Journal of the Brazilian Computer Society, 2024, 30:1, doi: 10.5753/jbcs.2024.3759

Extracting Features from Text Flows based on Semantic Similarity for Text Classification: an Approach Inspired by Audio Analysis

Larissa Lucena Vasconcelos [**Federal Institute of Paraíba** | ***larissa.vasconcelos@ifpb.edu.br***]
Claudio E. C. Campelo [**Federal University of Campina Grande** | ***campelo@computacao.ufcg.edu.br***]

Federal Institute of Paraiba, PB-264, Monteiro - PB, Brazil

Federal University of Campina Grande, Aprigio Veloso 882, Campina Grande - PB, Brazil

Received:16 October 2023 • **Accepted:** 03 May 2024 • **Published:** 25 September 2024

Abstract Text classification is a mainly investigated challenge in Natural Language Processing (NLP) research. The higher performance of a classification model depends on a representation that can extract valuable information about the texts. Aiming not to lose crucial local text information, a way to represent texts is through flows, sequences of information collected from texts. This paper proposes an approach that combines various techniques to represent texts: the representation by flows, the benefit 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. In order to perform text classification, this 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 extraction of twenty-five features inspired by audio analysis (named Audio-Like Features). 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. We evaluated the method in three NLP classification tasks: Fake News Detection in English, Fake News Detection in Portuguese, and Newspaper Columns versus News Classification. The approach efficacy is compared to baselines that embed semantics in text representation: the Paragraph Vector and the BERT. The objective of the experiments was to investigate if the proposed approach could compete with the baselines methods improve their efficacy when associated with them. The experimental evaluation demonstrates that the association between the proposed and the baseline methods can enhance the baseline classification efficacy in all three scenarios. In the Fake News Detection in Portuguese task, our approach surpassed the baselines and obtained the best effectiveness (PR-AUC = 0.98).

Keywords: NLP, Semantic Similarity, Text Classification, TextFlow, Lexicon-based Representation

1 Introduction

Text classification is one of the most discussed and studied challenges in Natural Language Processing (NLP) research. Fake news detection, spam filtering, scientific articles categorization, idioms identification and sentiment analysis are examples in the vast field of text classification [Feldman and Sanger, 2006; Aggarwal and Zhai, 2012; Aggarwal, 2018; Sebastiani, 2002; Khadhraoui *et al.*, 2022; Briskilal and Subalalitha, 2022]. 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 knowledge about the texts [Aggarwal, 2018; Giannakopoulos *et al.*, 2012; Jin *et al.*, 2016; Li *et al.*, 2022; Gasparetto *et al.*, 2022].

A prevalent text representation model is 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 [Goldberg and Hirst, 2017; Li *et al.*, 2022]. The BoW representation does neither express words order or text syntax or semantics. 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 semantic information is a powerful tool to recognize different contexts even when a similar vocabulary is used [Aggarwal and Zhai, 2012; Gasparetto *et al.*, 2022]. Word embeddings are a widespread representation that embeds semantics on representation, in which vectors of various dimensions can express context information of words by adding a dependency between the words to the representation: how much more approximate in a context, more dependent [Goldberg and Hirst, 2017; Dharma *et al.*, 2022]. Word2Vec [Mikolov *et al.*, 2013], Paragraph Vector [Le and Mikolov, 2014], Glove [Pennington *et al.*, 2014] and FastText [Bojanowski *et al.*, 2016] 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 [Dev *et al*., 2020; Li *et al*., 2022]. As examples of contextualized word representations, it is possible to list: ELMo [Peters *et al*., 2018], BERT [Devlin *et al*., 2019], and GPT [Radford *et al*., 2019].

Another way of embedding semantic information on text representation is recurring to the use of lexicons. In the domain of NLP, lexicons are sets of terms or expressions that reflect a specific semantic context of a language [Mutinda *et al*., 2023; Muñoz and Iglesias, 2022; Bhowmik *et al*., 2022]. 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 [Araque *et al*., 2019].

Recent research has been associating the power of the word embeddings representation with the additional knowledge that lexicons promote to achieve a more accurate text representation model [Araque *et al*., 2019; Wu *et al*., 2019; Bao *et al*., 2019; Jeronimo *et al*., 2020; Fu *et al*., 2018; Muñoz and Iglesias, 2022; Mutinda *et al*., 2023]. The widely used pre-trained word embeddings contain semantic information for a general context. Associating these pre-trained word embeddings to lexicon knowledge 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 [Kusner *et al*., 2015]). 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 [Jeronimo *et al*., 2019, 2020]. As Aker *et al*. [2019] 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 sentences and computing the semantic similarity for each sentence. In this case, the sequence of the text sentence's semantic similarity constitutes the text representation. We name this representation as *Text Flow*, following Mao and Lebanon [2007] definition of flow: a sequence of information collected from the words, sentences, or paragraphs of the text.

As Mao and Lebanon [2007], some other studies [Wachsmuth and Stein, 2017; Ghanem *et al*., 2021], represent texts as flows (not based on semantic similarity) and use the flows to perform classification tasks. As Filatova [2017] and Seo and Jeon [2009], other works obtain good results by extracting relevant features from the flows before performing the classification task.

In a previous work [Vasconcelos *et al*., 2020], we presented a preliminary method that represents texts by flows that incorporate lexicon information and then extracts few features inspired by audio analysis from the flows. Singularly, it combines the text representation by flows, the benefit 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.

