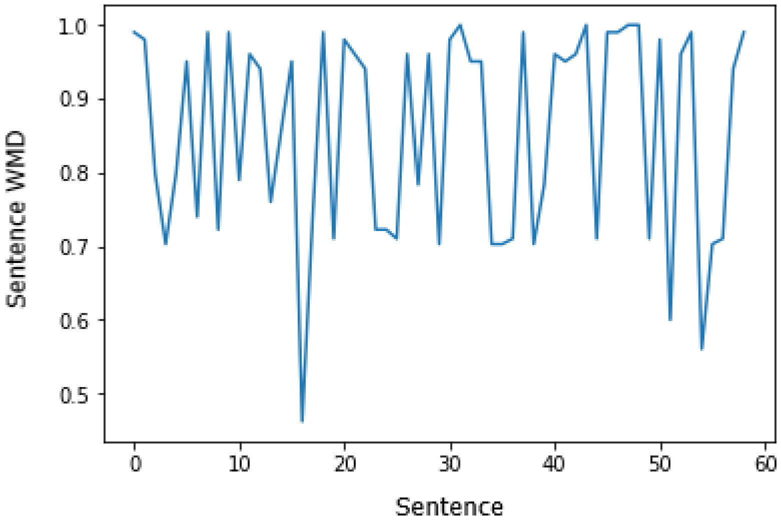

Figure 1.1: Text Flow graph representation

Before extracting the features, the method fragments the text flow into smaller parts called frames (as in audio analysis). The approach implements twenty-five features, sixteen extracted from frames and nine from the entire flow. The features, called Audio-Like Features (ALFs), are Energy, Median-Crossing Rate, Energy Entropy, Linear Prediction Median-Crossing Ratio, Text "Waveform" Minimum, Text "Waveform" Maximum, Text "Waveform" Diff, Volume, High WMD Segments Mean, High WMD Segments Standard Deviation, High WMD Segments Median, Low WMD Segments Mean, Low WMD Segments Standard Deviation, Low WMD Segments Median, Area of High WMD Segments, Log Attack Position, Spectral Flux, Spectral Entropy, Spectral Energy Ratio, Spectral Flatness, Spectral Crest Factor, Spectral Skewness, Spectral Kurtosis, Pitch, Jitter.

The audio analysis-inspired extracted features are much more sophisticated than only a summarization manner of representing texts, bringing valuable information to the representation.

The conducted experimental evaluation comprises five text classification tasks: 1) Fake News Detection in English, 2) Fake News Detection in Portuguese, 3) Newspaper Columns versus News Classification, 4) Movie Reviews Sentiment Polarity Classification in English, and 5) Book Reviews Sentiment Polarity Classification in Portuguese.

The method efficacy is compared to two baselines that embed semantics in text representation: the strong Paragraph Vector (D2V) [60] - a static word embedding - and BERT [28] - a contextualized word representation. Besides the classification employing models created using the ALFs, D2V, and BERT features separately, a classification involving models obtained by combining the ALFs and D2V features (D2V+ALF) and the ALFs and BERT features (BERT+ALF) was performed. These experiments verify if the approach can compete with the D2V and BERT methods efficacy or improve the baselines efficacy when associated with them.

The datasets used in the tasks were:

- A subset of All The News Dataset[1] to represent legitimate news in English and the fake news dataset compiled by Torabi ASR and Taboada [11].
- The legitimate and fake news in Portuguese dataset made available by Jeronimo et al. [51].
- The newspaper columns dataset, firstly presented in our work [105].
- The IMDB [67] movie reviews in English dataset.
- The Skoob book reviews in Portuguese dataset[2].

The lexicons employed in the tasks were:

- A combination of three English lexicons: Recasens et al. [91] bias-inducing lexicons, Wilson et al. [109] and Deng et al.[24] sentiment polarity lexicons.

[1] https://www.kaggle.com/snapcrack/all-the-news

[2] Available on https://gdarruda.github.io/2019/07/27/corpus-skoob.html

- Bing Liu's sentiment polarity lexicon in English [48].
- Reli-Lex, a sentiment polarity lexicon in Portuguese [35].
- Amorim et al. [8] subjectivity lexicons in Portuguese.

The experimental evaluation demonstrates that the method is capable of enhancing the strong baseline methods text classification efficacy in all scenarios, except the Movie Reviews Sentiment Polarity Classification in English. In the Fake News Detection in Portuguese scenario, the approach surpassed the baselines and obtained the best efficacy (PR-AUC = 0.98). The ALFs alone obtained better results on shallow learning models than deep learning in three tasks. In some scenarios, the features extracted from a unique frame reached the same results as all ALFs. None subset of features appeared among the most impacting ones in all classification tasks.

1.3 Bibliographical Contributions

The bibliographical contributions resulting from this research are:

- An article contemplating a preliminary version of the proposed approach. Published in the proceedings of the 12th edition of the Language Resources and Evaluation Conference - LREC 2020 [105].
- An article comprising an evaluation of the impact of combining subjectivity and sentiment lexicons on text classification in Brazilian Portuguese using the preliminary version of the proposed method. Submitted to the international journal Language Resources Evaluation Special Issue: Computational Approaches to Portuguese (under review; submission date: June 15th, 2021).
- An article presenting the current version of the approach and the most recent experimental results. To be submitted to the international journal Natural Language Engineering.

1.4 Thesis Structure

The remainder of this document is structured as follows:

- Chapter 2 describes the necessary theoretical foundations for a better understanding of this work.
- Chapter 3 presents a selection of studies related to the work developed in this thesis.
- Chapter 4 introduces the proposed approach in detail.
- Chapter 5 presents the experimental evaluation and discussion of the obtained results.
- Chapter 6 presents the conclusions and future work.

Chapter 2

Background

This chapter summarizes the background information needed to understand this research. Initially, the chapter presents an overview of the Natural Language Processing area and approaches some concepts related to the area explored in this work. Afterwards, concepts in the area of audio analysis important for understanding the development of this research are presented.

2.1 Natural Language Processing

Natural Language Processing (NLP) investigates how to use computers to process and understand human (i.e., natural) language. NLP is an interdisciplinary area involving computational linguistics, computer science, cognitive science, and artificial intelligence. The aim of NLP is to model the cognitive mechanisms used by humans to understand and produce their language. In practice, NLP focuses on developing new models and applications that facilitate the interaction between machine and human languages. Examples of typical NLP applications include text classification, spoken language understanding, lexical analysis, information retrieval, translation, natural language summarization, sentiment analysis, among others [26].

The NLP problems are very challenging precisely because human language is ambiguous. Furthermore, it is constantly changing and evolving. Computers do not easily understand the rules governing the passage of information through natural language [41]. These rules can often be abstract and high-level, such as the use of sarcasm, and other times quite concrete and low-level like the use of the letter' s' to denote plural. This imprecision makes it difficult for computers to handle natural language.

Deng and Liu [26] describe the three phases of NLP evolution, since the first studies that began in the 50s, joining the areas of Artificial Intelligence and Linguistics.

The first phase lasted until the mid-1980s was characterized by building intelligent systems based on manually defined rules. These rules intended to incorporate some level of linguistic information into these systems. The second phase of the NLP evolution uses datasets and learning models of machine and statistical modeling to construct intelligent systems. The approaches developed in this phase are called empirical because they are fundamentally based on data (e.g., examples) rather than predefined rules. The empirical approaches absolutely dominated the studies related to NLP until around 2010. Today, this research strategy coexists with the deep learning approaches that compound the third phase.

Deng and Liu (2018) name the third phase as Deep Learning, and it represents the most current state of development of the NLP. In machine learning models (or shallow learning models), standard features used in prediction models are extracted by humans, which is a costly and time-consuming process. Models based deep learning allow going beyond these limitations, providing, through multilayer neural networks, models with a great generalization power, permitting the representation of highly complex functions [42]. The main current advances in processing languages are linked to deep learning models. However, despite the revolution brought about by these models within the NLP, the limitations of these models still pose significant challenges. Among the main ones is the lack of interpretability of these models, generating the so-called "black box" model, making it difficult to interpret and explain them. Another significant limitation is the need for a large amount of data that these models tend to use for training. The development brought by the third stage of the NLP opened new paths for the area, particularly in applications that seek to study semantic relationships in documents. A significant advance in this regard was the use of word embeddings for extracting semantic representations of words in a document [72; 28].

The rest of this section will present concepts related to NLP necessary to understand the work described in this thesis.

2.1.1 Word Embeddings

One of the main challenges in NLP is the way to represent natural language. Models based on vector space are among the most popular forms of representation. Each document is represented by a vector whose dimensions correspond to features extracted from the text.

A form of representation that stands out in this context is the distributed representation, such as word embeddings, in which vectors of various dimensions can express information from the context of the words by adding a dependency between the words to the representation: the closer in a context, more dependent they are. In other words, word embedding is a text representation where words with the same meaning have a similar representation [41].

The most recent techniques use neural networks to learn the word embeddings through models trained from large masses of raw data. The word representations are given by dense vectors with low dimensionality in a predefined vector space [26]. The representation can capture syntactic and semantic information of each word. Goldberg [41] emphasizes that the use of word embeddings brings benefits regarding the power of generalization: if some features can provide similar information, it is essential to provide a representation capable of capturing these similarities. If a sufficiently large database is used, the so-called neural embeddings can capture semantics features of terms, even allowing the calculation of semantic similarity between them. For example, operations on the vectors "king - man + woman" would return a vector next to that represented by "queen".

Examples of neural embeddings are Word2Vec (W2V) [72], GloVe [82] and FastText [18]. Word2Vec relies on two neural network architectures to produce a distributed representation of words:

- Continuous bag-of-words (CBOW): in this architecture, the model predicts the most likely word in a given context. Hence, words with equal probability of occurring in the text are considered similar and take close places in the vector space.

- Skip-gram: this architecture is similar to CBOW, but it works in reverse mode. In other words, given a word, the model predicts its context.

The D2V model is an extension of the W2V and learns fixed-length feature representations from variable-length portions of texts, such as sentences, paragraphs, and documents [60]. Also, the model relies on two network architectures:

- Distributed Memory version of Paragraph Vector (PV-DM): the model assigns a paragraph vector sentence while sharing word vectors among all sentences. Then it either averages or concatenates the paragraph vector and words vector to get the final sentence representation. It is the extension of the CBOW architecture, predicting the next sentence given a set of sentences.

- Distributed Bag of Words version of Paragraph Vector (PV-DBOW): the skip-gram extension, the model samples random words from the sentence and predicts a sentence from a classification task.

Unlike W2V, which is based on predictive architectures considering only the local context, the *GloVe* model is trained on a global matrix of co-occurrence generated from a collection of texts. This matrix is decomposed to form a denser and more expressive vector representation. Both W2V and GloVe have the disadvantage of not representing unknown words. In turn, FastText was proposed to solve this difficulty. Based on W2V architectures, FastText breaks words into sub-words (n-grams) and feeds them into the neural network. Then, each word is represented by the sum of the vectors of its n-grams. Therefore, a new word can be represented since there is a high probability that the trained model already knows its n-grams.

The inability to represent polysemy is a weakness presented by W2V, GloVe, and FastText models (known as *static word embeddings*). They portray each word by a single word embedding, regardless of the context. Thus, the contextualized word representations models were proposed to solve this difficulty. These models consider the context in which the word occurs to represent it. Therefore, the same word in different contexts will have different representations. As examples of *contextualized word representations*, it is possible to list: ELMo [83], BERT [28], and GPT-2 [88].

ELMo, BERT, and GPT-2 are language models created from deep learning models. Their internal word representations are a function of the entire sentence. ELMo creates contextualized representations of each token, concatenating the internal states of a bi-directional two-layer Long Short-Term Memory (Bi-LSTM) network trained on a bi-directional language model. In turn, BERT and GPT-2 are language models based on bi-directional and uni-directional transformers, respectively [106]. Each transformer layer creates a contextualized representation of each token by participating in different parts of the input sentence.

In this work, the W2V is used to create the vector representation of the textual terms, using the Skip-Gram algorithm to build the embeddings. Then, the text sentences and lexicons are represented in this vector space, and a semantic similarity distance between them is calculated to generate the flows. In the approach evaluation, D2V and BERT models were used as strong baselines as means of efficacy comparison.

2.1.2 Lexicon

In linguistics, a lexicon is the vocabulary of a person, language, or branch of knowledge. It is a collection of information about the words of a language concerning the lexical categories to which they belong, including meanings, use, form, and relationships with other words.

In NLP, a lexicon is a system that contains semantic or grammatical information about individual words or expressions (lexical entries) [43]. It is habitually structured as a collection of lexical entries, including additional information about their roles.

The most frequent kind of lexicon in NLP is the sentiment polarity lexicons. These lexicons usually have two or three dimensions representing sentiment polarities: negative, positive, and neutral (not present in all lexicons). Each dimension has a set of entries related to that polarity in the underlying language. Examples of sentiment polarity lexicons are Sentiwordnet [13], Hu and Liu's [48], and Deng et al. [24] lexicons in English and Relilex[35] in Portuguese. Despite a vast amount of sentiment polarity lexicons, there are some lexicons from other categories, for example, the Recasens et al. bias-inducing lexicon [91] in English, and the Amorim et al. subjectivity lexicon [8] in Portuguese.

This work uses lexicons to add information to the text representation by calculating semantic similarity distances between text sentences and the lexicons on the same embedding space.

2.1.3 Semantic Textual Similarity

Semantic Textual Similarity - or just *Semantic Similarity* (SS), for simplicity - is the degree of semantic equivalence between two words, sentences, paragraph and document [3]. SS is one of the fundamental tasks in NLP, and the existence of good models capable of calculating

SS is crucial for many applications in NLP, such as information retrieval, classification, and text summarization [90]. For example, search engines need to compute SS between two short texts to model a document's relevance to a query. Likewise, question-and-answer sites need to compute SS to conclude if a question has already been asked.

One of the most recent and used approaches for calculating SS is first embedding the two words, sentences, or documents into a vectorial space (by using W2V, for example) and then computing an appropriate metric between the embedding vectors [111]. The Cosine Similarity is an often-used metric to calculate SS [90]. It is a metric that determines how similar two words, sentences, or documents are, regardless of their sizes. Mathematically, as Figure 2.1 illustrates, it is the measure of the cosine of the angle Θ between two projected vectors (V1 and V2) in a multidimensional space.

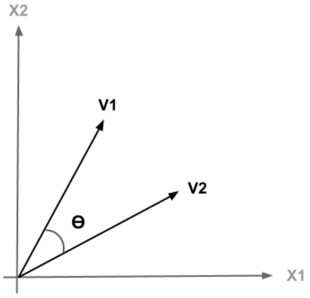

Figure 2.1: Illustration of the angle between two vectors projected in a two-dimensional space.

An alternative very usual metric to calculate SS is *Word Mover's Distance* (WMD) [57]. Presented by Kusner et al., the WMD is a distance function that measures the (dis)similarity between two text documents. It is defined as the smallest distance that the word embeddings of a document need to "travel" until reaching the word embeddings of another document. Kusner et al. claim that distances between vectors of word embeddings are semantically significant. Using this property of word embeddings, WMD represents text documents as a cloud of weighted points in a vector space - the set of word embeddings in the document. The distance between two documents is the minimum cumulative distance that words in one document must "travel" to meet the other text's cloud points.

Figure 2.2 illustrates the approach behind calculating WMD. It depicts two sentences from different documents in the side boxes: (1) "*Obama speaks to the media in Illinois.*" in blue and (2) "*The President greets the press in Chicago.*" in black. The sentences, despite not having words in common (except the *stop words*[1] "*the*" and "*in*"), have practically the same meaning. In the center of the figure, the words are illustrated, represented in the vector space. The blue dots form the sentence cloud (1), and the black ones form the sentence cloud (2). The WMD between sentences (1) and (2) will be the sum of the minimum distances that each word in the sentence (1) must "travel" to find precisely the corresponding point in the document cloud to which sentence (2) belongs.

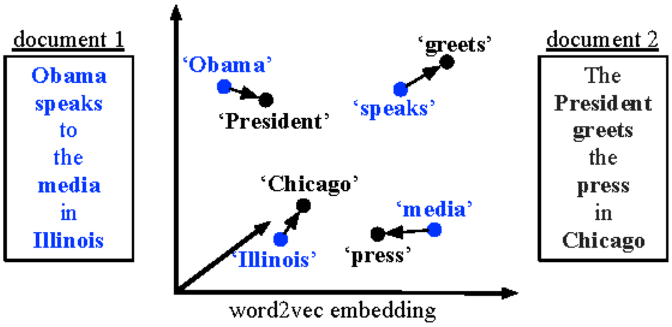

Figure 2.2: Illustration of *Word Mover's Distance*. (Source: Kusner et al. [57])

This research uses the WMD to calculate the SS between the text sentences and the lexicons dimensions word embeddings to produce the text flows.

2.1.4 Text Preprocessing

Research in NLP has to encode unstructured text data into structured data, as word embeddings, prior to performing the tasks. Nevertheless, before the encoding phase, it is necessary to perform text preprocessing, a method to remove text data noise as punctuation and different case.

Text preprocessing is generally performed in three steps: segmentation or tokenization, normalization, and noise removal. The segmentation step concerns splitting texts into smaller pieces - for example, document into paragraphs, paragraphs into sentences, sentences into words. Normalization intends to set all text on a flat playing field, e.g., converting all char-

[1] Widespread words in the language that, in general, do not add much sense to the text. They are commonly removed from the texts before the tasks in NLP are performed.

acters to lowercase and removing the stop-words. Stop-words are words that have a limited contribution to overall meaning, given that they are generally the most common words in a language (e.g. 'the', 'and', 'an' in English). Noise removal cleans up the text, e.g., remove extra whitespaces, URLs, and HTML tags.

Given the indisputable importance of text preprocessing, this research performs this process before generating the text sentences' word embeddings.

2.1.5 Text Classification

Text classification is one of the most essential and common tasks in NLP. The purpose of text classification is to determine the category of a given document, considering a set of predefined categories, principally concerned about distinguishing relevant documents from extensive collections of raw text [92; 20].

After the text preprocessing phase, the text classification task's next step is the text representation and feature extraction (phase regarded by this thesis). Then the classification model construction is the final step [20]. This section will overview the text classification tasks performed and the machine learning (shallow and deep) models used in the proposed method evaluation.

Fake News Detection

Fake news is a publication with journalistic characteristics (i.e., news in article format, containing title, author, and body) without a factual basis, aiming to deceive the audience. This news is usually published by communication vehicles (newspapers or news blogs) with the conscious intent of misinforming or deceiving [6; 100; 54].

Fake News Detection is a text classification task that intends to differentiate fake from legitimate news. Given the massive dissemination of fake news provided by social media and messaging applications and the significant consequences this dissemination can lead to, the need to detect fake news is increasingly present[2].

[2] https://www.acritica.com/channels/coronavirus/news/belief-in-fake-news-potentiates-spread-and-pandemic-problems

Newspaper Columns versus News Classification

A *newspaper column* is a recurring feature written by the same author in a newspaper. It is an opinionated text frequently defined by the voice and personality of the writer, in opposition to objective news that reports facts. The *Newspaper Columns versus News Classification* aims to distinguish between these two kinds of texts: one that expresses an opinion and the other that expresses facts. It is an essential task because, nowadays, there is an increase in people's confusion between fact and opinion. For example, some people cannot distinguish between what is news and fact published by the journalistic vehicle and the political agenda of the vehicle itself or the journalists who work in it.

Such confusion may derive from the consumption of information through social media, a strategy adopted by many newspapers, in which journalistic information is consumed through posts, most of the time, without distinction between news and column. For instance, the existence of such ambiguity in the contemporary world with an intense political debate is worrying. It could reinforce the existence of 'bubbles' where people only provide feedback for their own beliefs, using others' opinions as facts[3].