This paper proposes an enhanced version of that method, significantly augmenting the number and complexity of the features extracted, bringing a more valuable text representation. The idea of proposing this text representation method comes from the insight that ensembling valuable techniques (commonly used alone) could enhance the quality of text representation. Therefore, we use the flow representation (which avoids the loss of local information in the text) associated with semantic information (brought by the combination of word embeddings and lexicons).

To guide the experiments, we defined the following Research Questions (RQ):

- RQ1: Is a classifier model using the proposed text representation approach competitive to robust baseline methods in terms of efficacy[1]?
- RQ2: Can the association of the proposed method with robust baseline methods improve the baselines' efficacy on text classification tasks?
- RQ3: Is there a subset of features (from the proposed set) that could perform equally or even better than the entire set of features in all considered classification tasks?
- RQ4: Does the method achieve better efficacy when used to feed shallow or deep learning classification algorithms?

We evaluated the new method version in three NLP classification tasks: Fake News Detection in English, Fake News Detection in Portuguese, and Newspaper Columns versus News Classification. These experiments are more robust than those performed on the preliminary work since they comprised stronger baselines (Paragraph Vector (Doc2Vec) and BERT). The classification models created from the method set of 25 features surpass the effectiveness of the baselines features models in some scenarios. We also performed the classification by creating models by combining our method features and the features of each baseline. In all scenarios, these classification models improve the classification tasks' efficacy.

The rest of the paper is structured as follows. Section 2 summarizes previous works regarding enhancing text representation through flows and lexicons. Section 3 presents the enhanced version of the extraction method comprising the

[1] In this article, we use the term efficacy to refer to the accuracy of text classification models, being represented on the experiments results by the PR-AUC metric.

new features. Section 4 thoroughly explores the executed experiments, including used datasets, lexicons, and experimental setup, as well as the experimental results and discussion. Finally, the paper concludes with Section 5, which depicts the conclusions drawn from the evaluation and outlines the possible future lines of work.

2 Related Work

As previously discussed, an approach to avoid the loss of relevant information is representing a text as a flow. Mao and Lebanon [2007] propose a variant of conditional random fields [Lafferty *et al.*, 2001] to proceed with local sentiment prediction in reviews. Their solution 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. Then, they use the predicted local sentiment flows to predict the global review's sentiment.

In their study, Wachsmuth and Stein [2017] represent the text's discourse-level structure as a flow of rhetorical moves. They propose a clusterisation in training flows and compare test flows to training cluster's centroids to perform global reviews sentiment classification. They obtain the rhetorical moves from state-of-art methods. Similar to Mao and Lebanon [2007] work , Wachsmuth and Stein Wachsmuth and Stein [2017] use the entire flow to perform classification but apply different information (rhetorical moves) and way to generate the flows (clusterization and comparison to cluster centroid). In this study, we ground our text representation on flows as Mao and Lebanon [2007] and Wachsmuth and Stein [2017]. However, we generate the flows by semantic similarity distances (WMD) from lexicons and extract features to perform classification.

Alike our work, Filatova [2017] and Lee *et al.* [2010] extract features from flows to execute classification. Nevertheless, the manner they generate flows and extract features are substantially different from ours. Filatova [2017] models product reviews as sentiment flows, assigning sentiment labels to each sentence using the Stanford Sentiment Analysis tool [Socher *et al.*, 2013]. The researcher utilizes sentiment switchings between sentences as features for sarcasm detection. In their paper, Lee *et al.* [2010] propose to represent texts as a combination of a sentiment flow and a relevance flow (defined in Seo and Jeon [2009] study) to proceed with opinion retrieval. A score that reflects its 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 sentence scores, the fraction of peaks, and the first peak position.

Another way to model flows for representing texts is using neural networks. Maharjan *et al.* [2018] propose to model the flow of various emotions over a book aiming to capture patterns that should represent the emotional arcs of the story. They extract emotion vectors from different book chunks and feed them into a recurrent neural network to create the emotional flow. Then they proceed with a prediction of success in books. In their research, Ghanem *et al.* [2021] obtained promising results on the fake news detection task by modeling the flow of affective information in fake news articles using a neural architecture.

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 knowledge if the exact term appears on text. In our method, 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.

In addition to the studies of Lee *et al.* [2010], Maharjan *et al.* [2018], and Ghanem *et al.* [2021], several other works rely on lexicons to improve their text representation with relevant knowledge, but still not use semantic similarity as our work. Tumitan and Becker [2013] and Avanço and Nunes [2014] works calculate sentences polarity by summarizing the polarity of lexicon terms identified on text (add positive terms, and subtract negative terms). Tumitan and Becker [2013] use a sentiment lexicon to calculate a sentence-polarity score aiming to conduct a study to rate comments on Brazilian political news. Proposing to perform sentiment classification for Brazilian Portuguese technology product reviews, Avanço and Nunes [2014] calculated the sentences and text's polarity, combined with a linguistic knowledge about contextual valence shifting. Muñoz and Iglesias [2022] present a method to detect stress in textual data, combining a lexicon-based feature framework with distributional representations to enhance classification performance. The framework based on lexicons exploits affective, syntactic, social, and topic-related features.