Sentiment Analysis

Sentiment Analysis is a research area that focuses on the classification of sentiments in texts. To this end, texts can be labeled into several categories, commonly positive and negative polarities [22].

Opinionated information is widely available online and plays a vital role in evaluating whether a product or service is pleasing its consumers or not. In this context, Sentiment Analysis of product or service reviews is a commonly exploited field since it focuses on the classification of sentiments or opinions expressed in human-generated texts [10]. For example, it can enhance the capabilities of customer relationship management and recommendation systems, allowing to find out which features customers are pleased about or exclude from the recommendations items that have received very negative feedback [22].

[3] https://jornal.usp.br/radio-usp/distinguir-fato-de-opiniao-e-imprescindivel-para-a-democracia/

Classification Algorithms

A wide variety of algorithms for classification is available, each one with its strengths and weaknesses. There is no such learning algorithm that operates properly with all the classification problems [79]. Examples of classification algorithms are artificial neural networks, decision trees, k-nearest neighbors, Naıve Bayes classifiers.

This subsection gives an overview of the classification algorithms used in the proposed approach evaluation, namely Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and two Artificial Neural Network algorithms (ANN): Feed-Forward Network (FFN) and Bidirectional Long Short-Term Memory Network (BiLSTM).

The Random Forest classifier is an ensemble machine learning method that considers multiple learning algorithms while generating a prediction result. It consists of a collection of tree-structured classifiers, and each tree casts a unit vote for the most popular class to the input [19]. The individual trees are built via bagging (i.e., aggregation of bootstraps - multiple train datasets created via sampling of records with replacement) and split using fewer features. The resulting diverse forest of uncorrelated trees exhibits reduced variance; therefore, it is more robust towards change in data and carries its prediction accuracy to new data.

The XGBoost classifier is also known as the regularized boosting technique. It is an advanced implementation of the gradient boosting algorithm but implements regularization to reduce overfitting. Boosting is an ensemble technique that combines a set of weak learners and delivers improved prediction accuracy, a strong learner. Each weak learner results from a base learning algorithm applied to a (different) distribution. In an iterative process, each time this base learning algorithm is applied, it generates a weak prediction rule. After many iterations, the boosting algorithm combines these weak rules into a single strong prediction rule. In the gradient boosting classifier, each new weak model gradually minimizes the loss function of the whole ensemble using the gradient descent method. This algorithm aims to construct new base learners that can be maximally correlated with the negative gradient of the loss function associated with the whole ensemble.

An Artificial Neural Network (ANN) is a computational model inspired by the biological neural networks in the human brain. One of the most popular ANN algorithms is backpropagation that performs learning on a feed-forward ANN [44]. During the learning phase, the network learns by adjusting the weights iteratively for predicting the correct class label of the given input features.

A Feed-Forward Network consists of: An entrance layer having input neurons that provide information from the outside world to the network; One or more hidden layers having hidden neurons that perform computations and transfer information from the input to the output neurons; An output layer contains output neurons responsible for computations and transferring information from the network to the outside world. Typically, network design is a trial-and-error process and may affect the accuracy of the resulting trained network [95].

The BiLSTM is a kind of Recurrent Neural Network (RNN). An RNN is a particular ANN capable of modelling sequential data by having recurrent connections. It maintains a hidden state, a 'memory' of previous inputs, as each neuron represents an approximation function of all previous data. One weakness of the RNN is that it can suffer from the vanishing gradient problem. This problem occurs when the gradient becomes vanishingly tiny, effectively preventing changing the network's weight and even wholly stopping the network's training. Thus the RNN is ineffective for long-term dependencies [84].

A particular type of RNN called Long Short Term Memory (LSTM) [46] was invented to address the vanishing gradient problem. LSTM cells follow a more sophisticated mechanism introducing a complex cell that utilises 'forget' gates to selectively decide what to forget. This cell can be trained to ensure that the gradient of the objective function does not vanish [99].

However, an LSTM network can only perform forward passes on sequential data, modelling data dependencies only uni-directionally. A BiLSTM network is a way to overcome this limitation, combining two equal LSTMs, one in forwarding mode and the other in reverse mode.

Classification Models Interpretability

Some modern machine learning classification models act like "black box" models, and it becomes hard to understand how they make their decisions. Understanding how these models conclude their classification is essential, particularly on running critical systems.

In order to interpret classification models, a recurrent strategy is the feature importance analysis. Feature importance refers to techniques for assigning scores to input features of a predictive model that reveals the comparative importance of each feature when making a prediction. Inspecting the importance score provides understanding about a specific model and which features are the most or less important to the model. Feature importance analysis allows reducing dimensionality by excluding less important features to a model [56].

This research used the SHapley Additive exPlanations (SHAP) values [66] to perform feature importance on the approach evaluation. The SHAP values represent each feature's importance to the prediction of machine learning models. For instance, features that significantly impact the result of a classification model are considered more relevant and receive a higher SHAP value. The SHAP values analysis allows a more objective understanding of the classification model's decisions, generating insights into the problem discussed.

2.2 Audio Analysis

Audio Analysis is a well-established research area that studies and develops knowledge about audio content. The knowledge already produced has proven to be valuable in various tasks such as segmentation and classification for music recommendation [65; 45], classification of audio as speech or music [17], emotion analysis in songs [49] and sentiment extraction in streams of speech [55].

As well as any natural phenomenon to be studied digitally, sound waves must be represented as a discrete-time signal. In audio analysis, the original waves are sampled and quantized to generate a discrete signal, which vectors of real numbers can represent (Figure 2.3).

Frequently, to perform the analysis, the audio signal is fragmented into smaller parts, overlapping or not, called frames. This fragmentation is often necessary because the characteristics of a general audio signal, by nature, vary a lot (often quickly) throughout the signal [75; 7]. To better understand the reason for using this technique, consider audio that presents a conversation, and a gunshot occurs in the middle of it. If the average signal strength is calculated considering all the samples, the ones that present the shot will dominate the result. If only this metric is considered in an analysis, the conclusions could be skewed. On the other hand, some audio signals, like a pure speech signal, present a slowly varying nature. In this case, the signal is also processed in blocks (frames) over which the properties of the speech

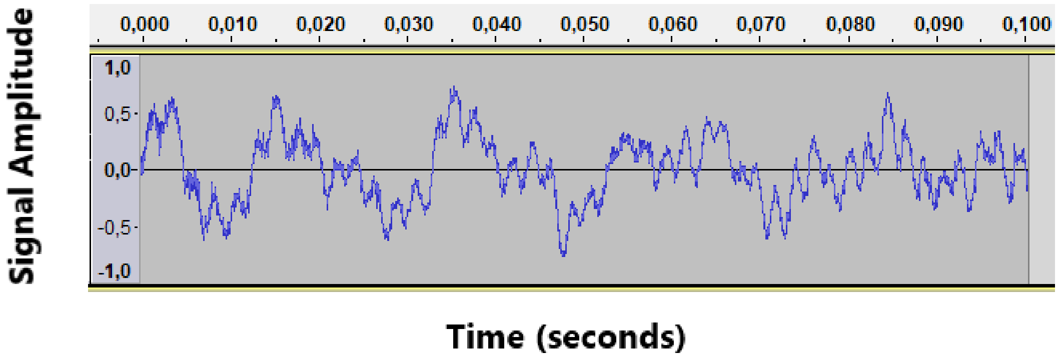

Figure 2.3: Representation of a sound signal in discrete time.

waveform can be assumed to remain relatively constant (signal stationarity) [87].Performing analysis using frames allows capturing more reliable information to the audio content [39].

Once the audio signal is split into frames, it is possible to extract features that produce a valuable representation of the signal, identifying essential characteristics and, therefore, useful for further analysis [85]. These features can be classified by type and by domain. Regarding the type, they can be called short-term or mid-term features. The short-term features are extracted from each frame individually, referring to the characteristics of that frame. The mid-term features are statistics such as mean, variance, standard deviation calculated from short-term features of consecutive frames that make up a mid-term segment or window [75].

Regarding the domain, the features can be originated , for example, from the time or frequency domains. The time-domain features are extracted directly from the representation of the audio signal from discrete-time samples. The frequency-domain features (also known as spectral-domain features) are extracted from the representation of the frequency distribution of the sound signal content (sound spectrum) [40]. It is worth transforming the audio signal from the time to the frequency domain to compute the spectral features. This transformation is usually fulfilled from the Short-Time Fourier Transform (STFT) or derived from an autoregression analysis [7].

The literature presents a wide variety of audio features from the time and frequency domains. The present research has evaluated deep-seated features presented in two audio analysis features reviews [75; 7] to choose viable and valuable features to the text analysis domain and incorporate them into the proposed method.

This approach presents twenty-five adapted features. Sixteen features are from the time domain audio analysis features: Energy, Zero-Crossing Rate (ZCR), Energy Entropy, Linear Prediction Zero-Crossing Ratio, Audio Waveform Minimum, Audio Waveform Maximum, Shimmer, Volume, Amplitude Descriptors (seven features) Log Attack Time. The remaining nine features (Spectral Flux, Spectral Entropy, Subband Energy Ratio, Spectral Flatness, Spectral Crest Factor, Spectral Skewness, Spectral Kurtosis, Pitch, Jitter) are from the frequency domain.

- Energy conveniently represents the amplitude variation over time [113]. It provides a way to distinguish voiced from unvoiced fragments because values for the unvoiced components are generally significantly smaller than those of the voiced components. It can also be used to distinguish audible sounds from silence, and the change in its pattern over time may reveal the rhythm and periodicity properties of sound. As shown by Zhang and Kuo [113], energy helps their method to perform a more than 90% accurate voiced and unvoiced speech classification.

- The Zero-Crossing Rate is a metric of signal noisiness. In other words, ZCR reflects the variation level of the sounds on the signal frame. For instance, it is used to distinguish between voiced and unvoiced signals because unvoiced speech components typically have much higher ZCR values than voiced ones. The unvoiced fragment presents much more variation than the voiced, which is more stable (even if it present a bigger Energy) [113]. To exemplify its efficacy, ZCR also helps Zhang and Kuo's method [113] performing a more than 90% accurate voiced and unvoiced speech classification.

- Energy Entropy can measure the "peakiness", or the abrupt changes in the signal energy level. For example, unvoiced sounds are flatter (less abrupt changes) and present higher entropy than voiced sounds. Several research efforts have experimented with the entropy of energy in the context of detecting the onset of abrupt sounds [40].

- Linear Prediction Zero-Crossing Ratio (LP-ZCR) is the ratio of the zero-crossing ratio of the frame waveform, and the zero-crossing ratio of the output of a linear prediction analysis filter [30]. The feature quantifies the degree of correlation in a signal. It helps distinguish between different audio types, such as voiced (higher correlated) and unvoiced speech (lower correlated). In their experiment, Maleh et al. [30] show that LP-ZCR helps to improve the baseline performance on 23.3% on a speech/music classification task.

- The Audio Waveform Minimum and Maximum features give a compact description of the shape of a waveform by computing the minimum and maximum amplitude samples within frames. Their purpose is to display and compare waveforms, been used as an efficient feature in environmental sound recognition, for instance [**?**; **?**].

- Shimmer computes the intraframe cycle-to-cycle variations of the waveform amplitude [**?**; **?**]. It has been heavily used to discriminate voiced and non-voiced regions of a music or, as shown by Farrús et al. [31], to improve speaker recognition by 17%.

- Volume is defined as the Root-Mean Square (RMS) of the waveform magnitude within a frame in audio analysis. It reveals the magnitude variation over time and improves silence detection and speech/music segmentation [64].

- Mitrovic et al. [74] described the Amplitude Descriptors (AD), a set of features capable of describing characteristics of the waveform, such as peaks and silence. Based on an adaptive threshold, initially, the features express the length of high and low amplitude sequences of samples and the area corresponding to the high amplitude sequences. The authors consider statistical properties of the initial features to build features that describe entire sample files. The features helps the baseline to achieve 91% of precision on discriminating animal sounds [74].

- Log Attack Time is the logarithm of the elapsed time from the beginning of a sound signal to its first local maximum, and characterizes the beginning of a sound. This feature has been essential to improve environment sounds classification [34].

- Spectral Flux quantifies abrupt changes in the spectral energy distribution over time. For example, signals with slowly varying or nearly constant spectral properties (e.g., noise) have low Spectral Flux. This feature plays an essential role on improving speech/music discrimination [**?**; **?**].

- The Spectral Entropy is computed similarly to the Energy Entropy, but now, the computation occurs in the spectral domain, that means, using the STFT output [73]. The discussion is also equivalent; Spectral Entropy measures abrupt changes in the Energy level of an audio signal on the spectral domain (flow STFT output). Misra et al. [73] show that, in Automatic Speech Recognition, the spectral entropy helps to improve the baselines results in 23.7%, mainly when the noise level grows up.

- The Subband Energy Ratio is usually defined as a measure of the normalized signal energy along with a predefined set of frequency subbands, describing the signal energy distribution of the spectrum. It has been used for audio segmentation and music analysis applications, and environmental sound recognition, helping to improve the baselines efficacy. [75; 7; 76].

- The Spectral Flatness measures uniformity in the frequency distribution of the power (squared) spectrum in audio analysis. It is computed as the ratio between the geometric and the arithmetic mean of a subband. Noise-like sounds have a higher flatness value, while tonal sounds have lower flatness values. This feature has been efficiently used on audio fingerprinting and music classification [7; 75].

- The Spectral Crest Factor measures the “peakiness” of a spectrum, being inversely proportional to the Spectral Flatness and also used to distinguish noise-like and tone-like sounds, being used in conjunction with Flatness on audio fingerprinting and music classification [7; 75].

- The Spectral Skewness measures the asymmetry of the spectral distribution around its mean value. If the Skewness is negative, there is more energy on the right side of the spectral distribution; if positive, more energy on the left side [81]. At the same time, the Spectral Kurtosis describes the flatness of the spectral distribution around its mean. A Kurtosis value lower than three describes a flatter spectral distribution, while a value bigger than three describes a peaker distribution.Both features are heavily applied on sound description tasks [81].

- Also known as Fundamental Frequency, the Pitch feature is defined as the first peak of the local normalized spectro-temporal autocorrelation function [23]. Autocorrelation is the similarity between observations considering a time lag between them, being useful for finding repeating patterns in the signal. Autocorrelation values range from -1 to 1. A negative value represents negative autocorrelation, and a positive value represents positive autocorrelation. It has been beneficial in various audio analysis tasks, including voicing level detection [23].

- Jitter is the cycle-to-cycle Pitch variations or the absolute mean difference between consecutive periods of an audio signal. Besides typically being applied to analyze pathological voices, it also present improvements around 17% on speaker recognition, as presented by Farrús et al [31].

2.3 Final Considerations

This chapter presented the definition of various topics used during this research that the reader may ignore. The background comprised concepts on the NLP area, like word embeddings, semantic textual similarity, lexicons, and text classification. Since our proposed approach is inspired by audio analysis, some concepts of this area were also addressed.

The next chapter discusses some works related to enhancing text representation, emphasizing research on flows and semantics.

Chapter 3

Related Work

This chapter presents the efforts of some previous studies concerning themes related to this research. Section 3.1 presents works that represent texts as flows to avoid the loss of relevant information on classification tasks. On the other hand, Section 3.2 presents studies that focus on enhancing text representation with semantics and performing text classification.

3.1 Representing Texts as Flows

A traditional way to represent texts is by gathering numerical information (e.g., semantic similarity) of parts of the text, like words, sentences, or paragraphs, and calculating a summary metric (as an average, median or maximum) to represent the whole text. In their work, Aker et al. [4] discussed that this type of representation could mislay critical information, especially for long texts. Different kinds of texts can present singularities in the semantics of distinct text parts that could be decisive in improving the classification efficacy, and summary metrics probably would miss them. An approach to avoid the loss of relevant information is representing a text as a flow.

In their study, Mao and Lebanon [69] use sentiment flows to represent texts. The authors propose a variation of the statistical modeling method Conditional Random Fields (CRF) [59] - the Isotonic Conditional Random Fields (ICRF) - which includes ordinal data to predict the sentiment of each sentence, called local sentiment. The sequence of predicted local sentiments forms the flow of sentiments in a text. They also find that flows of different texts will have different sizes, which would make it difficult to compare them or train models to use them. To solve this problem, they propose a normalization of text flows, using a smoothing function of sentence values, transforming the flow into a smoother transition curve. The experiments presented in Mao and Lebanon's work [69] show that ICRF obtains better results than *CRF*, Naive-Bayes, and Support Vector Machine (SVM) when predicting local feelings. They also show that using sentiments flows better predict the global sentiments of texts than using a BoW representation, awakening the power of flows.

On the other hand, Wachsmuth and Stein [107] present the representation of the discourse structure of a text as flows of rhetorical moves (communicative functions of argumentative text segments, generally linked to the argument's final objective).

Wachsmuth and Stein's work aims to capture the overall structure of an argumentative text by comparing its flow with a flow pattern extracted from a set of training texts. Therefore, it is necessary to unify the flows to extract a pattern. Two forms of unification are suggested: flows normalization to the same vector space and abstraction of variations between flows, mapping similar flows to a unique one. For normalization, a target text size is determined, and interpolation is used in the flows so that they all fit the chosen size. For abstraction, the authors suggest excluding three behaviors: similar rhetorical moves in sequence, leaving only one as a representative; cycles of two or more identical rhetorical moves (leaving only one cycle); and exclusion of minor rhetorical moves classes.

After unifying the flows, a clustering of training flows is performed, using similarity distances in vector space such as Manhattan Distance for the normalized flows and Minimum Edit Distance for abstracted flows. Then, to check which cluster a test flow is most similar, they use the same similarity distances between the test flow and the centroid of the training clusters.

Wachsmuth and Stein [107] perform global (hotel, movie, and product) reviews sentiment classification experiments. In some cases, flows better rank global feelings; in others, the baseline methods (BoW and sentiment frequency) get better marks; a combination between flows and baselines obtains better accuracy in other cases. In other words, there is no unique method that is more effective in all tested situations.

Similar to Mao and Lebanon's work [69], Wachsmuth and Stein [107] use the entire flow to perform classification but apply different information and way to generate the flows. In this study, we ground our text representation on flows; however, we generate the flows by semantic similarity distances (WMD) from lexicons and extract features to perform classification.

Alike our work, Filatova [33], Seo and Jeon [97], Lee et al. [61] and Pateria [80] extract features from flows to execute classification. Nevertheless, how they generate flows and extract features differs substantially from ours.

In her research, Filatova [33] studies the detection of sarcasm in large texts such as product reviews.The study shows that features that betray sarcasm in small texts social media messages (like emoticons, hashtags, and strong punctuation) do not have the same effect in more extensive texts. In large texts, she claims that understanding the context is very important, as sarcasm is detected in sentences that have a specific polarization in an opposite context. Therefore, the work proposes to model product evaluations as sentiment flows to capture the evaluation context and utilizes sentiment switchings between sentences (from negative to positive and vice versa) as features for sarcasm detection. Filatova [33] models the sentiment flows assigning sentiment labels to each sentence using the Stanford Sentiment Analysis tool [103].

The results of the performed experiments show that the method proposed by Filatova [33] obtains better results than the random baseline when both positive and negative evaluations are considered. However, the best results are presented when only positive evaluations are considered. Concerning negative evaluations, Filatova's approach presents results very close to those of the random baseline, which suggests that the behavior of sarcasm in these evaluations differs from the behavior in the positive evaluations.

Concerning the relevant document retrieval, Seo and Jeon [97] discuss that traditional BoW models use statistics to measure the relevance of documents. It is challenging for BoW to detect non-relevant documents containing multiple query terms at random or out of context. Facing this problem, the authors propose a representation of the texts through a relevance flow comprised of the relevance level of each sentence concerning the query.

In the solution proposed by Seo and Jeon [97], the relevance flows of the N best-ranked documents by the search engine baseline are generated, and the following features of these flows are extracted: average and relevance level variance; the ratio between the number of peaks and the number of sentences in the document; the position of the first peak; and mean and variance of peak positions in the document. The results show a statistically significant improvement when the documents are reranked using the features extracted from the flows.

Similar to Seo and Jeon [97], Lee et al. [61] propose to represent texts as a combination of a sentiment flow and a relevance flow to proceed with opinion retrieval. A score reflecting the relevance (concerning a query) and opinion (the frequency of a lexicon's opinion words presence) is computed for each sentence. As features, Lee et al. use the variance of sentences scores, the fraction of peaks, and the first peak position.

In his research, Pateria [80] addresses the problem of ambiguity on estimating sentiment associated with a specific aspect within a review (Aspect Based Sentiment Analysis). The author structures a review as a non-directed graph of connected aspects and sentiment nodes, a definition similar to an aspect-driven sentiment flow. He represents the reviews based on units, groups, and links. A unit is a set of terms related to an aspect obtained from parse relations. Completing the composition of a unit, the author defines terms related to the aspect terms and sentiment information. A group is a cluster of continuous units with a similar sentiment. A link is a connection between two units or groups or a unit and a group, represented by words that express contrast or addition. The proposed model first performs classification using a baseline set of features, including sentiment scores obtained from sentiment lexicons and neighbors' sentiment score features. Based on this, the probability estimations are obtained, which indicate ambiguities and preliminary information about neighbor sentiments. Then another classifier uses the local and non-local neighbor information (first-stage probabilities and textual terms) for prediction.

Another way to model flows for representing texts is using neural networks. By hypothesizing readers enjoy the emotional rhythm, Maharjan et al. [68] propose to model the flow of various emotions over a book to capture patterns that should represent the emotional arcs of the story. The objective is to enhance the prediction of a book's potential success. In addition, they show that using the book's entire content yields better results because if only a fragment is considered, it disregards significant emotional changes (aligned to Aker et al. [4] discussion).

Maharjan et al. [68] extract emotion vectors from different book chunks and feed them into a Recurrent Neural Network (RNN) to create the emotional flow. They divide the book into different chunks based on the number of sentences. Then, they create an emotion vector for each sentence by counting the presence of words of the ten different types of emotions of the NRC Emotion Lexicons [77]. After that, they condense these sentence emotion vectors into a chunk emotion vector by taking the average and standard deviation of sentence vectors in the chunk. Then, they aggregate the encoded sequences into a single book vector using an attention mechanism that summarizes the contextual emotion flow information from both directions. Finally, they predict success in books, applying a linear transformation that maps the book vector to the number of classes. Their approach shows better results than the baselines.

In their research, Ghanem et al. [37] obtained promising results on the fake news detection task by modeling the flow of affective information in fake news articles using a neural architecture. They hypothesize that fake news has a different distribution of affective information across the text from legitimate news. Therefore, modeling the flow of such information may help to discriminate fake from legitimate news. The proposed model (FakeFlow) learns this flow by combining topic and affective information extracted from text.

The FakeFlow model first divides an input document into N segments. Then it uses both word embeddings and other affective features (e.g., emotions) to catch the flow of emotions in the document. The model learns to pay attention to the flow of affective information throughout the document to detect whether it is fake or real. The topic-based sub-module uses a Convolutional Neural Network (CNN) to extract topic-based information from articles. This representation highlights essential words in which the topic information of the segment is summarized. The affective flow sub-module uses Bidirectional Gated Recurrent Units (Bi-GRUs) to model the flow of the affective information within the articles. The flows rely on affective lexicons term frequency representation weighted by the texts' length. The evaluation results indicate that the proposed method presents a statistically significant improvement over most baselines.

Both Maharjan et al. and Ghanem et al.'s studies resemble ours by representing texts as flows and, to generate these flows, they also divide texts into smaller parts and use lexicons. Although, adversely from our research, they use the entire flow to perform classification and use the term frequency of lexicons to create the flows. This way, they only capture the lexicon's information if the exact term appears on text. In our approach, we represent the text and lexicons terms through word embeddings before calculating the similarity distances. Thereby, we can catch the context background from the word embeddings on the vectorial space, a more robust representation than the presence of the exact terms.

3.2 Enhancing Text Representations with Semantics

Representing the natural language of texts has been a challenge for the NLP area, with vector space models being one of the most popular forms of representation. Each document is represented by a vector whose dimensions correspond to features extracted from the text. An example of this model type is BoW, where features are independent words. Despite being a simple model, BoW obtains good results in several tasks, being even challenging to supplant in several scenarios.

Even so, there are scenarios that demand a semantically more elaborate text representation to improve the model's performance on NLP tasks [32].

Distributed representation, such as word embeddings, is a prominent form of representation in this context.

The use of word embeddings has become quite popular and is proving to be a representation that can present much important information, as shown in the work of Baroni, Dino, and Kruszewski [16]. The authors conclude that models based on word embeddings present better results against models based on co-occurrence counting in different semantic tasks.

Another research tendency in this area is to propose models that, based on word embeddings, add even more information to the representation. An example is the Sinoara et al. [102] work, which presents two approaches to generate a semantic representation of document collections (NASARI+Babel2Vec and Babel2Vec) based on disambiguating. It uses word senses alone or word senses associated with embeddings based on words.

The representations proposed by Sinoara et al. [102] use the same vector space as embeddings and do not need large amounts of documents to train the models. The NASARI method [21], used as part of one of the approaches, builds vectors of word senses that can be applied to several languages, as it is based on the BabelNet [78]. The authors state that both NASARI+Babel2Vec and Babel2Vec representations present vectors close to related words or word senses. However, the NASARI+Babel2Vec representation is more advantageous, as the neighborhood based on word senses has more meaning and is more easily interpretable.

Sinoara et al. [102] detail various experiments carried out using six machine learning algorithms, nine datasets, various types of classification tasks at different difficulty levels, and two languages: English and Portuguese. These experiments indicate a strong performance by the proposed approaches, especially in the more complex scenarios in English. However, in the Portuguese language experiments, representations using BoW had better results, even though they were not considered suitable. The authors claim that the poor results on Portuguese datasets are due to the small coverage of linguistic resources in this language, which leads to the necessity of research efforts concerning this language.

Another way to add relevant information to the representation of a text is to use lexicons information. In addition to the studies of Lee et al. [61], Pateria [80], Maharjan et al. [68], and Ghanem et al. [37], several other works rely on lexicons to improve their text representation with relevant information. Unlike our work, the cited studies do not use semantic similarity in their representation.

Tumitan and Becker [104] and Avanço and Nunes [12] works calculate sentences and text polarity by summarizing the polarity of lexicon terms identified on text (add positive terms, and subtract negative terms).

Tumitan and Becker [104] use a sentiment lexicon to calculate a sentence-polarity score aiming to conduct a study to rate comments on Brazilian political news. They intended to detect whether user comments on online newspapers reflect external indicators of public acceptance (e.g., vote intention). They analyzed data referring to a São Paulo mayoral elections expressed as comments on Folha Online newspaper and used the external indicators provided by Datafolha polls. Among other findings, the authors discovered that considering the metrics developed, the sentiment has a moderate correlation with vote intention for the elected candidates. Firstly, Tumitan and Becker performed sentence polarity classification of the sentences that mention candidates. They used a Portugal Portuguese sentiment lexicon enriched with Brazilian and domain-specific terms. Sentiment words are identified, summarized (positive terms are added, and negative terms are subtracted), and the resulting sentiment is assigned to the sentence. Then, they calculated five metrics that relate positive and negative sentiments of an entity (candidate) and between entities. Finally, they compared the overall sentence sentiment concerning a candidate and the intent of voting for him.

Avanço and Nunes [12] proposed to perform sentiment classification for Brazilian Portuguese technology product reviews. They compared the performance of three different sentiment lexicons combined with simple strategies on the task. The authors proposed three simple lexicon-based classifiers and compared their achieved results. The first classifier considers the text's polarity as the sum of the identified sentiment lexicon words. If the sum is positive (strictly greater than zero), the opinion is positive; otherwise, it is negative. The second and third classifiers combine the lexicon polarities with linguistic information about contextual shifting. The second classifier considers negation contexts, and the third adds negation and intensification contexts. The results showed that the third classifier, which incorporates more semantics, performs better.

In order to obtain more sophisticated and valuable information from texts taking advantage of lexicons, some research associate the power of word embeddings and lexicons. Pre-trained (widely used) word embeddings contain semantic and syntactic information for a general context. Associating lexicon information can grant context-specific information to the representation. Some research implements this word embedding and lexicons association by calculating semantic similarity metrics between texts word embeddings and lexicons word embeddings, thus becoming independent of the presence of the exact terms exclusively. Examples of this approach are our proposed method and the studies of Jeronimo et al. [51], and Araque et al. [10]. Similar to our work, Jeronimo et al. use WMD as a semantic similarity metric. Araque et al. use the Cosine Distance. Both Jeronimo et al. and Araque et al. do not use the flow representation, leading to a possible loss of valuable information.

Jerônimo et al. [52] propose a subjectivity-based representation of texts to perform fake news detection. Claiming that the subjectivity levels of legitimate and fake news are significantly different, the authors use five Brazilian Portuguese subjectivity lexicons created by Brazilian linguists [8] to create subjectivity vectors to represent each document from a Brazilian news dataset. The authors calculate the WMD between each news sentence and the subjectivity lexicons on the embedding space. Then, an average of the distances of all sentences in each document for each lexicon is calculated, and these five averages finally form the feature vector. The results of the experiments presented in the work [52] are promising. The proposed method is as good as the TF-IDF (baseline) in the experiment that classifies news in general. The proposed method obtains better and more robust results when scenarios between domains (type of news - culture, sports, politics, and economics - or sources - Estadão and Folha) are considered. In another paper, Jeronimo et al. [51] used the same approach to perform two more tasks: automated essay scoring and identifying subjectivity bias in Brazilian presidential elections.

On the other hand, Araque et al. [10] proposed a model to perform sentiment analysis. The model represents each text as a concatenation of two vectors: the first, a vector representing the text by word or paragraph embeddings [72; 60]; and the second, a vector of the cosine distances between words in the text and selected words from a sentiment lexicon. To calculate the similarity vector, the model by Araque et al. first chooses the most frequent lexicon words in the training set. Then it generates a similarity matrix by calculating the cosine distances between the words in the text and the chosen lexicon words. So, it reduces this matrix to a vector, choosing the highest similarity value obtained for each word in the lexicon.

The authors present the results of several experiments conducted using datasets of short and long texts and even various lexicons. They compared the classification results using only the word embeddings representation, only the semantic similarity representation, and the association of both representations (proposed method). The results show that the proposed method presents better performance compared to the classification using only the semantic similarity representation in all combinations of dataset and lexicon. This fact is not repeated when compared to the classification only using word embeddings.

Likewise our and other works presented in this document, another possible effort to utilise the lexicon's power is to aggregate this information to neural networks. A tendency is to include the lexicon information in the word embeddings to represent the texts.

Wu et al. [110] use the lexicon to train a sentiment word classifier and use this classifier to create a sentiment word embedding, which is concatenated with the original word embedding to represent the words in the texts and feed the neural network for sentiment classification. Exhaustive experiments show that the proposed model obtains better results than other models used as baselines.

Fu et al. [36] create an attention mechanism based on the correlation of each sentence word embeddings and the lexicons (positive and negative) words embeddings. The final representation of sentences solves the problem of semantic ambiguity of words and feeds a Bi-LSTM network. The output of this network feeds another attention mechanism that captures essential information about the context of different representation subspaces in different positions. At the top of this mechanism, there is a classification layer. From the experiments in four datasets for sentiment classification, the proposed approach surpasses the baseline methods in three of them.

3.3 Positioning in relation to Related Work

This section presents the position of this work compared to the related works described in this chapter. Table 3.1 shows the comparison in various dimensions.

The categories (columns) considered for the construction of the table are explained next:

- Flow: works that use flows to represent texts;
- Feature Extraction from Flows: studies that initially represent texts as flows and perform feature extraction from them to conduct NLP tasks;
- Pre-trained Word Embeddings: approachs that base text representation on pre-trained word embeddings;
- Lexicons: researches that enhance the text representation with lexicon information;
- Semantic Similarity: studies that use semantic similarity calculated in a vectorial space to embed the lexicon information to the solution;
- Lexicon Enhanced Word Embeddings: researches that base text representation on incorporating lexicon information to pre-trained word embeddings.

This research contemplates five of the six categories presented. To the best of our knowledge, our method is the only one representing texts as flows through calculating semantic similarity metric between each sentence and a lexicon over a embedding vectorial space and then extracting features based on the audio analysis to perform classification tasks. In this way, this work tries to gather and avail the benefits that each presented technique can bring to enrich the text representation to enhance efficacy on classification tasks.

Table 3.1: Comparative table of related works.

		Feature Extraction	Word		Semantic	Lexicon Enhanced
	Flow	from Flows	Embeddings	Lexicons	Similarity	Word Embeddings
Mao and Lebanon [69]	X					
Wachsmuth and Stein [107]	X					
Filatova [33]	X	X				
Seo and Jeon [97]	X	X				
Lee et al. [61]	X	X		X		
Pateria [80]	X	X		X		
Maharjan et al. [68]	X			X		
Ghanem et al. [37]	X			X		
Sinoara et al. [102]			X			
Tumitan and Becker [104]				X		
Avanço and Nunes [12]				X		
Jeronimo et al. [51]			X	X	X	
Araque et al. [10]			X	X	X	
Wu et al. [110]			X	X		X
Fu et al. [36]			X	X		X
This Research	X	X	X	X	X	

Chapter 4

Text Flows Representation and Audio-Like Features Extraction

This chapter presents our proposed approach for representing texts by flows, incorporating lexicon information via semantic similarity distance on an embedding space. This chapter also depicts the so-called Audio-Like Features (ALFs) extraction.

Figure 4.1 shows a diagram of the proposed approach. First, the method splits texts into sentences to avoid losing local information that can be crucial for differentiating two kinds of texts.

Thenceforward, the approach calculates the WMD from each sentence to a lexicon on an embedding vector space to compose the text flow (or flow - for short). Therefore, besides incorporating the specific context information from the lexicon into the representation, the approach avoids depending only on finding (and counting) known vocabulary terms.

Then, the method proceeds with the ALFs extraction. As in audio analysis, a small number of ALFs are calculated over the entire flow (flow-based features). Most ALFs are also calculated over smaller flow parts, called frames. So, each flow is fragmented into frames, and then both flow-based and frame-based ALFs are calculated. These features are then used to feed classification algorithms, creating models to perform classification tasks.

In the following sections, the flows representation creation, frame fragmentation, and the ALFs extraction are detailed and discussed.

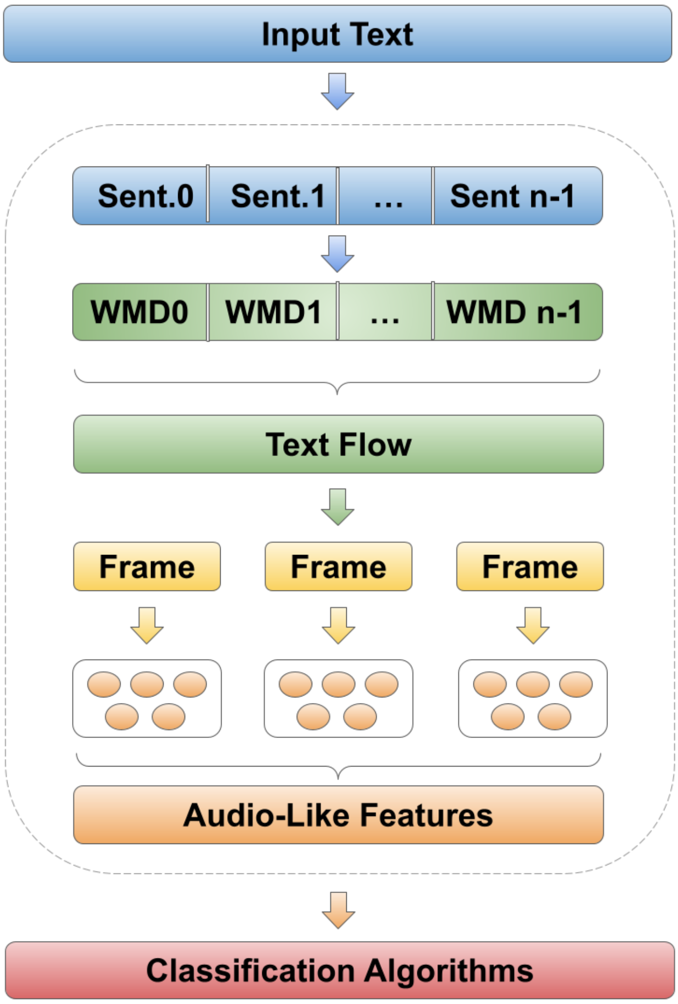

Figure 4.1: Proposed approach: Text Flow representation and Audio-Like Features extraction.

4.1 Text Flows Representation

Fig. 4.2 illustrates how the flows representation is created. Our approach produces the flows by calculating the sequence of semantic distances (WMD) from each sentence of a text to a lexicon in an embedding space. In other words, the WMD is computed between the word embeddings representation of the sentences and the lexicon. The WMD values belong to the [0,1] interval. The smaller the WMD value is, the greater the similarity between the sentence and the lexicon.

Our approach depends on the word embeddings model and lexicons in a given language.

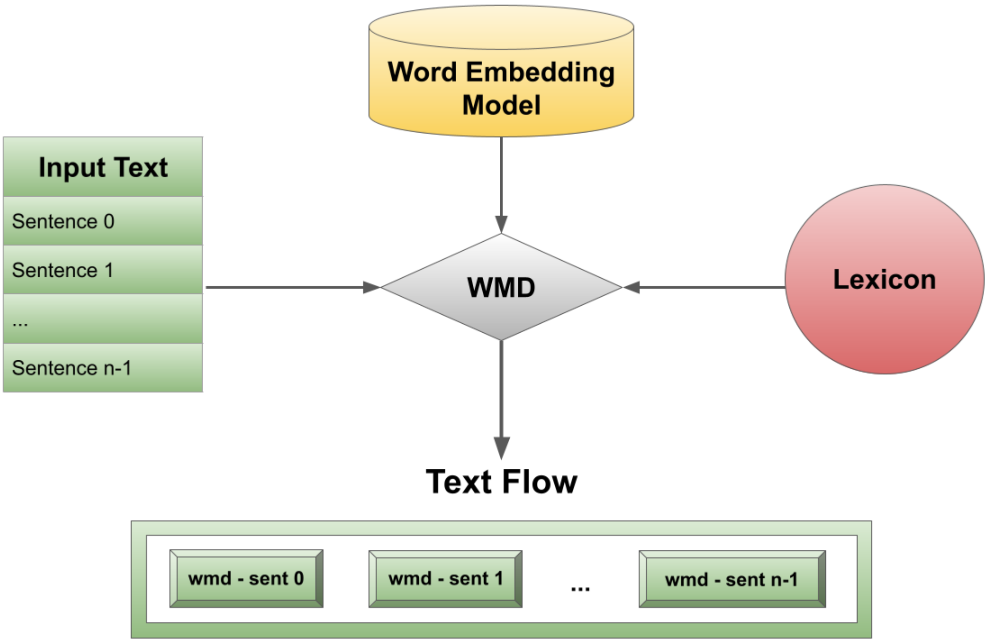

Figure 4.2: Text Flow Creation.