In order to obtain more sophisticated and valuable knowledge from texts taking advantage of lexicons, some research calculates semantic similarity metrics between texts and lexicons, thus becoming independent of the presence of the exact terms exclusively. Pawar and Mago [2019] proposed to compute the semantic similarity between words and sentences by finding the Shortest Path Distance over the WordNet hierarchy. On the other hand, our proposed method, Jeronimo *et al.* [2020], and Araque *et al.* [2019] studies extract semantic similarity distances between texts and lexicons over a vectorial embedding space. Similar to our work, Jeronimo *et al.* [2020] uses WMD as semantic similarity metric, and Araque *et al.* [2019] uses the Cosine Distance.

Jeronimo *et al.* [2020] proposed a new set of Brazilian Portuguese subjectivity lexicons. They use the average of the WMD calculated for each sentence of the text to the lexicons as features to perform three tasks: automated essay scoring, identifying subjectivity bias in Brazilian presidential elections, and fake news detection. Araque *et al.* [2019] 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 [Mikolov *et al.*, 2013; Le and Mikolov, 2014]; and the second, a vector of the Cosine Distances between words in the text and selected words from a sentiment lexicon. These two studies differ from our method mainly by the method of representing texts and extracting features, as we use flows and extract features from them.

Likewise our and other works presented here, another possible effort to utilise the lexicon's power is to aggregate this knowledge to neural networks. A tendency is to include the lexicon information in the word embeddings that will represent the texts. Wu *et al*. [2019] 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. Fu *et al*. [2018] create an attention mechanism based on the correlation of each sentence word embeddings and the lexicons (positive and negative) words embeddings. Mutinda *et al*. [2023] propose a sentiment classification model combining sentiment lexicon, N-grams and BERT to vectorize words from the input text, and then, use CNN to perform sentiment classification.

To the best of our knowledge, our method is the first one to represent texts as flows through calculating semantic similarity metric between each sentence and a lexicon over a vectorial space. Moreover, we believe this is the first attempt reported in the literature to extract features based on audio analysis techniques to perform text classification tasks. In this way, this approach aims to combine the benefits of each technique presented to improve text representation, resulting in more accurate text classification models.

3 Text Flows Representation and Audio-Like Features Extraction

This section presents our proposed approach for representing texts by flows, incorporating lexicon information via semantic similarity distance on an embedding space[2]. Additionally, we introduce all the proposed features based on the ones used in audio analysis.

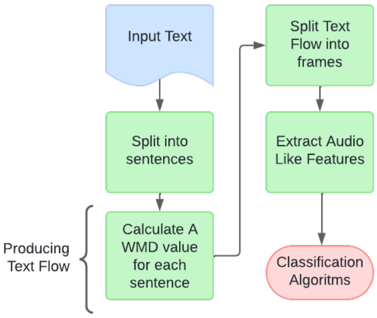
Input Text
Split into sentences
Calculate A WMD value for each sentence
Producing Text Flow
Split Text Flow into frames
Extract Audio Like Features
Classification Algoritms

Figure 1. Proposed approach: Text Flow representation and Audio-Like Features extraction.

Figure 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.

3.1 Text Flows Representation

Figure 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.

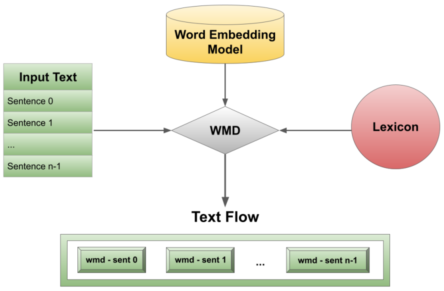

Figure 2. Text Flow Creation.

Presented by Kusner et al., the WMD is a distance function that measures the similarity of two text documents. Kusner et al. claim that distances between vectors of word embeddings are semantically significant. WMD is defined as the shortest distance that the word embeddings of a document need to "travel" over the embedding space until reaching the word embeddings of another document. In other words, calculating the WMD involves:

- representing text documents as a weighted point cloud (as we can see in Figure 3) of embedded words using word embedding techniques (e.g., W2V);
- comparing the vector representations of words (points on cloud) between two documents over the embedding space;
- finding the shortest possible distance, considering the cost of transporting words from one document to another.

The final distance between two documents is the minimum cumulative distance that words in one document must traverse to meet the other text's set of points on word embedding space.

[2] Code available on https://github.com/larissalucena/AudioLikeFeatures

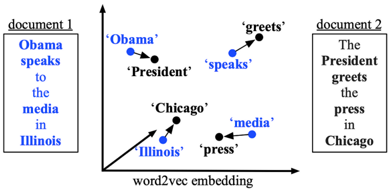

Figure 3. An illustration of the word mover's distance. [Kusner *et al.*, 2015]

Figure 3 presents a simple example of how WMD is calculated. A weighted point represents each word of both documents, and the minimum cumulative distance is calculated. For example, the word "Obama" is closer to "President" than to "greets" or any other word in Document 2. Then, this distance will be part of the final calculation. This calculation is repeated for every word in the text. Word Mover's Distance (WMD) is inspired by the Earth Mover's Distance (EMD) problem, a well-known problem in transportation theory and linear optimization. EMD calculates the distance between two mass distributions, finding the minimum work required to transform one distribution into another. WMD adapts this concept to measure the semantic distance between documents, treating words as "mass" that needs to be transported from one document to another, minimizing the cost of this transportation.

In face of that, besides incorporating the specific context information from the lexicon into the representation, our approach avoids depending only on finding (and counting) known vocabulary terms.