Regarding the word embeddings, there are some widely used pre-trained models, especially in English; furthermore, a particular word embedding model can be trained when an appropriate one is not available. As for lexicons, there are several available in the literature dealing with, for example, sentiment, subjectivity, or argumentation (again, mainly in English). However, there is also the possibility of creating new lexicons or translating existing lexicons into other languages.

4.2 Text Flows Frame Fragmentation

As already discussed, given the text flow's approximation to a digital signal and the relationship between text, speech, and audio, the ALFs are inspired by well-established audio analysis features. The majority of the audio analysis features (and ALFs, consequently) are calculated over frames: smaller parts of the audio signal (as discussed in Section 2.2). So, the fragmentation of the flows into frames is necessary.

However, unlike the fragmentation method of audio analysis that splits the audio signal into frames of the same size, the proposed approach fragments the flows in a fixed number of frames. This adaptation is necessary to compare the same excerpts from different texts (which often tend to have different sizes). For example, regardless of the number of sentences in a text, the first frame represents the first part of the text, being comparable to the first part of another text.

The definition of the number of frames to fragment the flows depends on the dataset under use. For example, it is possible to obtain many more frames when dealing with books than with evaluations of products or services.

In audio analysis, frames can be overlapped, generating intermediate frames before the feature extraction. Frequently, this overlapping procedure enhances the analysis. During this research, we empirically tested 20%, 30%, and 50% frame overlapping, but the obtained classification results were lower than using non-overlapping frames. The main reason could be the vast difference between the number of samples present in audio and texts. While there are numerous samples for each second in audio, our proposed approach uses each sentence as a sample in texts. Thus, overlapping the flows frames end up confusing the classification models.

4.3 Audio-Like Features

This section presents the procedure taken in this research to choose well-established audio features and how they were adapted to the text domain, creating the Audio-Like Features extracted from the text flows.

We followed the taxonomy for audio features presented by the review of Mitrovic et al. [75] and extended by the study of Alias et al. [7]. The taxonomy initially divides audio features based on their semantic interpretation that indicates whether or not the feature represents elements of human perception. The perceptual features approximate semantic properties known by human listeners (e.g., loudness and harmonicity). The physical features represent audio signals in mathematical, statistical, and physical properties without directly highlighting human perception (e.g., Fourier transform coefficients and the signal energy). As the perception of a sound does not apply to texts, all the ALFs are adapted from physical features.

Another property present in the taxonomy is the audio feature domain. The domain allows for understanding the feature data and provides information about the extraction process and the computational complexity. This method elected features from the time and frequency domains among all existing domains due to their low complexity and adequacy to the low number of samples obtained from the flows. Time-domain features directly describe the waveform, not requiring any transformation on the original audio signal. Thus, the adapted ALFs in this domain are directly extracted from the flow. As in audio analysis, the flow is submitted to a Short-Time Fourier Transform or derived from an autocorrelation analysis to extract ALFs in the frequency domain. The features on other domains are not suitable for the text domain or are much more complex and need a large number of samples to be correctly computed, and then an adaptation would not be applied to small and medium texts.

The Mitrovic et al. [75] taxonomy also presents the temporal scale of a feature property. This property is related to the portion of the signal the features are extracted. Features can be frame-level when extracted from individual frames; intraframe features when the calculation is based on two or more frames; global features when computed for the entire audio signal. The majority ALFs are frame-level features. The remaining features are global features, called flow-level features.

Our approach introduces twenty-five features. The features were chosen for being well-established and showing good performance in the audio analysis research field. Other determining characteristics were simplicity and facility to adapt. Thus, this choice would be a safer way to validate the implementation of this method. All features had to be implemented and not reused from other libraries since we had to adapt each of them to the new text-domain. Sixteen features are from the time domain, and nine are from the frequency domain. Also, (not the same) sixteen features are frame-level, and nine are flow-level. Table 4.1 presents the names of the features, their domain, and temporal scale. Following, we describe and discuss all proposed features.

4.3.1 Time-Domain Frame-Level Features

This subsection presents all the time-domain frame-level features extracted by this proposed approach.

Table 4.1: Audio-Like Features.

Feature	Domain	Temporal Scale
Energy	Time	Frame-level
Median-Crossing Rate	Time	Frame-level
Energy Entropy	Time	Frame-level
Linear Prediction Median-Crossing Ratio	Time	Frame-level
Text “Waveform” Minimum	Time	Frame-level
Text “Waveform” Maximum	Time	Frame-level
Text “Waveform” Diff	Time	Frame-level
Volume	Time	Frame-level
High WMD Segments Mean	Time	Flow-level
High WMD Segments Standard Deviation	Time	Flow-level
High WMD Segments Median	Time	Flow-level
Low WMD Segments Mean	Time	Flow-level
Low WMD Segments Standard Deviation	Time	Flow-level
Low WMD Segments Median	Time	Flow-level
Area of High WMD Segments	Time	Flow-level
Log Attack Position	Time	Flow-level
Spectral Flux	Frequency	Frame-level
Spectral Entropy	Frequency	Frame-level
Spectral Energy Ratio	Frequency	Frame-level
Spectral Flatness	Frequency	Frame-level
The Spectral Crest Factor	Frequency	Frame-level
Spectral Skewness	Frequency	Frame-level
Spectral Kurtosis	Frequency	Frame-level
Pitch	Frequency	Frame-level
Jitter	Frequency	Flow-level

Energy

The Energy feature reflects the total magnitude of the lexicon in the text. Let $x_i(n), n = 1, ..., F_L$ be the sequence of WMD of the i-th frame, where x is a WMD from a sentence to a lexicon and F_L is the length of the frame. The implementation of Energy is defined as in Equation 4.1:

$$E(i) = \frac{1}{F_L} \sum_{n=1}^{F_L} |x_i(n)|^2 \tag{4.1}$$

Here we normalised the Energy by dividing it by F_L to remove the dependency on the frame length.

Since the WMD is a distance metric between two texts, the bigger the WMD is, the farer the sentence is from the lexicon, and the weaker the relationship between sentence and lexicon. So, the stronger a lexicon appears in the frame, the smaller the frame's Energy.

In audio analysis, the Energy conveniently represents the amplitude variation over time [113; 87]. It provides a way to distinguish voiced from unvoiced fragments because values for the unvoiced components are generally significantly smaller than those of the voiced components. It can also be used to distinguish audible sounds from silence, and the change in its pattern over time may reveal the rhythm and periodicity properties of sound.

Comparatively, in the text domain, Energy can help distinguish between texts or parts of texts that present weak or strong relationships to the lexicon. For example, suppose the flow representation considers a subjectivity lexicon. In that case, the Energy can help differentiate more subjective texts (lower Energy, since WMD is a distance metric) from less subjective texts.

Median-Crossing Rate

Median-Crossing Rate (MCR) is an adaptation of the Zero-Crossing Rate (ZCR) audio analysis feature. As in the audio signal the amplitude varies from -1 to 1, the ZCR is the number of times the signal changes value (crossing the zero line).

As the WMD values range is $[0, 1]$, the MCR implementation uses the median of all WMD of the flow as "line" to calculate the number of times it is traversed in a frame. The median metric was chosen to generate an equilibrium between the number of WMD values above and below the "line", considering the entire flow.

Therefore, MCR is the rate that the flow crosses its median line (considering the frame). The MCR is defined according to Equation 4.2:

$$MCR(i) = \frac{1}{2F_L} \sum_{n=1}^{F_L} |msgn[x_i(n)] - msgn[x_i(n-1)]| \tag{4.2}$$

where x is a WMD from a sentence to a lexicon, F_L is the length of the frame and $msgn$ is a modification of sign function, the Median Sign Function, denoted by Equation 4.3:

$$msgn[x_i(n)] = \begin{cases} 1, & \text{if } x_i(n) > median. \\ -1, & \text{if } x_i(n) < median. \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

In audio analysis, the ZCR is a metric of signal noisiness. In other words, ZCR reflects the variation level of the sounds on the signal frame. For instance, it is used to distinguish between voiced and unvoiced signals because unvoiced speech components typically have much higher ZCR values than voiced ones. The unvoiced fragment presents much more variation than the voiced, which is more stable (even if it present a bigger Energy) [113].

Following the same reasoning, MCR can be interpreted as a measure of the lexicon variation present on a text, helping to distinguish between texts that present different variation levels of a considered lexicon.

Energy Entropy

Shannon entropy plays a central role in information theory as a measure of information, choice, and uncertainty [98]. In audio analysis, the same entropy can measure the “peakines”, or the abrupt changes in the signal energy level. For example, unvoiced sounds are flatter (less abrupt changes) and present higher entropy than voiced sounds [40; 85].

Adapting this feature to the text domain, the Energy Entropy measures abrupt changes in the lexicon Energy level of a flow. For example, it can detect if a frame presents sentences with profoundly different levels of subjectivity.

The implementation follows the Shannon Entropy formula. First, each frame is divided into K sub-frames. Then, for each sub-frame j, we compute the $Esubframe_i$, its Energy as in (4.1) and divide it by the total frame Energy, $Eframe_i$. The division is necessary to treat the resulting sequence of sub-frame energy values, e_j, $j = 1, ..., K$, as a sequence of probabilities, as in (4.4):

$$e_j = \frac{Esubframe_j}{Eframe_i} \tag{4.4}$$

where

$$Eframe_i = \sum_{k=1}^{K} Esubframe_k \tag{4.5}$$

At a final step, the entropy, $Ent(i)$ of the sequence e_j is computed according to Equation 4.6:

$$Ent(i) = -\sum_{j=1}^{K} e_j * log_2(e_j) \tag{4.6}$$

The more significant changes the frame presents, the lower the Entropy Energy resulting value is.

Linear Prediction Median-Crossing Ratio

In audio analysis, Linear Prediction Zero-Crossing Ratio (LP-ZCR) is the ratio of the zero-crossing ratio of the frame waveform, and the zero-crossing ratio of the output of a linear prediction analysis filter [30; 86]. The feature quantifies the degree of correlation in a signal. It helps distinguish between different audio types, such as voiced (higher correlated) and unvoiced speech (lower correlated).

The Linear Prediction Median-Crossing Ratio (LP-MCR), the proposed adaptation of LP-ZCR, is calculated as the Equation 4.7.

$$LP - MCR = \frac{MCRflow}{MCRlp} \tag{4.7}$$

where $MCRflow$ is the MCR obtained from the flow and $MCRlp$ is the MCR obtained from the output of the Levinson-Durbin linear prediction filter over the flow (both considering the frame and calculated as explained in Equation 4.2.

LP-MCR helps discriminate between flows (or frames) that show different correlation degrees. For example, considering an argumentation lexicon, a more argumentative text is more correlated than an informative one.

Text “Waveform” Features

Now we introduce a set of three features called Text “Waveform” Features: Text “Waveform” Minimum (TW_Min), Text “Waveform” Maximum (TW_Max), and Text “Waveform” Diff (TW_Diff).

The adaptation of TW_Min and TW_Max features comes from the audio analysis MPEG-7 audio waveform (AW) descriptor. The AW descriptor gives a compact description of the shape of a waveform by computing the minimum and maximum amplitude samples within frames. The descriptor’s purpose is to display and compare waveforms, been used as a feature in environmental sound recognition, for instance [75; 7]. Therefore, TW_Min and TW_Max are the minima, and the maximum WMD observed in a frame, respectively.

The adaptation of TW_Diff is an approximation to the shimmer feature. Shimmer computes the intraframe cycle-to-cycle variations of the waveform amplitude. As in the text domain we do not have many samples that could generate various cycles in a frame; we propose the difference between TW_Min and TW_Max (TW_Diff) as the shimmer approximation.

These three features can help differentiate texts or parts of a text that present crucial and particular dissimilar points in their flows. For example, when considering a positive polarity sentiment lexicon, positive and negative texts would present different TW_Min (minimum WMD).

Volume

Volume is defined as the Root-Mean Square (RMS) of the waveform magnitude within a frame in audio analysis. It reveals the magnitude variation over time and is commonly used for silence detection and speech/music segmentation [64].

In the text domain, Volume’s adaptation reveals the flow WMD behavior variation throughout the text and is calculated by the RMS of each frame WMDs. The flow or frame WMDs can present different behaviors in different kinds of texts. For example, regarding a positive polarity sentiment lexicon, the first frame WMD variation on a positive review is probably different from a negative review.

4.3.2 Time-Domain Flow-Level Features

This subsection describes the time-domain features extracted from the entire flow.

Amplitude Descriptors

Mitrovic et al. [74] described the Amplitude Descriptors (AD), a set of features capable of describing characteristics of the waveform, such as peaks and silence. Based on an adaptive threshold, initially, the features express the length of high and low amplitude sequences of samples and the area corresponding to the high amplitude sequences. The authors consider statistical properties of the initial features to build features that describe entire sample files.

In the adaptation proposed in this thesis, the AD is a set of seven individual features that characterize the flow in terms of "near" and "far" segments to the lexicon [74]. In other words, it identifies regions of the flow that present low and high WMD.

The implementation first split the flow into segments through an adaptive threshold. The threshold is the sum of the flow WMDs mean and standard deviation (often used in audio analysis). Based on this threshold, we calculate the length of high WMDs segments (LoHWS). The length of a high WMDs segment represents the number of consecutive sentences with a value greater or equal to the threshold. LoHWS outlines the distribution of the length of peaks (the more distant sentences from the lexicon) in the flow.

Similarly, we determine the length of a low WMDs segment (LoLWS) as the number of consecutive samples with a lower value than the threshold. LoLWS describes the distribution of length of the valley portions (the more close sentences from the lexicon) in the flow.

Sequences with high WMDs segment can be additionally defined by the corresponding area below the flow. We compute the area of high WMDs (AHW) as the area between the threshold and the signal in a LoHWS. In other words, the AHW represents the extent of peaks in the flow.

Finally, the AD set is formed by the mean, standard deviation, and median of all the calculated LoHWS (AD_HWS_Mean, AD_HWS_Std, and AD_HWS_Median, respectively); by the mean, standard deviation, and median of all the calculated LoLWS (AD_LWS_Mean, AD_LWS_Std, and AD_LWS_Median, respectively); and by the mean of all the calculated AHW areas (AHW_Mean).

These features can help discern texts that present different portions of sentences with a strong or a weak relationship with the lexicon.

Log Attack Position

Log Attack Position is the logarithm of the position of the highest WMD in the flow. It approximates the Log Attack Time from audio analysis, the logarithm of the elapsed time from the beginning of a sound signal to its first local maximum, and characterizes the beginning of a sound [34]. We correlated the elapsed time to the position of the sentence in the text. Considering the existence of few samples on texts, we correlated the waveform first local maximum to the highest WMD (the highest peak and farther to the lexicon sentence). This feature distinguishes texts that present the farther sentence to the lexicon on different points.

4.3.3 Frequency-Domain Frame-Level Features

The frequency-domain audio features constitute the most extensive group of audio features reported in the literature [75]. Hence, it was possible to adapt some of these features to represent texts. Among the spectral features presented in this subsection, seven are computed after generating the STFT output. Only the Pitch feature is extracted from the flow resulting from an autocorrelation function.

Spectral flux

In audio analysis, Spectral Flux quantifies abrupt changes in the spectral energy distribution over time. For example, signals with slowly varying or nearly constant spectral properties (e.g., noise) have low Spectral Flux [75; 7].

In our text domain, the Spectral Flux quantifies abrupt changes in the spectral energy related to a lexicon between two consecutive frames. Spectral Flux could help discern between texts with different lexicon distance levels throughout the text. For example, considering an argumentation lexicon, an argumentative text may present a slowly varying behavior (low Spectral Flux). In contrast, a text that only presents a few argumentative sentences in its final portion would show high Spectral Flux in the last frame.

Let $X_i(k), k = 1, ..., Wf_L$ be the sequence of the magnitude of the STFT coefficients of the i-th frame (STFT output), where x is a magnitude of one coefficient, and Wf_L is the length of the frame. Spectral flux is computed as the squared difference between the normalized magnitudes (like Energy) of the spectra of the two successive frames, as showed by Equation 4.8:

$$SFlux_{(i,i-1)} = \sum_{k=1}^{Wf_L} \left(EN_i(k) - EN_{i-1}(k)\right)^2 \tag{4.8}$$

where $EN_i(k)$ is the k-th STFT coefficient at the i-th frame, calculated as in Equation 4.9.

$$EN_i(k) = \frac{X_i(k)}{\sum_{l=1}^{Wf_L} (X_i(l))} \tag{4.9}$$

Spectral Entropy

The Spectral Entropy is computed similarly to the Energy Entropy (Equation 4.6), but now, the computation occurs in the spectral domain, that means, using the STFT output [73]. The discussion is also equivalent; Spectral Entropy measures abrupt changes in the lexicon Energy level of a flow on the spectral domain (flow STFT output).

Spectral Energy Ratio

Spectral Energy Ratio is a feature adapted from the Subband Energy Ratio. The Subband Energy Ratio is usually defined as a measure of the normalized signal energy along with a predefined set of frequency subbands, describing the signal energy distribution of the spectrum [75; 7; 76].

The Spectral Energy Ratio is implemented as the ratio between the frame Spectral Energy and the entire flow Spectral Energy (take the Equation 4.1 as reference). In a broad sense, it also roughly describes the flow energy distribution of the spectrum.

Spectral Flatness and Spectral Crest Factor

The Spectral Flatness measures uniformity in the frequency distribution of the power (squared) spectrum in audio analysis. It is computed as the ratio between the geometric and the arithmetic mean of a subband. Noise-like sounds have a higher flatness value, while tonal sounds have lower flatness values [7; 75]. This thesis adaptation of Spectral Flatness is implemented as the ratio of the geometric and the arithmetic mean of the squared spectral magnitudes (power spectrum) from the STFT output of each frame [89].

In audio analysis, the Spectral Crest Factor measures the "peakiness" of a spectrum, being inversely proportional to the Spectral Flatness and also used to distinguish noise-like and tone-like sounds [75]. This proposed method adapted the Spectral Crest Factor as the ratio of the maximum power spectrum and the mean power spectrum of a frame [62][1].

Portions of texts (represented by frames) that present a less varying distance to a lexicon have a higher flatness value and a lower crest factor value. In contrast, those with more variance on distances to the lexicon have lower flatness values and higher crest factor values.

Spectral Skewness and Spectral Kurtosis

Spectral Skewness and Spectral Kurtosis are two features calculated following the probability theory and statistics in audio analysis and were also implemented by our proposed approach. They are the third and fourth moments of the spectral distribution, respectively.

The Spectral Skewness measures the asymmetry of the spectral distribution around its mean value. If the Skewness is negative, there is more energy on the right side of the spectral distribution; if it is positive, there is more energy on the left side [81]. At the same time, the Spectral Kurtosis describes the flatness of the spectral distribution around its mean. A Kurtosis value lower than three describes a flatter spectral distribution, while a value bigger than three describes a peaker distribution [81]. These are two other features that can capture the relationship between text and lexicon, distinguishing texts with a different connection to the lexicon.

Pitch

Also known as Fundamental Frequency, the Pitch feature is defined as the first peak of the local normalized spectro-temporal autocorrelation function [23]. Autocorrelation is the similarity between observations considering a time lag between them, being useful for finding

[1] The Spectral Flatness and Spectral Crest Factor implementation are based on the Librosa Python library implementation (https://librosa.org/)

repeating patterns in the signal. Autocorrelation values range from -1 to 1. A negative value represents negative autocorrelation, and a positive value represents positive autocorrelation.

The present method identifies Pitch as the first peak of the frame's spectral autocorrelation function output[2]. Texts flow presenting different Pitch values demonstrate distinct maximum autocorrelation values considering a particular lexicon.

4.3.4 Frequency-Domain Flow-Level Feature

The unique feature to be presented in this subsection is Jitter. Jitter is the cycle-to-cycle Pitch variations or the absolute mean difference between consecutive periods of an audio signal [31]. Jitter is adapted to a frame-by-frame pitch variation in the domain proposed in this thesis. It is calculated by the mean of the absolute difference between the pitches of two consecutive frames [31]. The Jitter feature is a form of analysing Pitch variation throughout the flows.

4.3.5 Discussion About Not Implemented Audio Features

Besides the perceptual features and other more complex and unsuitable physical domain features, for example, wavelet, image, and cepstral domains, some features on the considered time and frequency domains were analysed and discarded to compose the ALFs set.

The time-domain MPEG-7 temporal centroid represents the time instant containing the signal's most significant average energy. It is calculated as the temporal mean over the signal envelope, thus measured in seconds. Considering that it does not make sense a time measure on a continuous instant in texts and the absence of a significant samples number that justifies an envelope on frames implementation, this feature was dumped.

The rhythm-based physical features are a sub-category of time-domain physical discarded because the idea of rhythm does not make sense in the text domain. It would also require a more considerable number of samples and signal repetition.

The extended Alias et al. taxonomy [7] comprises some other categories of frequency-domain features (compared to [75]). This approach did not adapt any feature of three cate-

[2] We used the Spectrum Python library autocorrelation function to implement our Pitch feature (https://pyspectrum.readthedocs.io/)

gories. The primordial reason to discard the autoregression-based features is that they require a fixed size of frames, and this approach proposes frames of different sizes. The brightness-related features demand the frequency of sampling rate, not applicable on the studied domain. The chroma-related features were not implemented because they are closely related to musical notes.

The ALFs are based on features of the STFT-based, Tonality-related, and Spectrum shape-related frequency-domain physical categories. The majority of the discarded features that belong to these categories required numerous samples. Some other features were not adapted due to other reasons besides needing a larger number of samples, for example: stereo panning spectrum and pitch profile features. The stereo panning spectrum feature was discarded because it relates to stereo (two channels) sounds. The pitch profile feature considers different instruments.

4.4 Final Considerations

This chapter presented the proposed approach for representing texts through flows and extracting valuable features from them. Our method is inspired by the area of audio analysis, bringing to NLP concepts such as frame division and feature extraction typical of that area. Innovatively, the method merges various techniques to enhance text representation to improve classification efficacy.

The chapter detailed the implementation of the approach, including the presentation of twenty-five Audio-Like Features, describing their implementation, and discussing their relation to the audio analysis features and impact on text analysis. The chapter also discussed the motives for not implementing some audio analysis features.

It is worth discussing that, to be correctly calculated, some features require a certain minimum number of sentences per frame or sub-frame, for example, MCR, Energy Entropy, and Spectral Entropy. Thus, this minimum requirement must be considered when defining the number of frames and the K parameter, as it is required at least two sentences per subframe. Considering this, our model is not appropriate to analyse tiny texts, such as those microblogs.

The next chapter describes the experiments performed to verify the approach's effectiveness in classification tasks. Also, the results of these experiments are presented and discussed.

Chapter 5

Experimental Evaluation and Discussion

This chapter presents the experimental evaluation of the proposed approach, describing the configuration of the different experiments performed and discussing the results obtained.

The experiments assess this thesis method efficacy in five text classification tasks: 1) Fake News Detection in English, 2) Fake News Detection in Portuguese, 3) Newspaper Columns versus News Classification, 4) Movie Reviews Sentiment Polarity Classification in English, and 5) Book Reviews Sentiment Polarity Classification in Portuguese.

We evaluate the approach's competitiveness by comparing its classification results to two baselines that embed semantic information in text representation: Paragraph Vector (D2V) [60] - a static word embedding - and the strong BERT [28] - a contextualized word representation. Besides the classification employing models created from the ALFs, D2V, and BERT features separately, we performed a classification involving models obtained by a combination of the ALFs and D2V features (D2V+ALF) and the ALFs and BERT features (BERT+ALF), aiming to verify if associating our method can improve the D2V and BERT methods efficacy.

5.1 Overall Experiments Configuration

5.1.1 Flows Creation and Frame Division

Initially, all involved texts were submitted to a pre-processing step to remove stop-words and accent marks. The pre-processed version of texts was used as input to the comparison baselines and to construct the flows.

Proceeding with the generation of the flows, texts were fragmented into sentences, and sentences containing up to two words were removed. Subsequently, the approach calculates the WMD distances between each text sentence to the lexicon of each task in the embedding space (as described in Section 4.1). In the tasks involving the English language, the widely used Word2Vec article [72] pre-trained word embeddings[1] were used. On the other hand, tasks involving the Portuguese language use the word embeddings word2vec model (made available by Sales et al. [94]), which was created from a large volume of Wikipedia articles. The sequence of all sentences WMD of a text constitutes the text flow.

Once the flows were created, it was necessary to fragment them into frames to extract the ALFs. It is the moment to decide the number of frames and the K parameter (number of sub-frames to break each frame into). It is essential to consider that in the text domain, we have so much fewer samples (sentences in texts) than in the audio analysis due to the nature of sounds and the sampling procedure. It is also worth remembering that each frame must respect the minimum requirements. In other words, each sub-frame must have at least two sentences to achieve correct features computation. Moreover, all ALFs become more descriptive when frames and sub-frames have more sentences than the minimum requirements because it is possible to consider more information to the ALFs calculation.

For each classification task, the average size of the texts of each class was analysed to define the frame number and K-parameter. All the texts classes are news or reviews, small texts presenting an average size of between 14 and 41 sentences. Through this average size analysis and some empirical tests, the flows were divided into three frames and two sub-frames (K=2), in all classification tasks. Therefore, in the present experiments, the texts might have no less than twelve sentences (three frames x two sub-frames x two sentences per sub-frame).

We propose a padding technique employed to texts that do not achieve the minimum number of sentences, called Last Frame Sentence Padding (LFSP). This technique is applied after the process of splitting the flow into frames. The LFSP consists of repeating the WMD value of the last sentence of each frame until it reaches the minimum defined size (four sentences in this thesis evaluation setup case).

[1] https://code.google.com/archive/p/word2vec/

Fig. 5.1 illustrates the application of the LFSP to a flow containing seven sentences (WMDs) that needs to be split into three frames with K = 2. When split in frames, the flow presents three, two, and two sentences in Frame 0, Frame 1, and Frame 2, respectively. Therefore, as K = 2, each frame must have four sentences for the correct calculation of all ALFs; the value of the last sentence of Frame 0 is repeated once, and the last value of the other frames, twice. In a rarer case, if the text does not present at least the number of sentences equal to the number of frames, the value of the text's last sentence is repeated until the end.

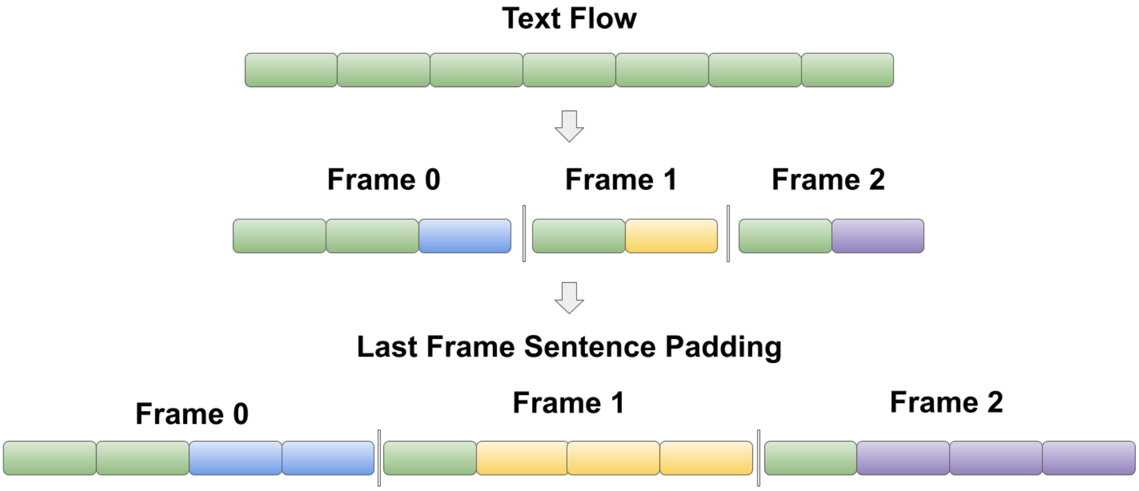

Figure 5.1: Last Frame Sentence Padding applied to a seven-sentences flow.

This technique can be more advantageous than, for example, performing the padding only at the final of the flow, sustaining some essential text characteristics. For instance, preserving each sentence's positioning is feasible since it continues on the original frame (the original portion of the text). Repeating the value of the last sentence also allows maintaining characteristics that the author wanted to explore in that portion of the text, such as continuity of the presented content, which would not occur if, for example, a predetermined value was filled in. Considering the experiments performed, the LFSP was applied to 4.76% of the texts, on average.

After applying the LFSP to the flows that need it, it is time to extract the ALFs to each flow. Each text is represented by one flow per lexicon dimension used in the experiment.

For example, suppose a sentiment polarity lexicon with negative and positive dimensions is used. In that case, each text will be represented by two flows, one formed by WMDs to the negative polarity lexicon dimension and the other formed by WMDs to the positive polarity lexicon dimension. As the method proposes sixteen frame features and nine flow features, and the number of frames to split the flows into is three, the feature vector comprises 57 features (3 frames x 16 frame features + 9 flow features) for each lexicon dimension.

5.1.2 Baselines

To conduct the D2V experiments, we trained a model using the remaining texts from each dataset that would not be used in the experiments. Then, we created a 100-dimension D2V representation for each text.

The BERT models used were the English BERT-base for the English tasks and the multilingual BERT-base for the Portuguese tasks. Each text's representation was generated considering the 512 first words and comprised 768 dimensions.

5.1.3 Classification Models

This study employed two groups of classification models: Shallow Learning (SL) and Deep Learning models (DL).

A train-validation-test split was randomly applied with a 70/15/15 distribution for all tasks.

The SVM, Logistic Regression, Random Forest, and XGBoost models were considered concerning the SL algorithms. A grid search was performed in all algorithms aiming to find a better suitable model configuration for each task by testing various hyperparameter configurations. The grid search used the train and validation splits. Then, the classification of the test split was done using the best model of each SL algorithm.

Concerning the DL models, this study involved the classification employing CNN, BiLSTM, and GRU models as they have demonstrated excellent efficacy for text classification [50]. All models were trained using the train and validation splits, considering several learning rates, numbers of neurons, and epochs.

This thesis presents and discusses the results obtained by each task's best SL and DL models.

5.2 Fake News Detection in English

The necessity for fake news detection is clear and present given the massive dissemination allowed by social media and messaging applications and its consequences[2].

Usually, documents committed to reporting facts truthfully and impartially, such as the news from reliable newspapers, tend to use more objective language, avoiding words and expressions that denote sentiment or a more argumentative tone. On the other hand, documents that seek to persuade the reader tend to use a more subjective language [71; 108]. For example, fake news tries to convince the reader of something not true. Even so, fake news aims to pass as legitimate, making its identification often not trivial.

5.2.1 Experiment Description

Dataset

The dataset used was compiled and made available by Jeronimo et al. [58] and encompasses 5,994 legitimate news and 218 fake news written in English. The legitimate news was taken from the *All The News Dataset* available at Kaggle[3], with 2,598 coming from CNN[4], 1,798 from The Guardian[5] and 1,598 from The New York Times[6], published between 2016 and 2017. Fake news, in turn, were compiled by Torabi ASR and Taboada [11], with 103 political news coming from Snopes[7], 75 political news coming from Horne and Adali's work [47] and 40 stories from Buzzfeed's top-ranked fake news[8]. All fake news are fact-checked.

[2] https://www.acritica.com/channels/coronavirus/news/crenca-nas-fake-news-potencializa-disseminacao-e-problemas-da-pandemia

[3] https://www.kaggle.com/snapcrack/all-the-news

[4] www.cnn .com

[5] www.theguardian.com

[6] www.nytimes.com

[7] https://github.com/sfu-discourse-lab/

[8] https://github.com/BuzzFeedNews/ 2017-12-fake-news-top-50

Lexicons

In this experiment, three sets of lexicons were combined, representing different dimensions of subjectivity in English.

The first set was compiled by Recasens et al. [91] and comprises six different dimensions of terms that tend to induce bias in texts, called "bias-inducing" by the authors:

- Factive Verbs: assume the truth of a complement clause. Comprises 27 terms, for example: realize, forget, exciting.
- Implicative Verbs: implies the truth or falsity of the complement clause. Comprises 32 terms, for example: succeed, fail, neglect.
- Assertive verbs: verbs whose complements assert a proposition. Comprises 66 terms, for example: believe, figure, affirm.
- Hedges: terms used to reduce commitment to the truth of a proposition. Comprises 100 terms, for example: apparently, could, estimate.
- Reporting Verbs: often used to describe the activities or actions of a third person. Comprises 181 terms, for example: accuse, assure, claim.
- Bias-inducing lemmas: examples of the 654 terms that compose it: advocate, amazing, barbarian.

The second set of lexicons was presented by Wilson et al. [109] and is part of the Multi-Perspective Question Answering project (MPQA) Subjectivity Lexicons[9]. This set is divided into positive and negative sentiment polarities (two dimensions) and classified into strong and weak subjectivity. Only terms belonging to the category of strong subjectivity of both polarities were used, resulting in 3,078 terms of negative polarity and 1,482 of positive polarity.

The third set was proposed by Deng et al. [24], based on a type of opinion inference that arises when opinions are expressed concerning events, generating positive or negative effects related to them. The terms extracted from documents that present subjectivity, such as blogs and editorials, present sentiment polarity (negative or positive) in two categories:

[9] https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

Gold Standard, extracted by annotators; and EffectWordNet, extracted automatically by the method proposed by Deng et al.[24]. Only terms from the Gold Standard category were used, comprising 1,003 terms of negative polarity and 493 of positive polarity.

Evaluation Metric

Given the imbalance in each class' number of texts and the greater interest in the fake news minority class, this one was configured as the positive class. It is worth mentioning that fake and legitimate news numbers are also imbalanced in the real world.

The class imbalance is also considered on choosing the evaluation metrics. The Area Under the Precision-Recall Curve (PR-AUC) was used because, as discussed in the works [93; 25], it is particularly suitable for the analysis of scenarios in which there is a significant class imbalance. The PR-AUC considers precision values at each recall threshold, providing a more global analysis of the classifier's performance[10].

In order to avoid oversampling techniques that would not reflect a realistic scenario, we decided to follow the four-to-one proportion earlier adopted by Jeronimo et al. [51] to execute the experiment. In his article, Silverman [101] indicates the presence of this proportion in the dissemination of news during the United States elections in 2016.

Classification Models Configuration

Among all models tested, the Random Forest (RF) and Bi-LSTM presented the best results concerning SL and DL models, respectively. The RF best model configuration was set with 103 trees in the forest, and the tree's maximum depth equals 20. The best Bi-LSTM model was set with 256 neurons, a learning rate of 5e-5, and 100 epochs.

5.2.2 Results and Discussion

This subsection presents and discuss the results obtained from this experiment. Table 5.1 presents the PR-AUC from all models experimented with, namely: best SL model trained with only the ALFs features, with only the D2V features, and with the combination of both of them; also the best DL model trained with only the ALFs, D2V or BERT features and

[10] Tables containing more evaluation metrics of each task are available in the Appendix.

with the combination of D2V and ALFs (D2V+ALF) and BERT and ALFs (BERT+ALF). No models in SL involving BERT were executed due to the large number of BERT features (768).

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
PR-AUC	0.62	0.49	0.65	0.78	0.73	0.81	0.83	**0.84**

Table 5.1: Fake News Detection in English PR-AUC results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

Aiming to discover if the difference between the results presented by two models is statistically significant, we proceeded with the McNemar statistical hypothesis test [70; 29] throughout this work. The McNemar is a test suitable for paired data situations, testing the consistency in responses across two variables. The McNemar H0 assumes that the two cases disagree with the same amount; in other words, there is no difference in the disagreement. In this experiment, H0 assumes that there is no difference between the hits and misses of the two models (compared to the ground truth). Therefore, the H1 hypothesis assumes that the difference exists, meaning that if a model presents a better result, its positive impact is statistically significant. All tests in this thesis have been performed considering p-value = 0.05.

In this scenario, the DL models obtain the best results (RQ4). The BERT model presents the best result among the models with no combination. However, the overall best model is trained with the combination BERT+ALF. The McNemar test result between BERT and BERT+ALF models rejects the H0 (p-value = 0.02), meaning that combining ALFs to BERT features is beneficial, in fact. Despite the results presented by BERT+ALF and D2V+ALF models being quite similar, the McNemar test proves that the BERT+ALF better result is statistically significant (p-value = 0.043). Thus, the BERT+ALF proves to be the best model in this scenario.

Comparing the results of all baselines alone and combined with ALFs (D2V and D2V+ALF on ML and DL, and BERT+ALF), the results combined with ALF enhanced those of baselines alone. Therefore, the ALFs combination positively impacts this scenario in all situations, affirmatively answering the RQ2. Also, we can notice that the ALFs alone present better results than the D2V alone both in SL and DL. The poor performance of D2V in this scenario could have been due to the low number of texts used to train the model compared to the number of texts used to train the pre-trained word embedding used to generate the ALFs. So, this scenario experiments answer the RQ1 affirmatively, referring to the D2V baseline, but negatively concerning BERT.

Thus, in the case of Fake News Detection in English scenario, the ALFs perform better than the D2V and improve the BERT result.

Aiming to analyse the ALFs individual impact on classification, we proceeded with a feature importance evaluation with ALFs using SHAP values [66] to verify what features most impact the classification tasks.

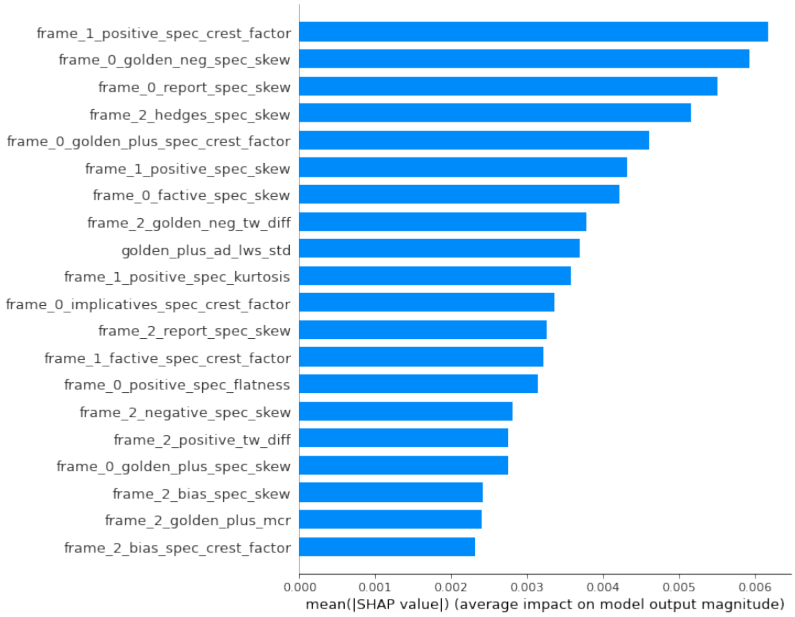

Figure 5.2: Feature Importance Bar Plot - Fake News Detection in English Classification.