3.2 Analysing Text Flows Inspired by Audio Analysis

The text flow can be approximated to a digital signal [Liang *et al.*, 2017] if we consider each sentence of a document as a point in "time" and each sentence WMD value as the "signal" amplitude (flow amplitude). Figure 4 shows the flow of a document containing sixty sentences. The y-axis presents the WMD value calculated between each sentence and the lexicon. We can perceive that the first sentence is semantically dissimilar to the lexicon, and the sixteenth is the most similar.

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 [Yu and Deng, 2015]. Speech is a digital signal, specifically an audio signal, and, therefore, can be investigated by audio analysis [Rabiner and Schafer, 2007, 2010]. 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.

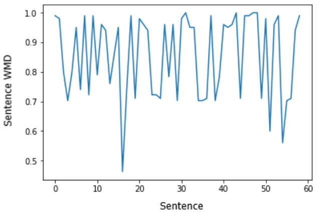

Figure 4. Text Flow graph representation

3.3 Audio-Like Features Extraction and Text Flow Frame Fragmentation

After the Text Flow creation, the method proceeds with the so-called Audio-Like Features (ALFs) extraction (Fig. 1). Given the text flow's approximation to a digital signal and the relationship between text, speech, and audio, 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. 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, comparable to another text's first part. 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 reviews of products or services. The definition of the number of frames adequate for each dataset can be obtained empirically.

3.4 Audio-Like Features

This section presents the procedure taken in this research to choose what well-established audio features should (or could) be 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 Mitrović *et al.* [2010a] and extended by the study of Alías *et al.* [2016]. 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 selected features from the time and frequency domains among all existing domains due to their low complexity and adequacy to

Table 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

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 Mitrović *et al.* [2010a] 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; 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 the facility to adapt them. Thus, this choice would be a safer way to validate the implementation of this method. 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 1 presents the names of the features, their domain, and temporal scale.

It is worth highlighting that, in the previous article, we introduced only three time-domain frame-level ALFs: *Energy*, *Median-Crossing Rate*, and *Energy Entropy*. The newly implemented features enhanced the efficiency of the method in the more robust performed experiments (presented in the following section).

Following, we will describe all these features.

3.4.1 Time-Domain Frame-Level Features

We now present the time-domain frame-level features extracted by this proposed method.

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 sentences of the i-th frame, where F_L is the length of the frame. The implementation of Energy is defined as in Equation 1:

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

Here we normalised the Energy by dividing it by F_L to remove the dependency on the frame length. 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 [Zhang and Kuo, 2001; Rabiner and Schafer, 2007]. 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 (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 2:

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

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 3:

$$msgn[x_i(n)] = \begin{cases} 1, & \text{if } x_i(n) > median. \\ -1, & \text{if } x_i(n) < median. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

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) [Zhang and Kuo, 2001].

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.

Shannon entropy plays a central role in information theory as a measure of information, choice, and uncertainty [Shannon, 2001]. In audio analysis, the same 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 [Giannakopoulos and Pikrakis, 2014; Rabiner and Schafer, 1978].

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 (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):

$$e_j = \frac{Esubframe_j}{Eframe_i} \tag{4}$$

where

$$Eframe_i = \sum_{k=1}^{K} Esubframe_k \tag{5}$$

At a final step, the entropy, $Ent(i)$ of the sequence e_j is computed according to Equation 6:

$$Ent(i) = -\sum_{j=1}^{K} e_j * log_2(e_j) \tag{6}$$

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

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 [El-Maleh *et al.*, 2000; Rabiner and Schafer, 2010]. 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 7.

$$LP - MCR = \frac{MCRflow}{MCRlp} \tag{7}$$

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 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.

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 [Mitrović *et al.*, 2010a; Alías *et al.*, 2016]. 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 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 [Liu *et al.*, 1998].

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.

3.4.2 Time-Domain Flow-Level Features

In this subsection, we describe the time-domain features extracted from the entire flow.

Mitrovic *et al.* [2006] 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 article, the AD is a set of seven individual features that characterize the flow in terms of "near" and "far" segments to the lexicon [Mitrovic *et al.*, 2006]. 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 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 [ISO/IEC, 2002]. 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.

3.4.3 Frequency-Domain Frame-Level Features

The frequency-domain audio features constitute the most extensive group of audio features reported in the literature [Mitrović *et al.*, 2010a]. Hence, it was possible to adapt some of these features to represent texts. Aiming to compute the frequency (or spectral) features, it is worth transforming the representation (flow, in this case) on the time domain to the frequency domain. This transformation is usually fulfilled from the Short-Time Fourier Transform (STFT) or derived from an autoregression analysis [Alías *et al.*, 2016]. In this study, we used the Scipy API v1.3.2 signal.stft function, using default parameters[3] and passing the Text Flow (document sentences WMD values array) as the time series of measurement values. Additional information can be found in the Scipy API[4].

Among the spectral features presented in this subsection, seven are computed after generating the STFT output. Only the Pitch feature is extracted from flow resulting from an autocorrelation function.

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 [Mitrović *et al.*, 2010a; Alías *et al.*, 2016].