Fig. 5.2 presents the twenty most impacting features in the Fake News Detection in English task in descent order (SHAP values plot). We notice that the Spectral Crest Factor (spec_crest_factor in the figure) and the Spectral Skewness (spec_skew) features are the most frequent ones, playing an essential role in the task. So, the flow spectral peakiness and spectral energy distribution around the mean helped discern fake from legitimate news.

The features of frame 2 (the final frame) are the most numerous, revealing that the information present in the ending part of the texts is significant to the task. The majority of features among the most impacting are extracted from the frequency domain, and, among the few ones of the time domain, the majority are flow-level features. 0.006 is the most significant average impact magnitude, highlighting that the classification results do not depend on one or a few features. Indeed, all features positively impact the results since no feature presents a 0.00 impact (not shown on the figure for clarity). These findings negatively answer RQ3.

Seeking to analyse the impact of single frame features (i.e., features extracted from a single frame) and flow features (features exclusively extracted from the flow), we performed the same experiment with the best model (DL-BERT+ALF) using these ALFs subsets separately (i.e., BERT+ALF Frame0 feats, BERT+ALF Frame1 feats, BERT+ALF Frame2 feats, BERT+ALF Flow feats). Table 5.2 presents the mentioned results. ALL feats column present the results using all ALFs to facilitate the comparison.

It is noticed that neither the single frame ALFs model nor the flows ALFs model could surpass the BERT+ALF all feats model (attested by the McNemar test - p-value in order of 0.027). This fact highlights that, in this scenario, the best result is achieved using all ALFs. In other words, the features extracted throughout the text are decisive to better distinguishing fake and legitimate news, also negatively answering RQ3. The Frame2 feats model presents a statistically significant bigger result than the other ALF subset models (p-value = 0.039), corroborating with the significant presence among the twenty more impactable features presented in Figure 5.2.

	Best Model - Deep Learning trained with BERT+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
PR-AUC	**0.84**	0.80	0.81	0.82	0.80

Table 5.2: Fake News Detection in English PR-AUC results. This table presents the results of the best model (DL-BERT+ALF) trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

5.3 Fake News Detection in Portuguese

The Fake News Detection in Portuguese task is presented and discussed in this subsection.

5.3.1 Experiment Description

Dataset

The Portuguese news dataset has 207,914 legitimate Brazilian news and 121 fact-checked fake news strongly disseminated in Brazil made available by Jeronimo et al. [51]. The dataset of legitimate news was collected from two of the biggest news sites in Brazil: Estadao[11] and Folha de Sao Paulo[12]. The dataset comprises legitimate news from 2014 to 2017, divided into different domains: Politics, Sports, Economy, and Culture. The fake news dataset was collected from more than 40 news sources strongly disseminated in Brazil from 2010 to 2017. All fake news were collected from two popular fact-checking services: e-Farsas[13] and Boatos[14].

Lexicons

This experiment uses the Reli-Lex, a sentiment polarity lexicon proposed by Freitas [35], Brazilian linguist. Created from book reviews written in Brazilian Portuguese, the lexicon consists of eight lists of words and expressions, described and exemplified next:

- Negative Adjectives (ADJ_NEG): adjectives related to negative emotions. For instance, ‘confused’ (*confuso*), ‘boring’ (*entediante*), ‘frustrating’ (*frustrante*).
- Positive Adjectives (ADJ_POS): adjectives related to positive emotions. Examples: ‘addictive’ (*viciante*), ‘unforgattable’ (*inesquecível*), ‘realistic’ (*realista*).
- Negative Expressions (MWE_NEG): Brazilian Portuguese colloquial multi-words expressions that refer to negative sentiments. For instance, ‘waste of time’ (*perda de tempo*), ‘so-so’ (*sem graça*).

[11] https://www.estadao.com.br/
[12] https://www.folha.uol.com.br/
[13] http://www.e-farsas.com/
[14] http://www.boatos.org/

- Positive Expression (MWE_POS): Brazilian Portuguese colloquial multi-words expressions that refer to positive sentiments. Examples: ‘love at first sight (*amor à primeira vista*), ‘be worth’ (*valer a pena*).
- Negative Nouns (SUB_NEG): nouns related to negative emotions. For instance, ‘cliche’ (*clichê*), ‘monotonous’ (*monótono*).
- Positive Nouns (SUB_POS): nouns related to positive emotions. For example, ‘ecstasy’ (*êxtase*), ‘favorite’ (*preferido*), ‘masterpiece’ (*obra-prima*).
- Negative Verbs (VER_NEG): a list of verbs that refer to negative sentiments, such as ‘to wear out’ (*cansar*), ‘to dissapoint’ (*decepcionar*), ‘to hate’ (*odiar*).
- Positive Verbs (VER_POS): a list of verbs that refer to positive sentiments, such as ‘to love’ (*amar*), ‘to delight’ (*encantar*), ‘to attract’ (*atrair*).

The Negative Adjectives, Adjective Positives, and Positive Verbs lists include some words that express sentiment exclusively in a cultural scope. In the present study, we omitted these words aiming to use a generalized scope lexicon.

It is worth mentioning that we also performed experiments on this scenario using only the Amorim et al. [8] subjectivity lexicons (described in Section 5.4.1) and the combination of these lexicons and the Reli-Lex. In the first case, the results showed to be worse; in the latter, the subjectivity lexicons could not improve the results presented in this thesis.

Evaluation Metric

As this scenario is also imbalanced, this evaluation uses the PR-AUC metric and four-to-one proportion described in section 5.2.1.

Classification Models Configuration

The best SL and DL models, respectively, were the RF with 100 trees in the forest, and the tree’s maximum depth equals 18, and the Bi-LSTM model set with 128 neurons, a learning rate of 5e-5, and 100 epochs.

5.3.2 Results and Discussion

This subsection presents and discusses the results obtained from the Fake News Detection in Portuguese experiment. Table 5.3 presents all PR-AUC achieved.

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
PR-AUC	**0.98**	0.46	0.98	0.96	0.90	0.88	0.96	0.96

Table 5.3: Fake News Detection in Portuguese PR-AUC results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

The ALFs alone on SL obtained the best efficacy in this scenario, affirmatively answering RQ1. The McNemar test only does not reject the H0 comparing ALF and D2V+ALF on SL models (p-value = 3.51). Even in this case, using uniquely the ALFs is beneficial considering a better performance, given that it would not depend on D2V model training and many unnecessary features. Confirming the best efficacy of ALFs alone on SL compared to DL, the McNemar test presented a p-value = 0.004, responding to RQ4.

Comparing the results of ALFs to baselines combined with ALFs makes it noticeable that the baselines do not positively or negatively impact the ALFs results. It seems like the classifiers ignore the D2V and BERT features. This fact evidences that our set of features is very robust in this scenario. However, comparing the baselines alone to their association with ALFs, the ALFs association positively impacts the efficacy, affirmatively responding to RQ2.

Unlike the Fake News Detection in English scenario, in this one, D2V achieves better efficacy than BERT (both alone).

Figure 5.3 shows the most impacting features in the Fake News Detection in Portuguese task. The most frequent features are Jitter (7/20) and Volume - vol (4/20), but Jitter is the first three most important. These are two different measures of variation. Although the most beneficial feature is Jitter, representing the set of flow extracted features, it is worth remembering that it is calculated over the frame feature Pitch.

Both Frame 0 and Frame 1 (representing the initial and middle part of the text, respectively) are very present among the most impacting features. Figure 5.3 shows ten time-

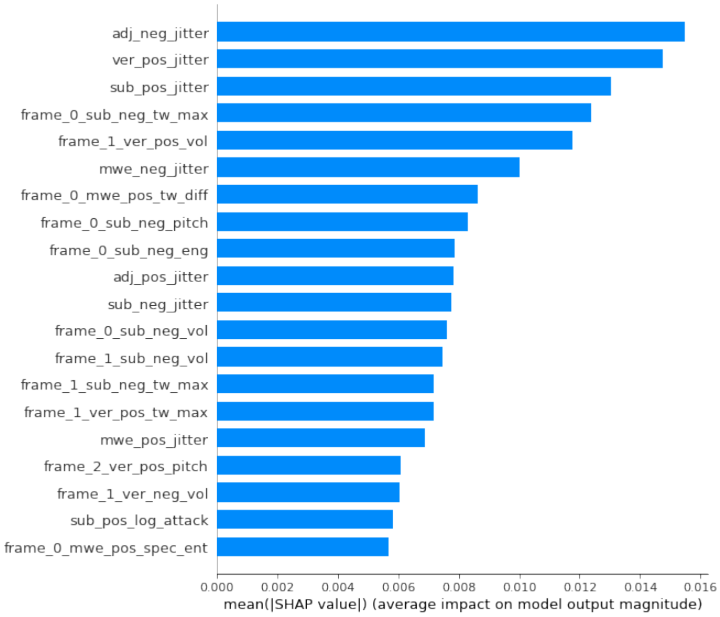

Figure 5.3: Feature Importance Bar Plot - Fake News Detection in Portuguese Classification.

domain and ten frequency-domain features, a balanced scenario in this case. The most significant average impact magnitude is 0.016, yet a little value, suggesting that the result is not owed to a small group of features.

Table 5.4 presents the best model (SL - ALFs alone) considering the frames and flow subsets of features. The ALL feats model only surpasses the Flow feats model (with statistical significance, p-value = 0.009), contradicting the SHAP feature importance plot. This fact could evince that even when very important in the classification, the flow features alone are not a meaningful enough text representation. The results also show the same efficiency on using all ALFs or any frames subset, showing that, in this scenario, the extracted features of any frame (representing distinct portions of the texts) are meaningful enough to differentiate fake and legitimate news. If only this scenario were considered, the response to RQ3 would be affirmative.

5.4 Newspaper Columns versus News Classification

A *newspaper column* is a recurring feature written by the same author in a newspaper. It is an opinionated text frequently defined by the voice and personality of the writer, in opposition

	Best Model - Machine Learning trained with ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
PR-AUC	**0.98**	0.98	0.98	0.98	0.94

Table 5.4: Fake News Detection in Portuguese PR-AUC results. This table presents the results of the best model (SL - ALF) trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

to objective news that reports facts.

5.4.1 Experiment Description

Dataset

On the Newspaper Columns versus News Classification task, we employed the Jeronimo et al. [51] legitimate news to represent the objective news. The newspaper columns dataset was firstly presented in our work [105], being formed by 7,062 newspaper columns articles.

Lexicons

Besides the Reli-Lex (described on Section 5.3.1), this experiment also applied the the Amorim et al. [8] subjectivity lexicons. Brazilian linguists build these subjectivity lexicons. The lexicons depict subjectivity in five different dimensions described next:

- The argumentation dimension (ARG) represents words and expressions related to a more argumentative discourse. Such discourse is often used when someone tries to convince another person of a specific point of view. Examples of markers present in this lexicon: ‘as if’ (*como se*), ‘rather than’ (*em vez de*), ‘somehow’ (*de certa forma*), ‘despite’ (*apesar de*).
- The presupposition dimension (PRE) encompasses terms related to a previous assumption of something. This kind of discourse is mainly used when the interlocutor assumes something is true, even when this is not the case. Examples of words belonging to this

lexicon are: ‘nowadays’ (*hoje em dia*), ‘to keep on doing’ (*continuar a*), and factive verbs.

- The sentiment dimension (SEN) contains words and terms related to overall emotional discourse. This lexicon has no polarity division. Some examples are: ‘with regret’ (*infelizmente*), ‘fortunately’ (*felizmente*), and ‘it is preferable’ (preferencialmente).
- The valuation dimension (VAL) expresses the amount or intensification of something, such as: ‘absolutely’ (*absolutamente*), ‘highly’ (*altamente*), and ‘approximately’ (*aproximadamente*).
- The modalization discourse (MOD) is used when the writer exhibits a stance towards its statement. Such markers are adverbs, auxiliary verbs, and modality clauses, for instance: ‘advise’ (*aconselhar*), ‘undoubted’ (*indubitável*), ‘presume’ (*presumir*)’ and ‘suppose’ (*supor*).

Experiments in this scenario were also performed using only the Amorim et al. [8] subjectivity lexicons or the Reli-Lex. In both cases, the results showed to be worse than using the combined lexicons.

Evaluation Metric

Due to the imbalance, this scenario maintains the PR-AUC metric and four-to-one proportion described in Section 5.2.1.

Classification Models Configuration

The best SL and DL models, respectively, were the XGB with a learning rate of 0.1, and the tree’s maximum depth equals 6, and the Bi-LSTM model set with 128 neurons, a learning rate of 5e-5, and 100 epochs.

5.4.2 Results and Discussion

Table5.5 shows the results obtained in this scenario. The DL - D2V+ALF model achieved better effectiveness among all models. The combination with ALFs could enhance the already impressive D2V efficacy (with statistical significance, p-value = 0.001). The difference in the results of DL - D2V+ALF and SL - D2V+ALF is also statistically significant (p-value = 0.006), meaning that the DL classifier improved the effectiveness of the D2V+ALF set of features in this scenario. So, the RQ2 is affirmatively answered.

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
PR-AUC	0.94	0.85	0.95	0.92	0.94	0.93	**0.97**	0.94

Table 5.5: Newspaper Columns versus News PR-AUC results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

The SL model trained with only the ALFs shows similar efficacy compared to the baselines alone. The SL - ALF model reaches the same effectiveness as DL - D2V and DL - BERT+ALF, showing that the ALFs are robust in this scenario, even using a simpler classification algorithm. The ALF presents better effectiveness than the D2V model considering the SL models. The improvement shown on the D2V+ALF result is statistically insignificant, reinforcing the robustness of ALF associated with more uncomplicated algorithms. These findings respond to RQ4 and affirmatively reply the RQ1.

Responding to RQ3, we can observe the most impacting features on Newspaper Columns versus News task model presented by Figure 5.4. The most frequent features are Text Waveform Minimum (tw_min) and Energy. Features measuring the amplitude variation and the flow shape.

Frame 0 Negative Verbs Energy (frame_0_ver_neg_eng) has an expressive impact as it achieves an average of 0.1, suggesting the strength of negative verbs lexicon dimension presented by the initial part of the text was essential for the performance of the classifiers. In other words, the presence of the semantics of the negative verbs dimension is notably diverse in newspaper columns and news. In this scenario, the average impact magnitudes are more significant than the other tasks, i.e., fewer features have more power. The initial part of the text (Frame 0) seems to contribute expressively to the classification task, evincing that the two kinds of texts are quite different at the beginning. The time-domain features are expressively represented among these twenty most impacting features, revealing that, in this scenario, the analysis is better on the simpler domain. The most frequent features are Energy

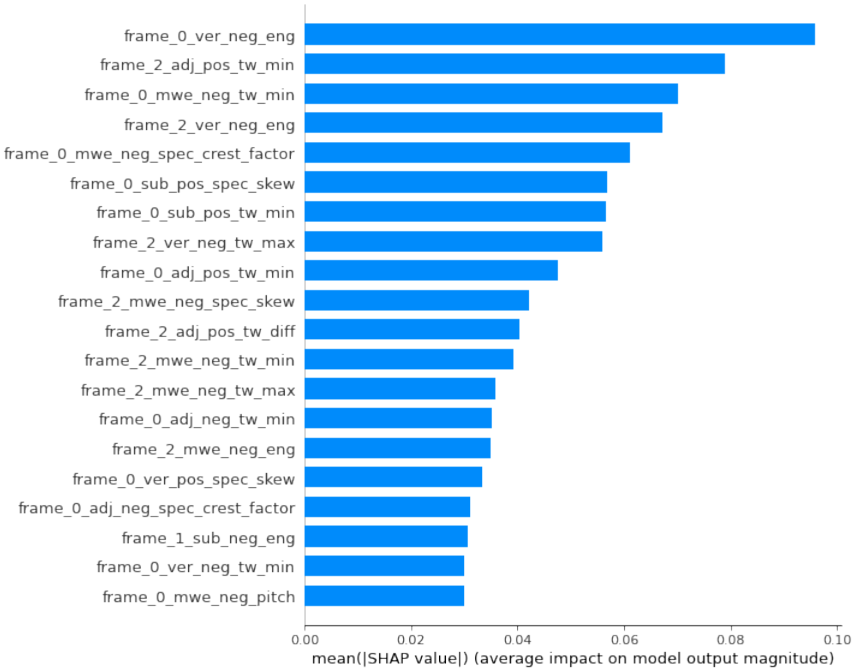

Figure 5.4: Feature Importance Bar Plot - Newspaper Columns versus News Classification.

TW$_{M}inandTW_{M}ax, presentingthatthelexicondimensionstrengthandtheclosestandthefurthestpoin$
$levelfeaturespermeatethefeaturesshowninFigure5.4, whichsuggeststhattheanalysisismoreeffec$
$Lexachievesworseresults.$

Table 5.6 presents the best model (DL - D2V+ALF) considering the frames and flow subsets of features. The Flow feats model surpasses the ALL feats model with statistical significance (p-value = 0.03). Both ALL feats and Flow feats models present statistical significance when comparing the results to each Frame feats model (p-value in order of 0.037). However, the difference is relatively small, making the Frame feats not being the best text representation but a good enough one in this scenario.

5.5 Movie Reviews Sentiment Polarity Classification in English

Opinionated information is widely available online and plays a vital role in evaluating whether a product or service is pleasing its consumers or not. In this context, sentiment analysis of product or service reviews is a commonly exploited field since it focuses on the

	Best Model - Deep Learning trained with D2V+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
PR-AUC	0.97	0.96	0.96	0.96	**0.98**

Table 5.6: Newspaper Columns versus News PR-AUC results. This table presents the results of the best model trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

classification of sentiments or opinions expressed in human-generated texts [10].

In order to evaluate our proposed approach efficacy in this context, we performed a sentiment polarity classification on movie reviews written in English.

5.5.1 Experiment Description

Dataset

The IMDB dataset [67] is a movie review dataset widely used in machine learning evaluations. It contains 25,000 positive and 25,000 negative polarities annotated reviews from the review site that gives the name[15]. It also contains 50,000 unlabeled reviews used in this experiment to train the D2V model.

Lexicons

Bing Liu lexicon [48] is used in this experiment. It is a widely used lexicon on sentiment analysis [10]. It consists of 2,006 positive polarity terms and 4,783 negative polarity terms. This lexicon contains frequent sentimental words, misspelled words, slang words, and common variants.

It is worth highlighting that we also performed this experiment using the lexicons described in Section 5.6.1, but the results were worse than using Bing Liu lexicons.

[15] http://www.imdb.com/

Evaluation Metric

Since this scenario is perfectly balanced, it is evaluated by the Area Under Receiver Operating Characteristic Curve (ROC-AUC). As discussed on [93; 25] papers, the ROC-AUC is an appropriate metric for balanced scenarios, reflecting the trade-off between True Positive Rate and False Positive Rate at different rating thresholds.

Classification Models Configuration

The best SL and DL models, respectively, were the XGB with a learning rate of 0.1, and the tree's maximum depth equals 4; and a three-layers Feed-Forward Network (FFN) model, each layer set with 256 neurons, a learning rate of 5e-5, and 100 epochs.

5.5.2 Results and Discussion

Table 5.7 presents the ROC-AUC results obtained in this scenario. The DL - BERT model obtains the best result (the same result presented by Alaparthi and Mishra [5]). The ALFs present a poor result, mainly the DL - ALF that achieves almost a random guessing classifier result (SL - ALF obtained a better result, responding to RQ4). The minor improvement of SL - D2V+ALF over SL - D2V is not statistically significant (p-value = 2.15). The DL - D2V+ALF model achieved the same result as DL - D2V. The ALFs combined with the BERT features worsen the DL - BERT model results, confusing the classifier. Therefore, ALFs are not beneficial or adequate in this scenario (negatively answering RQ1 and RQ2).