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 8:

[3] The main parameters values used by this Scipy version are: sampling frequency = 1.0; Window type = Hann; segment length = 256; overlapping = 128

[4] https://docs.scipy.org/doc//scipy-1.3.2/reference/generated/scipy.signal.stft.html

$$SFlux_{(i,i-1)} = \sum_{k=1}^{Wf_L} \left(EN_i(k) - EN_{i-1}(k)\right)^2 \tag{8}$$

where $EN_i(k)$ is the k-th STFT coefficient at the i-th frame, calculated as in Equation 9.

$$EN_i(k) = \frac{X_i(k)}{\sum_{l=1}^{Wf_L} (X_i(l))} \tag{9}$$

The Spectral Entropy is computed similarly to the Energy Entropy, but now, the computation occurs in the spectral domain, which means using the STFT output. 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 is a feature adapted from the Subband Energy Ratio [Mitrović *et al.*, 2010a; Alías *et al.*, 2016]. 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 [Mitrović *et al.*, 2010a; Alías *et al.*, 2016; Mitrović *et al.*, 2010b].

The Spectral Energy Ratio is implemented as the ratio between the frame Spectral Energy and the entire flow Spectral Energy (take the Equation 1 as reference). In a broad sense, it also roughly describes the flow energy distribution of the spectrum.

Another proposed feature is Spectral Flatness. 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 [Alías *et al.*, 2016; Mitrović *et al.*, 2010a]. Our 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 [Ramalingam and Krishnan, 2006]. It estimates to which degree the magnitudes in a spectrum are uniformly distributed [Alías *et al.*, 2016].

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 [Mitrović *et al.*, 2010a]. This proposed method adapted the Spectral Crest Factor as the ratio of the maximum power spectrum and the mean power spectrum of a frame [Li and Ogihara, 2005][5].

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 are two other introduced features. 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 [Peeters, 2004]. At the same time, the Spectral Kurtosis describes the flatness of the spectral distribution around its mean [Peeters, 2004]. A Kurtosis value lower than three describes a flatter spectral distribution, while a value bigger than three describes a peaker distribution [Peeters, 2004]. These are two other features that can capture the relationship between text and lexicon, distinguishing texts with a different connection to the lexicon.

The last feature in this subsection is Pitch. Also known as Fundamental Frequency, the Pitch feature is defined as the first peak of the local normalized spectro-temporal autocorrelation function [Cho *et al.*, 1998]. 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.

The present method identifies Pitch as the first peak of the frame's spectral autocorrelation function output[6]. Texts flow presenting different Pitch values demonstrate distinct maximum autocorrelation values considering a particular lexicon.

3.4.4 Frequency-Domain Flow-Level Feature

The unique feature to be presented in this subsection is Jitter. In audio analysis, jitter is the cycle-to-cycle Pitch variations or the absolute mean difference between consecutive periods of an audio signal [Farrús *et al.*, 2007]. In this study, Jitter is adapted to a frame-by-frame pitch variation in the domain. It is calculated by the mean of the absolute difference between the pitches of two consecutive frames [Farrús *et al.*, 2007]. The Jitter feature is a form of analysing Pitch variation throughout the flows.

4 Experimental Evaluation and Discussion

This section reports our experimental conduction, presenting the detailed evaluation method, the results of the evaluation tasks, and discussing the obtained results. The three classification tasks are: Fake News Detection in English, Fake News Detection in Portuguese, and Newspaper Columns versus News Classification. Unlike our preliminary work, the experiments reported in this paper include the use of deep learning classification models and a comparison between our method and robust baselines. Therefore, the new experiments are more robust than those presented in our previous work.

We evaluate our approach's competitiveness by comparing its classification results to two baselines that embed semantic information in text representation: Paragraph Vector (D2V) [Le and Mikolov, 2014] - a static word embedding - and BERT [Devlin *et al.*, 2019] - a contextualized word representation. Besides the classification employing models created from the ALFs, D2V, and BERT features separately,

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

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

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.

4.1 Datasets

For the Fake News Detection tasks, we use the datasets presented by Jeronimo *et al*. [2020]. The English dataset has 5,994 legitimate news collected from the *All The News Dataset* available at Kaggle[7], with 2,598 coming from CNN[8], 1,798 from The Guardian[9] and 1,598 from The New York Times[10], published between 2016 and 2017.

Fake news, in turn, were compiled by Torabi ASR and Asr and Taboada [2019], with 103 political news coming from Snopes[11], 75 political news coming from Horne and Adali [2017] work and 40 stories from Buzzfeed's top-ranked fake news[12]. All fake news on the dataset were fact-checked. Despite the existence of more extensive fake news datasets, we preferred using smaller datasets that we could confirm were fact-checked. We performed some double-fact-checking on larger datasets but found repeated news, non-fact-checked news, and even documents containing only one sentence being used as news. So, we chose to use smaller but fact-check-guaranteed datasets.

The Portuguese news dataset has 207,914 legitimate Brazilian news and 121 fact-checked fake news disseminated in Brazil made available by Jeronimo *et al*. [2020]. The dataset of legitimate news was collected from two of the biggest news sites in Brazil: Estadao[13] and Folha de Sao Paulo[14]. 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[15] and Boatos[16].

On the Newspaper Columns versus News Classification task, we employed the Jeronimo *et al*. [2020] legitimate news to represent the objective news. The newspaper columns dataset was firstly presented in our previous work [Vasconcelos *et al*., 2020], being formed by 7,062 newspaper columns articles collected by automated mining.