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
ROC-AUC	0.74	0.79	0.80	0.52	0.85	**0.92**	0.85	0.88

Table 5.7: Movie Reviews Sentiment Polarity Classification in English ROC-AUC results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

Figure 5.5 presents the twenty more impacting ALFs in this scenario, helping to understand why ALFs involve such poor results. The most significant impact magnitude is really high (0.25), and the first eighteen features concentrate a substantial amount of impact, presenting 0.1 at least. This fact is confirmed by the impact of the two last features shown, near 0.00. Thus, few features concentrate almost all the impact on the model; in other words, only these eighteen features aggregate valuable information to the classification. These conclusions suggest that the method performs poorly when only a few features are considered in classification, highlighting the importance of all twenty-five features (RQ3).

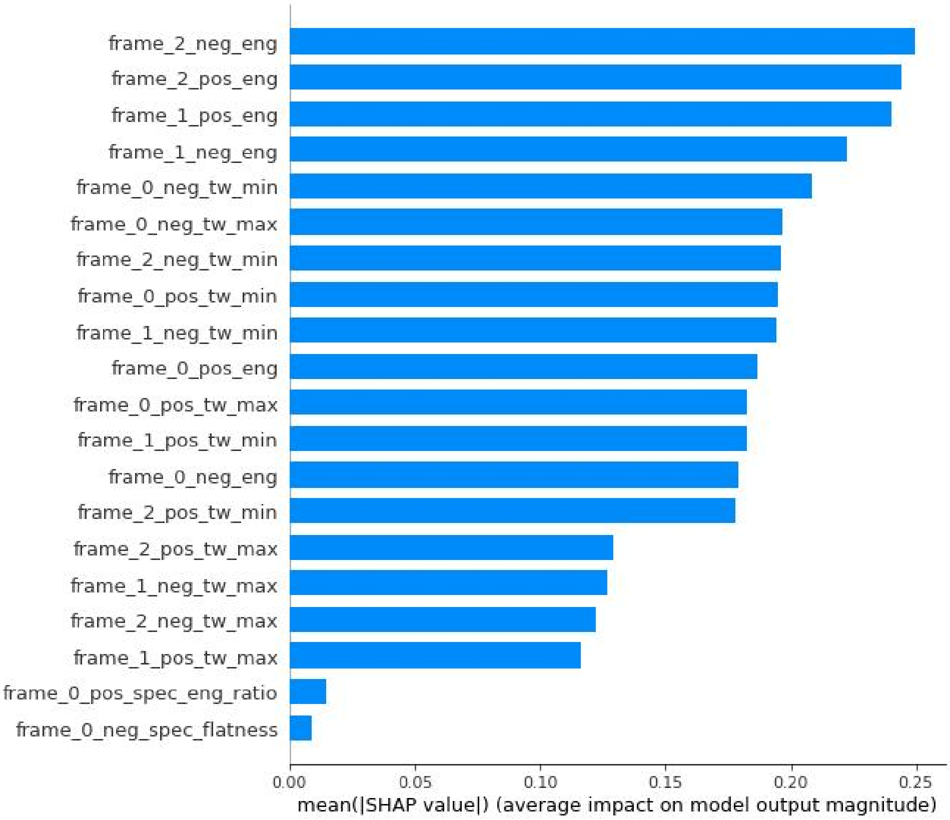

Figure 5.5: Feature Importance Bar Plot - Movie Reviews Sentiment Polarity Classification in English.

Only the two less impacting features are from the frequency domain. Moreover, the eighteen time-domain features are all Energy (eng), Text “Waveform” Min, and Max (twmin and twmax). These features are directly related to the flow amplitude and do not aggregate enough meaningful information to help distinguish between the two types of reviews. Other more complex features did not even have a minimum impact on the model. In the Newspaper Column versus News scenario, the Energy, twmin, and twmax also frequently appeared among the most impacting; however, presenting a much lower impact average, thus, the ALFs obtained good results.

It is also worth perceiving that the eighteen time-domain features are the positive and negative versions of eng, twmin, and twmax of all frames. It suggests that the shapes of positive and negative review flows are like mirrors in this scenario. For example, a positive review presents a negative lexicon flow quite similar to the positive lexicon flow of a negative review. This behaviour prevents the approach from extracting adequate information to discern the two text types in this case.

Another factor that might have led the method not to achieve a better result is related to the BingLiu's lexicon dimensions. This lexicon has only two dimensions (positive and negative) encompassing general terms – nouns, verbs, adjectives, etc, leading the method to generate only two quite generic flows per text. The Reli-lex appeared to be more suitable to be used by the proposed approach (as can be seen in all Portuguese experiments). It separates the positive and negative words into part-of-speech classes (noun, adjectives, etc), making the method generate more numerous and specific (consequently more meaningful) flows.

	Best Model - Deep Learning trained with BERT+ALF					
Features	Bert feats	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
ROC-AUC	**0.92**	0.88	0.83	0.84	0.84	0.79

Table 5.8: Movie Reviews Sentiment Polarity Classification in English ROC-AUC results. This table presents the results of the best model trained with BERT feats, BERT+all ALFs (All Feats), with the BERT+ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only BERT+ALFs calculated over the entire flow (Flow feats).

The use of the Google News word embeddings may also have contributed negatively to the results in this scenario, as the texts involved here are reviews.

We also experimented with the best model involving the ALFs with the single frames and flow features. The results can be found in Table5.8. The table shows a new column with the DL - BERT model result to simplify the comparison. As expected, the results could not surpass the BERT result or even the ALL feats model. These results confirm that the mirror behaviour is present throughout the flows. Otherwise, one of the frame models could achieve, at least, an approximate or better result than the ALL feats models.

5.6 Book Reviews Sentiment Polarity Classification in Portuguese

The Book Reviews Sentiment Polarity Classification in Portuguese task is presented and discussed in this subsection.

5.6.1 Experiment Description

Dataset

The dataset used in this task contains 630,665 book reviews[16] from the Skoob[17]. Skoob is a Brazilian social network for readers, which has a rich space of reviews made by users. The reviews are originally rated between 0 and 5 stars. The dataset presents 52,845 0-star-rating reviews, 15,788 1-star-rating reviews, 30,274 2-star-rating reviews, 93,597 3-star-rating reviews, 157,092 4-star-rating reviews, and 281,069 5-star-rating reviews. We considered the 0 to 2-star rated reviews as negative reviews (98,907 in total) and the 3 to 5-star rated reviews like positive reviews (531,758 in total).

The book reviews dataset is considerably vast. Inspired by the IMDb dataset, we have randomly chosen 25,000 positive and negative reviews, maintaining the original proportion of the rating stars in each set.

Lexicons

This experiment uses the combination of the Reli-lex and the subjectivity lexicons in Portuguese. Experiments in this scenario were also performed using only subjectivity lexicons or the Reli-Lex. In both cases, the results showed to be worse than using the combined lexicons.

[16] Available on https://gdarruda.github.io/2019/07/27/corpus-skoob.html

[17] https://www.skoob.com.br/

Evaluation Metric

Due to the classes balance, this scenario is also evaluated by the ROC-AUC metric.

Classification Models Configuration

The best SL model is the XGB with a learning rate of 0.1, and the tree's maximum depth equals 6. Same as the movie reviews in English scenario, the DL best model is a three-layers FFN model, each layer set with 256 neurons, a learning rate of 5e-5, and 100 epochs.

5.6.2 Results and Discussion

Table5.9 presents the results of the Book Reviews in Portuguese experiments. The best model is the DL - D2V+ALF. The difference between the DL - D2V+ALF and DL - D2V results is statistically significant (p-value = 0.006), meaning that the combination with ALFs is beneficial to the classification task efficacy. The combination of ALF also enhances the D2V result on SL, but not the BERT result (responding RQ2 affirmatively to D2V and negatively to BERT). Alone, ALFs achieve poor results on SL and DL (RQ4) in this scenario, negatively answering RQ1.

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
ROC-AUC	0.57	0.63	0.69	0.56	0.71	0.61	**0.79**	0.58

Table 5.9: Book Reviews Sentiment Polarity Classification in Portuguese ROC-AUC results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

Concerning replying to RQ3, Figure5.6 shows the SHAP results. Like the Movie Reviews scenario, the most frequent features are the time domain eng, twmin, and twmax. However, the bigger average impact magnitude is 0.07, indicating that the impact is distributed over a more extensive set of features. Features extracted from flows of both lexicons (subjectivity and Reli-Lex) permeate the most impacting ones shown in the figure, suggesting that both lexicons are significant to the result. The twenty most impacting features are almost balanced distributed on frames 0, 1, and 2, but no flow-level features are present.

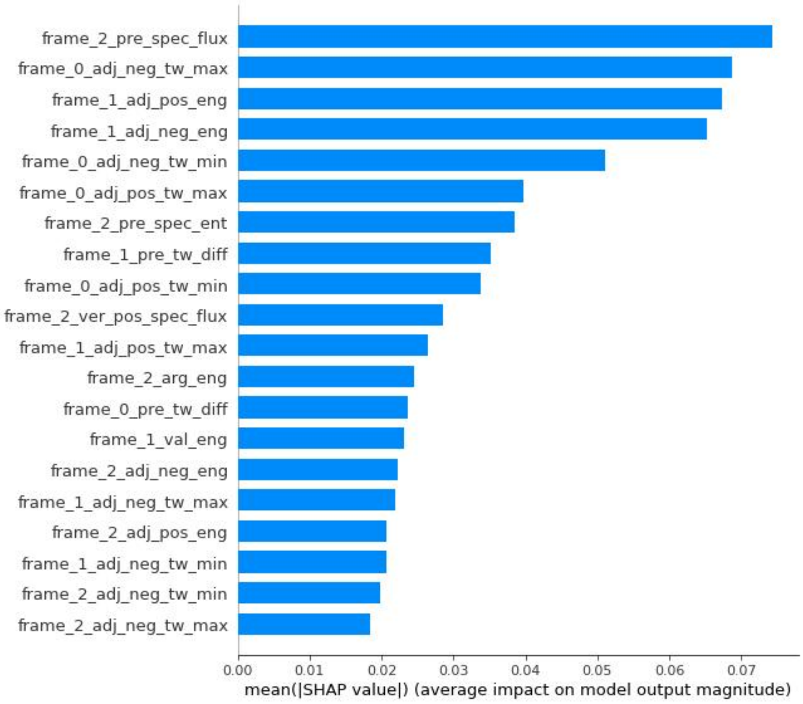

Figure 5.6: Feature Importance Bar Plot - Book Reviews Sentiment Polarity Classification in Portuguese.

Table5.10 presents the best model (DL - D2V+ALF) considering the frames and flow subsets of features. The ALL feats model only surpasses the Flow feats model (with statistical significance, p-value = 0.002), corroborating the SHAP feature importance plot.

The difference between ALL feats and Frame 0 models results is statistically significant (p-value = 0.047). However, the results show the same efficiency on using all ALFs or the Frame 1 or Frame2 subsets, showing that, in this scenario, the extracted features of these two frames are meaningful enough to differentiate positive and negative book reviews (RQ3).

5.7 Final Considerations

This chapter presented the experimental evaluation to assess the approach's effectiveness. Five NLP classification tasks were performed. For comparison purposes, we also performed the classification tasks using D2V and BERT baselines and combined each baseline with ALFs. Moreover, experiments using only individual frame features or flow-level features

	Best Model - Deep Learning trained with D2V+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
ROC-AUC	**0.79**	0.78	0.79	0.79	0.75

Table 5.10: Book Reviews Sentiment Polarity Classification in Portuguese ROC-AUC results. This table presents the results of the best model trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

were executed to estimate the effectiveness of these features subsets.

The experiments results showed that ALFs present the best results alone or combined to the baselines in most scenarios (4/5). These considerations indicate that our model is a valuable way of representing texts, extracting relevant information that can help to improve efficacy in classification tasks, affirmatively answering to RQ2.

Our method achieved the best results in Fake News Detection in Portuguese, affirmatively answering RQ1 in this task. Using ALFs combined with the baselines achieved the same result as the ALFs alone. In other words, adding the baselines to the classification does not enhance the ALFs results in this scenario.

Considering the Fake News Detection in English, Newspaper Columns versus News and Book Reviews experiments, the best results were achieved by combining baselines with ALFs, highlighting that our approach can improve strong baselines effectiveness. In the Fake News Detection in English and Newspaper Columns versus News scenarios, the proposed method approximated the baselines' results, even surpassing D2V in some cases. These facts affirmatively answer RQ2.

Considering the experiments in which only individual frame features or flow-level features were used, in the Fake News Detection in English, employing ALL feats remained with the best result. In the Fake News Detection in Portuguese, the results show the same efficiency on using all ALFs or any frames subset. On the other hand, in the Newspaper Columns versus News task, the flow-level features presented a slightly better result than ALL feats. Finally, in the Book Reviews task, using Frame 1 or Frame 2 features had the same efficiency as using ALL feats. Combined with the SHAP plots analysis, these considerations highlight that no single subset of ALFs (neither by frame or flow nor by feature type) helps the classifier achieve the best results in all tasks. Also, no feature is useless that could be discarded from the method. Additionally, no individual feature (or a subset of features) significantly impacted all tasks, suggesting that all features are vital to the classification. Since the text flow shape depends on the WMD to the underlying lexicon (both text and lexicons represented by word embeddings), the flows are considerably different from one task to another. Thus, the analysis varies according to the flow format, evincing a different group of features to each task. Therefore, a feature selection would only be effective if applied to each task. The feature importance also unveiled that the frame divisions play a fundamental role in the tasks, even revealing the texts' portion that is more capable of differentiating the texts classes through the ALFs. All these considerations bring a negative answer to the RQ3.

The approach poorly performed in the Movie Reviews task. The ALFs neither produced a good result nor enhanced the baseline's efficacy. The results in this scenario brought an negative answer to RQ1 and RQ2. The method's poor performance when only a few features were considered on classification contributed to this bad result. Also, creating the flows based on the Bing Liu's lexicon and extracting ALFs do not generated meaningful enough information to distinguish between the two review types. These findings evince the approach's dependency on consistent and adequate lexicons to task (or to texts on the dataset) to achieve satisfactory efficacy. As discussed in Taboada et al. [11] and Araque et al. [9], this is a frequent limitation among methods that rely on lexicons.

The approach performed better when feeding SL models in all but Fake News Detection in English task, answering the RQ4. This finding evinces that a sophisticated DL model not always performs better than a simpler SL model.

We highlight the results of the tasks involving news and the Portuguese language. In addition to the outstanding result obtained by ALFs on Fake News Detection in Portuguese task, on the Newspaper Columns versus News scenario, the combination with ALFs enhanced the impressive D2V efficacy. These facts suggest that the extracted information by our approach was especially beneficial in these scenarios. Besides the manner ALFs are extracted, we can attribute this result to the adequacy of the lexicons used to differentiate the related kinds of texts.

Although BERT is one of the most powerful NLP tools, only in the English language scenarios it obtained the best results (alone or combined with ALFs). This fact suggests that the multilingual model is not as accurate as the English model, as discussed by Gahbiche et al.[14], emphasising the difficulty of doing NLP research in other languages. Another possible reason could be the inviability of considering all texts terms when using BERT (a model would have to consider the larger text on the dataset). As discussed in Aker et al. [4], Ghanem et al.[37], and Maharjan et al. [68] papers, using the entire text is crucial to better classifying it.

Chapter 6

Final Considerations

Text classification is one of the mainly investigated challenges in Natural Language Processing (NLP) research. The higher performance of a classification model depends on a representation that can extract valuable common information about the texts.

Some scenarios may demand a semantically more elaborate text representation to enhance the classification model performance since semantics is a powerful tool to recognize different contexts even when a similar vocabulary is used. Recent research has been associating the power of the word embeddings representation (semantics for a general context) with the additional information that lexicons promote to achieve a more accurate text representation model.

The main objective of this doctoral research was to propose a method of improving text classification efficacy by enhancing text representations with semantics.

This thesis' main contribution is the proposed approach for representing texts by flows that incorporate lexicon information and then extracts features inspired by audio analysis from them. In an unprecedented way, the approach combines different techniques to enhance text representation to improve classification efficacy.

The proposed approach represents texts as flows by calculating WMD from each sentence to a lexicon on an embedding vector space, composing the text flow. The extracted features are very innovative on NLP, inspired by well-established audio analysis features.

For evaluating the method's efficacy, we performed five NLP classification tasks. For comparison purposes, we also performed the classification tasks using the strongs D2V and BERT baselines and combined each baseline with ALFs. Moreover, we performed experiments on each task using single frames and flow-level features on its best model.

The experimental evaluation demonstrated that the approach could enhance the baseline methods text classification efficacy in most scenarios. In the Fake News Detection in Portuguese task, the proposed method performed undoubtedly better than the baselines. It approximated the baselines' results in the remaining tasks (except Movie Reviews), even surpassing D2V in the Fake News Detection in English. Movie Reviews was the unique experiment in which the method performed inadequately.

The experiments involving the single frames and flow-level features highlighted that, depending on the task, these could be an adequate subset of features to be used in classification. The SHAP analysis suggested that no feature (type) could be definitively excluded from the method.The cause can be the dependence on the different embeddings and lexicons that generate quite different text flows shapes (between tasks), impacting the analysis.

Considering a particular language, suppose the lexicons and word embeddings limitations are overcome. The method has the potential to present good effectiveness in classification tasks in that language.

6.1 Limitations

As evinced in Section 5.5.2, our proposed approach depends on a lexicon that is adequate to the task and suitable enough for meaningful ALFs extraction to achieve satisfactory efficacy. The Movie Reviews experiment was essential to highlight this limitation empirically, confirming the discussion brought by Taboada et al. [11] and Araque et al. [9].

In addition to the lexicons, the method is also dependent on adequate and well-trained word-embeddings. Suitable lexicons and word-embeddings are essential to the quality of the text flows and, consequently, to the quality of ALFs. We used general pre-trained word embeddings in the experimentation; however, the results could be even better if we had built more specific word embeddings for each task.

Another limitation of our approach is not being suitable for tiny (e.g., microblogs) texts. As the method uses sentences as units to create the flows, tiny texts would provide an insufficient number of sentences for the correct ALFs extraction. This fact would force intense padding, possibly worsening the ALFs' quality. At the beginning of this research, we experimented with a very preliminary version of the method to classify the sentiment polarity of a product reviews dataset (made available by Wachsmuth and Stein [107]). The texts on this dataset presented less than four sentences, on average. The results evinced the poor efficacy presented by our approach to these conditions.

An experimental limitation presented by this study is the lack of performing tests. In cases where two scenario configurations presented similar efficacy, performing tests could highlight that the small efficacy gain would not offset a longer time or hardware consumption.

6.2 Future Work

The present work opens up several possibilities as future work, for instance:

- To apply the proposed approach to other NLP tasks using more extensive texts, like full-length research articles, book chapters, and books. This suggestion could use more frames and consider more sentences per frame, evaluating the impact on the ALFs capability of aggregating valuable information to the text analysis.
- If the application of the method to more extensive texts is successful, one option is to implement the features on time and frequency domains that require a more significant number of samples. Another possibility is extending the feature extraction to other (audio analysis) domains that also demand a more extensive number of samples.
- To evaluate if a normalization procedure like the proposed by Mao and Lebanon [69] to soften the flows generate more valuable ALFs.
- To evaluate if other similarity metrics like Cosine or Manhattan distances could produce more valuable ALFs.
- To adapt the approach to incorporate part-of-speech (POS) identification and assess if modeling POS could enhance the approach's efficacy.
- To evaluate the impact of using an attention mechanism based on the individual frame word embeddings and their respective ALFs on classification effectiveness.

- To propose an approach to combine this thesis' method to speech analysis, aiming to analyse the transcription of the considered speeches and enhancing the speeches contents analysis. For example, the combined approach could be used to analyse the speech and transcription of call center phone callings to evaluate customer satisfaction.

Bibliographical References

[1] Charu C. Aggarwal. *Machine Learning for Text*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[2] Charu C. Aggarwal and Cheng Xiang Zhai. *Mining Text Data*. Springer Publishing Company, Incorporated, 2012.

[3] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. Association for Computational Linguistics.

[4] Ahmet Aker, Hauke Gravenkamp, Sabrina Mayer, Marius Hamacher, Anne Smets, Alicia Nti, Johannes Erdmann, Julia Serong, Anna Welpinghus, and Francesco Marchi. Corpus of news articles annotated with article level subjectivity. 06 2019.

[5] Shivaji Alaparthi and Manit Mishra. Bidirectional encoder representations from transformers (BERT): A sentiment analysis odyssey. *CoRR*, abs/2007.01127, 2020.

[6] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Working Paper 23089, National Bureau of Economic Research, January 2017.

[7] Francesc Alías, Joan Claudi Socoró, and Xavier Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6(5), 2016.

[8] Evelin Amorim, Marcia Cançado, and Adriano Veloso. Automated essay scoring in the presence of biased ratings. In *Proceedings of the 2018 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 229–237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[9] Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sanchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246, 2017.

[10] Oscar Araque, Ganggao Zhu, and Carlos A. Iglesias. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, 165:346 – 359, 2019.

[11] Fatemeh Torabi Asr and Maite Taboada. Big data and quality data for fake news and misinformation detection. *Big Data & Society*, 6(1):2053951719843310, 2019.

[12] Lucas Vinicius Avanço and Maria das Graças Volpe Nunes. Lexicon-based sentiment analysis for reviews of products in brazilian portuguese. In *2014 Brazilian Conference on Intelligent Systems*, pages 277–281, 2014.

[13] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[14] Gaétan Baert, Souhir Gahbiche, Guillaume Gadek, and Alexandre Pauchet. Arabizi language models for sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 592–603, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[15] Lingxian Bao, Patrik Lambert, and Toni Badia. Attention and lexicon regularized LSTM for aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 253–259, Florence, Italy, July 2019. Association for Computational Linguistics.

[16] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[17] Mrinmoy Bhattacharjee, S. R. Mahadeva Prasanna, and Prithwijit Guha. Time-frequency audio features for speech-music classification. *ArXiv*, abs/1811.01222, 2018.

[18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[19] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, oct 2001.

[20] Jingjing Cai, Jianping Li, Wei Li, and Ji Wang. Deeplearning model used in text classification. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 123–126, 2018.

[21] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36 – 64, 2016.

[22] Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. *Affective Computing and Sentiment Analysis*, pages 1–10. Springer International Publishing, Cham, 2017.

[23] Yong Duk Cho, Moo Young Kim, and Sang Ryong Kim. A spectrally mixed excitation (smx) vocoder with robust parameter determination. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 2, pages 601–604 vol.2, 1998.

[24] Yoonjung Choi and Janyce Wiebe. +/-EffectWordNet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191, Doha, Qatar, October 2014. Association for Computational Linguistics.

[25] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[26] Li Deng and Yang Liu. *Deep Learning in Natural Language Processing*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[27] Sunipa Dev, Tao Li, Jeff M. Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7659–7666, Apr. 2020.

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

[29] Thomas G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923, 10 1998.

[30] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. Speech/music discrimination for multimedia applications. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 4, pages 2445–2448 vol.4, 2000.

[31] Mireia Farrús, Javier Hernando, and Pascual Ejarque. Jitter and shimmer measurements for speaker recognition. pages 778–781, 08 2007.

[32] Ronen Feldman and James Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[33] Elena Filatova. Sarcasm detection using sentiment flow shifts. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 264–269, 2017.

[34] International Organization for Standardization (ISO)/International Organization for Standardization (IEC). *Information Technology - Multimedia Content Description Interface - part 4: Audio.* ISO/IEC, Moving Pictures Expert Group, 1st edition, 2002.

[35] Cláudia Freitas. Sobre a construção de um léxico da afetividade para o processamento computacional do português. In *Rev. bras. linguist. apl.*, volume 13, pages 1031–1059, 2013.

[36] Xianghua Fu, Jingying Yang, Jianqiang Li, Min Fang, and Huihui Wang. Lexicon-enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 6:71884–71891, 2018.

[37] Bilal Ghanem, Simone Paolo Ponzetto, Paolo Rosso, and Francisco Rangel. FakeFlow: Fake news detection by modeling the flow of affective information. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 679–689, Online, April 2021. Association for Computational Linguistics.

[38] George Giannakopoulos, Petra Mavridi, Georgios Paliouras, George Papadakis, and Konstantinos Tserpes. Representation models for text classification: A comparative analysis over three web document types. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, New York, NY, USA, 2012. Association for Computing Machinery.

[39] Theodoros Giannakopoulos and Aggelos Pikrakis. Chapter 1 - introduction. In Theodoros Giannakopoulos and Aggelos Pikrakis, editors, *Introduction to Audio Analysis*, pages 3 – 8. Academic Press, Oxford, 2014.

[40] Theodoros Giannakopoulos and Aggelos Pikrakis. Chapter 4 - audio features. In Theodoros Giannakopoulos and Aggelos Pikrakis, editors, *Introduction to Audio Analysis*, pages 59 – 103. Academic Press, Oxford, 2014.

[41] Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan and Claypool Publishers, 2017.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[43] Louise Guthrie, James Pustejovsky, Yorick Wilks, and Brian M. Slator. The role of lexicons in natural language processing. *Commun. ACM*, 39(1):63–72, jan 1996.

[44] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[45] Patrick Helmholz, Michael Meyer, and Susanne Robra-Bissantz. Feel the moosic: Emotion-based music selection and recommendation. 06 2019.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[47] Benjamin D. Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news, 2017.

[48] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA, 2004. Association for Computing Machinery.

[49] Adit Jamdar, Jessica Abraham, Karishma Khanna, and Rahul Dubey. Emotion analysis of songs based on lyrical and audio features. *International Journal of Artificial Intelligence & Applications*, 6, 06 2015.

[50] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang, and Jong Kim. Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, 10:5841, 08 2020.

[51] Caio Jeronimo, Claudio Campelo, Leandro Marinho, Allan Sales, Adriano Veloso, and Roberta Viola. Computing with subjectivity lexicons. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3272–3280, Marseille, France, May 2020. European Language Resources Association.

[52] Caio Jeronimo, Leandro Marinho, Claudio Campelo, Adriano Veloso, and Allan Melo. Fake news classification based on subjective language. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, 2019.

[53] Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. Bag-of-embeddings for text classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2824–2830. AAAI Press, 2016.

[54] Edson C. Tandoc Jr., Zheng Wei Lim, and Richard Ling. Defining "fake news". *Digital Journalism*, 6(2):137–153, 2018.

[55] Lakshmish Kaushik, Abhijeet Sangwan, and John Hansen. Sentiment extraction from natural audio streams. 05 2013.

[56] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. SpringerLink : Bücher. Springer New York, 2013.

[57] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org, 2015.

[58] Caio L. M. Jeronimo, Claudio E. C. Campelo, Leandro Balby Marinho, Allan Sales, Adriano Veloso, and Roberta Viola. Computing with subjectivity lexicons. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3272–3280, Marseille, France, May 2020. European Language Resources Association.

[59] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[60] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents.

In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org, 2014.

[61] Seung-Wook Lee, Jung-Tae Lee, Young-In Song, and Hae-Chang Rim. High precision opinion retrieval using sentiment-relevance flows. pages 817–818, 01 2010.

[62] Tao Li and Mitsunori Ogihara. Music genre classification with taxonomy. volume 5, pages v/197 – v/200 Vol. 5, 04 2005.

[63] Qilian Liang, Jiasong Mu, Wei Wang, and Baoju Zhang. *Communications, Signal Processing, and Systems: Proceedings of the 2016 International Conference on Communications, Signal Processing, and Systems*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[64] Zhu Liu, Yao Wang, and Tsuhan Chen. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI Signal Processing*, 20, 04 1998.

[65] Lie Lu and Hao Jiang. Content analysis for audio classification and segmentation. *Speech and Audio Processing, IEEE Transactions on*, 10:504 – 516, 11 2002.

[66] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[67] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[68] Suraj Maharjan, Sudipta Kar, Manuel Montes, Fabio A. González, and Thamar Solorio. Letting emotions flow: Success prediction by modeling the flow of emotions in books. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume*

2 (Short Papers), pages 259–265, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[69] Yi Mao and Guy Lebanon. Isotonic conditional random fields and local sentiment flow. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 961–968. MIT Press, 2007.

[70] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, jun 1947.

[71] Rada Mihalcea, Carmen Banea, and Janyce Wiebe. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[72] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[73] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky. Spectral entropy based feature for robust asr. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–193, 2004.

[74] D. Mitrovic, M. Zeppelzauer, and C. Breiteneder. Discrimination and retrieval of animal sounds. In *2006 12th International Multi-Media Modelling Conference*, pages 5 pp.–, 2006.

[75] Dalibor Mitrović, Matthias Zeppelzauer, and Christian Breiteneder. Chapter 3 - features for content-based audio retrieval. In *Advances in Computers: Improving the Web*, volume 78 of *Advances in Computers*, pages 71–150. Elsevier, 2010.

[76] Dalibor Mitrović, Matthias Zeppelzauer, and Christian Breiteneder. Chapter 3 - features for content-based audio retrieval. In *Advances in Computers: Improving the Web*, volume 78 of *Advances in Computers*, pages 71–150. Elsevier, 2010.

[77] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[78] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217 – 250, 2012.

[79] Jasmina Dj. Novaković, Alempije Veljović, Siniša S. Ilić, Željko Papić, and Tomović Milica. Evaluation of classification models in machine learning. *Theory and Applications of Mathematics amp; Computer Science*, 7(1):Pages: 39 –, Apr. 2017.

[80] Shubham Pateria. Aspect based sentiment analysis using sentiment flow with local and non-local neighbor information. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2635–2646, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[81] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. 01 2004.

[82] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[83] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[84] Antreas Pogiatzis and Georgios Samakovitis. Using bilstm networks for context-aware deep sensitivity labelling on conversational data. *Applied Sciences*, 10(24), 2020.

[85] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

[86] Lawrence Rabiner and Ronald Schafer. *Theory and Applications of Digital Speech Processing*. Prentice Hall Press, USA, 1st edition, 2010.

[87] Lawrence R. Rabiner and Ronald W. Schafer. Introduction to digital speech processing. *Found. Trends Signal Process.*, 1(1):1–194, jan 2007.

[88] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[89] Arunan Ramalingam and Sridhar Sri Krishnan. Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting. *IEEE Transactions on Information Forensics and Security*, 1:457–463, 2006.

[90] Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. Enhancing unsupervised sentence similarity methods with deep contextualised word representations. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 994–1003, Varna, Bulgaria, September 2019. INCOMA Ltd.

[91] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[92] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Trans. Inf. Syst.*, 12(3):296–333, jul 1994.

[93] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. In *PloS one*, 2015.

[94] Allan Sales, Leandro Balby, and Adriano Veloso. Media bias characterization in brazilian presidential elections. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, HT ’19, page 231–240, New York, NY, USA, 2019. Association for Computing Machinery.

[95] Iqbal H. Sarker, A. S. M. Kayes, and Paul A. Watters. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data*, 6:1–28, 2019.

[96] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[97] Jangwon Seo and Jiwoon Jeon. High precision retrieval using relevance-flow graph. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 694–695, New York, NY, USA, 2009. ACM.

[98] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, jan 2001.

[99] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

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

[101] C Silverman. Hyperpartisan facebook pages are publishing false and misleading information at an alarming rate. buzzfeed, nov. 16, 2016.

[102] Roberta Sinoara, José Camacho-Collados, Rafael Rossi, Roberto Navigli, and Solange Rezende. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 10 2018.

[103] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[104] Diego Tumitan and Karin Becker. Tracking sentiment evolution on user-generated content: A case study on the brazilian political scene. In *SBBD*, 2013.

[105] Larissa Vasconcelos, Claudio Campelo, and Caio Jeronimo. Aspect flow representation and audio inspired analysis for texts. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1469–1477, Marseille, France, May 2020. European Language Resources Association.

[106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. pages 5998–6008, 2017.

[107] Henning Wachsmuth and Benno Stein. A universal model for discourse-level argumentation analysis. *ACM Trans. Internet Technol.*, 17(3):28:1–28:24, June 2017.

[108] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Comput. Linguist.*, 30(3):277–308, September 2004.

[109] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, page 347–354, USA, 2005. Association for Computational Linguistics.

[110] Chuhan Wu, Fangzhao Wu, Junxin Liu, Yongfeng Huang, and Xing Xie. Sentiment lexicon enhanced neural sentiment classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1091–1100, New York, NY, USA, 2019. Association for Computing Machinery.

[111] X Yang, X Yang, H Zhang, Y Ma, and Y Wu. Measurement of semantic textual similarity in clinical texts: Comparison of transformer-based models. *JMIR Med Informatics*, 8(11), nov 2020.

[112] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Signals and Communication Technology. Springer, London, 2015.

[113] T. Zhang and C.-C.Jay Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4):441–457, 2001.

Appendix A

Experiments Results - All Metrics

Chapter 5 only presents the most representative metrics collected on the experimentation: PR-AUC on the tasks involving news and ROC-AUC on the tasks involving reviews.

This appendix presents all the metrics collected: Precision, Recall, and F1 Score. Additionally, accuracy is presented on the balanced scenarios.

A.0.1 Fake News Detection in English

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
Precision	0.85	0.85	0.88	0.70	0.69	0.80	0.78	**0.81**
Recall	0.22	0.11	0.22	0.70	0.69	0.80	0.78	**0.81**
F1 Score	0.35	0.19	0.36	0.60	0.61	0.72	0.66	**0.74**
PR-AUC	0.62	0.49	0.65	0.78	0.73	0.81	0.83	**0.84**

Table A.1: Fake News Detection in English results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

	Best Model - Deep Learning trained with BERT+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
Precision	**0.81**	0.74	0.74	0.79	0.79
Recall	**0.81**	0.74	0.74	0.79	0.79
F1 Score	**0.74**	0.62	0.64	0.70	0.70
PR-AUC	**0.84**	0.80	0.81	0.82	0.80

Table A.2: Fake News Detection in English results. This table presents the results of the best model (DL-BERT+ALF) trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

A.0.2 Fake News Detection in Portuguese

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
Precision	0.96	0.65	0.96	**0.97**	0.82	0.80	0.97	0.97
Recall	**0.97**	0.18	0.97	0.97	0.82	0.80	0.97	0.97
F1 Score	**0.96**	0.28	0.96	0.95	0.61	0.67	0.95	0.95
PR-AUC	**0.98**	0.46	0.98	0.96	0.90	0.88	0.96	0.96

Table A.3: Fake News Detection in Portuguese results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

	Best Model - Machine Learning trained with ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
Precision	**0.96**	0.96	0.95	0.95	0.95
Recall	**0.97**	0.97	0.96	0.96	0.97
F1 Score	**0.96**	0.96	0.96	0.96	0.96
PR-AUC	**0.98**	0.98	0.98	0.98	0.94

Table A.4: Fake News Detection in Portuguese results. This table presents the results of the best model (SL - ALF) trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

A.0.3 Newspaper Columns versus News Classification

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
Precision	**0.94**	0.84	0.93	0.88	0.90	0.86	0.92	0.88
Recall	0.73	0.62	0.80	0.88	0.90	0.86	**0.92**	0.88
F1 Score	0.82	0.72	0.86	0.79	0.82	0.75	**0.87**	0.77
PR-AUC	0.94	0.85	0.95	0.92	0.94	0.93	**0.97**	0.94

Table A.5: Newspaper Columns versus News results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

	Best Model - Deep Learning trained with D2V+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
Precision	0.92	0.92	0.93	0.93	**0.94**
Recall	0.92	0.92	0.93	0.93	**0.94**
F1 Score	0.87	0.86	0.88	0.88	**0.90**
PR-AUC	0.97	0.96	0.96	0.96	**0.98**

Table A.6: Newspaper Columns versus News results. This table presents the results of the best model trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).

A.0.4 Movie Reviews Sentiment Polarity Classification in English

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
Accuracy	0.74	0.79	0.80	0.51	0.78	**0.85**	0.75	0.79
Precision	0.75	0.79	0.80	0.51	0.78	**0.85**	0.75	0.79
Recall	0.71	0.78	0.79	0.51	0.78	**0.85**	0.75	0.79
F1 Score	0.73	0.79	0.80	0.51	0.78	**0.85**	0.75	0.79
ROC-AUC	0.74	0.79	0.80	0.52	0.85	**0.92**	0.85	0.88

Table A.7: Movie Reviews Sentiment Polarity Classification in English results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

	Best Model - Deep Learning trained with BERT+ALF					
Features	Bert feats	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
Accuracy	**0.85**	0.79	0.74	0.75	0.75	0.73
Precision	**0.85**	0.79	0.74	0.75	0.75	0.73
Recall	**0.85**	0.79	0.74	0.75	0.75	0.73
F1 Score	**0.85**	0.79	0.73	0.75	0.75	0.72
ROC-AUC	**0.92**	0.88	0.83	0.84	0.84	0.79

Table A.8: Movie Reviews Sentiment Polarity Classification in English results. This table presents the results of the best model trained with BERT feats, BERT+all ALFs (All Feats), with the BERT+ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only BERT+ALFs calculated over the entire flow (Flow feats).

A.0.5 Book Reviews Sentiment Polarity Classification in Portuguese

	SL Models			DL Models				
Features	ALF	D2V	D2V+ALF	ALF	D2V	BERT	D2V+ALF	BERT+ALF
Accuracy	0.57	0.63	0.69	0.55	0.64	0.58	**0.72**	0.56
Precision	0.60	0.66	0.70	0.55	0.64	0.58	**0.72**	0.56
Recall	0.46	0.52	0.64	0.55	0.64	0.58	**0.72**	0.56
F1 Score	0.52	0.58	0.67	0.53	0.63	0.58	**0.72**	0.54
ROC-AUC	0.57	0.63	0.69	0.56	0.71	0.61	**0.79**	0.58

Table A.9: Book Reviews Sentiment Polarity Classification in Portuguese results. This table presents the results of the best SL and DL models trained with the ALFs, D2V, BERT, D2V+ALFs and BERT+ALFs features.

	Best Model - Deep Learning trained with D2V+ALF				
Features	ALL feats	Frame 0 feats	Frame 1 feats	Frame 2 feats	Flow feats
Accuracy	**0.72**	0.70	0.71	0.70	0.69
Precision	**0.72**	0.70	0.71	0.70	0.69
Recall	**0.72**	0.70	0.71	0.70	0.69
F1 Score	**0.72**	0.69	0.71	0.70	0.68
ROC-AUC	**0.79**	0.78	0.79	0.79	0.75

Table A.10: Book Reviews Sentiment Polarity Classification in Portuguese results. This table presents the results of the best model trained with all ALFs (All Feats), with the ALFs of a single frame (Frame 0 feats, Frame 1 feats and Frame 2 feats), and with only the ALFs calculated over the entire flow (Flow feats).