As all datasets are highly imbalanced, to avoid oversampling techniques that would not reflect a realistic scenario, we decided to follow the four-to-one proportion earlier adopted by us [Vasconcelos *et al*., 2020] and by Jeronimo *et al*. [2020] to execute all the experiments. In other words, we randomly chose 872, 484, and 28,248 legitimate news for the Fake News Detection in English, in Portuguese and Newspaper Columns versus News Classification tasks, respectively.

4.2 Lexicons

To generate the flows in English, we used a combination of three sets of lexicons: 1) the six subjectivity lexicons compiled by . Recasens *et al*. [2013]; 2) the subjectivity positive and negative polarities sentiment lexicons presented by Wilson *et al*. [2005]; 3) the positive and negative polarities sentiment lexicons proposed by Choi and Wiebe [2014].

To the Fake News Detection in Portuguese task, we applied the Reli-Lex, a set of eight sentiment polarity lexicons [Freitas, 2013]. Besides these sentiment polarity lexicons, we employed the five subjectivity lexicons proposed by Amorim *et al*. [2018] to the Newspaper Columns versus News task.

4.3 Baselines

We compare our method classification results to two robust baselines that embed semantic information in text representation: Paragraph Vector (D2V) [Le and Mikolov, 2014] and BERT [Devlin *et al*., 2019]. Concerning the D2V, we trained a model using the remaining legitimate news from each dataset that would not be used in the experiments. Then, we created a 100-dimension D2V representation for each text (like Araque *et al*. [2019]). The BERT models used were the English BERT-base for the English news and the multilingual BERT-base for the Portuguese News. Each text's representation was generated considering the 512 first words and comprised 768 dimensions.

4.4 Experimental Setup

We used Jeronimo *et al*. [2020] word embedding word2vec models to calculate the WMD in tasks involving the Portuguese language, in order to create the text flows. To the English language, we used the Word2Vec article [Mikolov *et al*., 2013] pre-trained word embeddings[17].

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

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

[8] www.cnn .com

[9] www.theguardian.com

[10] www.nytimes.com

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

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

[13] https://www.estadao.com.br/

[14] https://www.folha.uol.com.br/

[15] http://www.e-farsas.com/

[16] http://www.boatos.org/

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

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 LSFP consists of repeating the WMD value of the last sentence of each frame until it reaches the size of four sentences (in this evaluation setup case). Fig. 5 illustrates the application of the LFSP to a seven sentences flow. Fig. 5 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. 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).

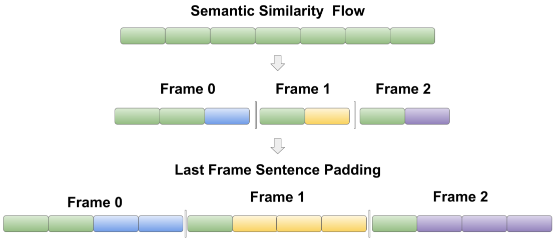

Figure 5. Last Frame Sentence Padding applied to a seven-sentences flow.

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.

4.5 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 XG-Boost 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 [Jang *et al.*, 2020; Muñoz and Iglesias, 2022]. All models were trained using the train and validation splits, considering several learning rates, numbers of neurons, and epochs.

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

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 our method could improve the D2V and BERT methods efficacy.

We evaluated the classification efficacy in terms of the Area Under the Precision-Recall curve (PR-AUC), a metric that satisfies our class imbalance scenario [Saito and Rehmsmeier, 2015; Davis and Goadrich, 2006].

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 [Kuhn and Johnson, 2013].

This research used the SHapley Additive exPlanations (SHAP) values [Lundberg and Lee, 2017] 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.

4.6 Results and Discussion

4.6.1 Fake News Detection in English

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.

Table 2 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 with the combination of D2V and ALFs (D2V+ALF) and BERT and ALFs

	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 2. 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.

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

Aiming to discover if the difference between the results presented by two models is statistically significant, we proceeded with the McNemar statistical hypothesis test [McNemar, 1947; Dietterich, 1998] 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 article 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 [Lundberg and Lee, 2017] to verify what features most impact the classification tasks.

Fig. 6 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.

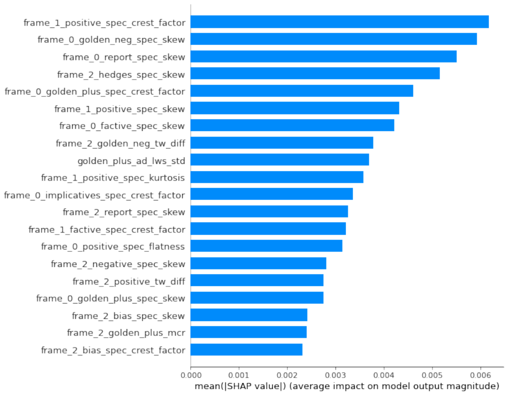

Figure 6. Feature Importance Bar Plot - Fake News Detection in English Classification.

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.

4.6.2 Fake News Detection in Portuguese

The best SL and DL models, respectively, were the RF with 100 trees in the forest, and the tree's maximum depth equals to 18, and the Bi-LSTM model set with 128 neurons, a learning rate of 5e-5, and 100 epochs. Table 3 presents all PR-AUC achieved in this scenario.

The ALFs alone on SL obtained the best efficacy in this scenario - PR-AUC = 0.98, 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

Features	SL Models			DL Models				
	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 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.

Features	SL Models			DL Models				
	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 4. 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.

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).

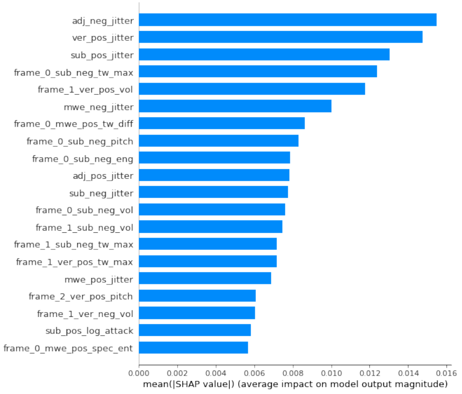

Figure 7. Feature Importance Bar Plot - Fake News Detection in Portuguese Classification.

Figure 7 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 7 shows ten time-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. Leading to a negative answer to RQ3.

4.6.3 Newspaper Columns versus News Classification

In this scenario, 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.

Table 4 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.

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.

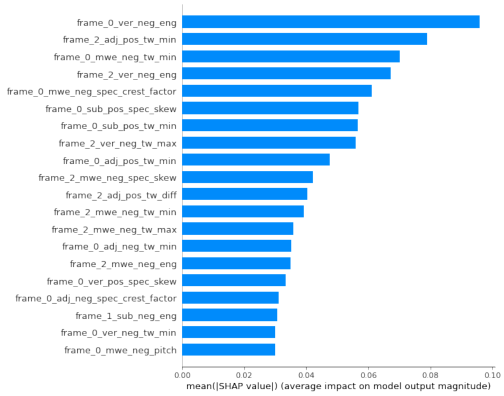

Figure 8. Feature Importance Bar Plot - Newspaper Columns versus News Classification.

Responding to RQ3, we can observe the most impacting features on Newspaper Columns versus News task model presented by Figure 8. 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 TW_Min and TW_Max, presenting that the lexicon dimension strength and the closest and the furthest points to the lexicon dimension are decisive on the classification task result. No flow-level features permeate the features shown in Figure8, which suggests that the analysis is more effective when breaking the texts into parts in this case. Despite the subjectivity lexicon dimensions are not present on the most important features, the experiment using just the Reli-Lex achieves worse results.

4.6.4 Overall Discussion

Analysing all the presented experiments results, the ALFs present the best results alone or combined to the baselines. 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. Furthermore, our method achieved the best results in Fake News Detection in Portuguese. In this, scenario, 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.

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

Still concerning the scenarios where the association of ALF to the baselines achieves better results than the baselines alone, we can interpret that, besides the robustness of the baselines models, some characteristics that can differentiate the types of text could be missed, and a method that can identify that information and add them to the classification model, like ours, is relevant. It is worth remembering that the proposed method uses an elaborate way to consider the lexicon information, representing the texts and lexicons on an embedding space and calculating WMD values all over the text. In that way, the text representation can retain more precise semantic information.

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 scenario 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 Baert *et al.* [2020], 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.* [2019], Ghanem *et al.* [2021], and Maharjan *et al.* [2018] papers, using the entire text is crucial to better classifying it.

Concerning threats of validity, in this article, we depicted the design of the experiments and chose adequate metric, statistical test, and even feature importance method, all consolidated on the literature. Also, the datasets were carefully selected. In some cases, after passing by a double fact-checked on fact news. We performed a train-validation-test split on all tasks. Also, we cared about reflecting reality, using the proportion between the classes of news on the experiments.

5 Conclusions and Future Work

In this paper, we have proposed an enhanced version of our previous method to represent texts. We model each text as text flows. These flows are obtained by calculating WMD from each text's sentence to a lexicon considering a word embedding space. Then, we fragment the flows into smaller parts, called frames. Latter, we calculate a bigger set of features adapted from the audio analysis, called Audio-Like Features. For evaluating the method's effectiveness, we performed three NLP classification tasks. For comparison purposes, we also performed the classification tasks using D2V and BERT baselines and combined each baseline with ALFs, improving the quality of the experiments in this paper.

Our method achieved the best results in Fake News Detection in Portuguese, affirmatively answering RQ1 in this task. It approximated the baselines' results in the remaining tasks, even surpassing D2V in the Fake News Detection in English.

Furthermore, we demonstrated that when using ALFs combined with the baselines, their classification results are improved. The feature importance analysis evinces that the new proposed features are crucial to the impact on the classification model. Additionally, no individual feature significantly impacts all tasks, suggesting that the set of features is vital to the classification. The feature importance also unveils 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.

As limitations, we can highlight that our proposed approach depends on a lexicon that is adequate to the task and suitable enough for meaningful ALFs extraction to achieve satisfactory efficacy. 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. One of the limitations of the experiments is the use of Mikolov's embeddings to create audio-like features instead of contextual embeddings such as Bert and Elmo due to limitations on available infrastructure. Future work using Bert or Elmo on the flows creation is expected to deliver even more robustness to the text's representation.The empirical determination of an adequate number of frames for each dataset is also a limitation.

In future work, we intend to apply this method to other NLP tasks using more extensive texts. Another possible work is to evaluate the efficacy of associating ALFs to an attention mechanism.

Declarations

Authors' Contributions

Both authors conceived of the presented idea. The first author developed the theory, performed the computations, verified the analytical methods and carried out the experiment. Both authors discussed the results. The first author wrote the manuscript and the second revised it.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Availability of data and materials Code and preprocessed data are available on https://github.com/larissalucena/AudioLikeFeatures

References

Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer Publishing Company, Incorporated, 1st edition. DOI: 10.1007/978-3-319-73531-3.

Aggarwal, C. C. and Zhai, C. X. (2012). *Mining Text Data*. Springer Publishing Company, Incorporated. DOI: 10.1007/978-1-4614-3223-4.

Aker, A., Gravenkamp, H., Mayer, S., Hamacher, M., Smets, A., Nti, A., Erdmann, J., Serong, J., Welpinghus, A., and Marchi, F. (2019). Corpus of news articles annotated with article level subjectivity. Available at:https://www.researchgate.net/profile/Ahmet-Aker-3/publication/334050945_Corpus_of_News_Articles_Annotated_with_Article_Level_Subjectivity/links/5d14841892851cf4404f2092/Corpus-of-News-Articles-Annotated-with-Article-Level-Subjectivity.pdf.

Alías, F., Socoró, J. C., and Sevillano, X. (2016). A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6(5). DOI: 10.3390/app6050143.

Amorim, E., Cançado, M., and Veloso, A. (2018). 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. Association for Computational Linguistics. DOI: 10.18653/v1/N18-1021.

Araque, O., Zhu, G., and Iglesias, C. A. (2019). A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, 165:346 – 359.

Asr, F. T. and Taboada, M. (2019). Big data and quality data for fake news and misinformation detection. *Big Data & Society*, 6(1):2053951719843310. DOI: 10.1177/2053951719843310.

Avanço, L. V. and Nunes, M. d. G. V. (2014). Lexicon-based sentiment analysis for reviews of products in brazilian portuguese. In *2014 Brazilian Conference on Intelligent Systems*, pages 277–281. DOI: 10.1109/BRACIS.2014.57.

Baert, G., Gahbiche, S., Gadek, G., and Pauchet, A. (2020). Arabizi language models for sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 592–603, Barcelona, Spain (Online). International Committee on Computational Linguistics. DOI: 10.18653/v1/2020.coling-main.51.

Bao, L., Lambert, P., and Badia, T. (2019). 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. Association for Computational Linguistics. DOI: 10.18653/v1/P19-2035.

Bhowmik, N. R., Arifuzzaman, M., and Mondal, M. R. H. (2022). Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms. *Array*, 13:100123. DOI: 10.1016/j.array.2021.100123.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*. DOI: 10.1162/tacl$_{a0}$0051.

Briskilal, J. and Subalalitha, C. (2022). An ensemble model for classifying idioms and literal texts using bert and roberta. *Information Processing Management*, 59(1):102756. DOI: 10.1016/j.ipm.2021.102756.

Cho, Y. D., Kim, M. Y., and Kim, S. R. (1998). 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. DOI: 10.1109/ICASSP.1998.675336.

Choi, Y. and Wiebe, J. (2014). +/-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. Association for Computational Linguistics. DOI: 10.3115/v1/D14-1125.

Davis, J. and Goadrich, M. (2006). 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. ACM. DOI: 10.1145/1143844.1143874.

Dev, S., Li, T., Phillips, J. M., and Srikumar, V. (2020). On measuring and mitigating biased inferences of word embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7659–7666. DOI: 10.1609/aaai.v34i05.6267.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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. Association for Computational Linguistics. DOI: 10.18653/v1/N19-1423.

Dharma, E. M., Gaol, F. L., Warnars, H., and Soewito, B. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification. *J Theor Appl Inf Technol*, 100(2):31. Available at:https://www.jatit.org/volumes/Vol100No2/5Vol100No2.pdf.

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923. DOI: 10.1162/089976698300017197.

El-Maleh, K., Klein, M., Petrucci, G., and Kabal, P. (2000). 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. DOI: 10.1109/ICASSP.2000.859336.

Farrús, M., Hernando, J., and Ejarque, P. (2007). Jitter and shimmer measurements for speaker recognition. pages 778–781. DOI: 10.21437/Interspeech.2007-147.

Feldman, R. and Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. DOI: 10.1017/CBO9780511546914.

Filatova, E. (2017). 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. Available at:https://cdn.aaai.org/ocs/15480/15480-68660-1-PB.pdf.

Freitas, C. (2013). Sobre a construção de um léxico da afetividade para o processamento computacional do português. In *Rev. bras. linguist. apl.*, volume 13, pages 1031–1059. DOI: 10.1590/S1984-63982013005000024.

Fu, X., Yang, J., Li, J., Fang, M., and Wang, H. (2018). Lexicon-enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 6:71884–71891. DOI: 10.1109/ACCESS.2018.2878425.

Gasparetto, A., Marcuzzo, M., Zangari, A., and Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, 13(2). DOI: 10.3390/info13020083.

Ghanem, B., Ponzetto, S. P., Rosso, P., and Rangel, F. (2021). 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. Association for Computational Linguistics. DOI: 10.18653/v1/2021.eacl-main.56.

Giannakopoulos, G., Mavridi, P., Paliouras, G., Papadakis, G., and Tserpes, K. (2012). 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. Association for Computing Machinery. DOI: 10.1145/2254129.2254148.

Giannakopoulos, T. and Pikrakis, A. (2014). Chapter 4 - audio features. In Giannakopoulos, T. and Pikrakis, A., editors, *Introduction to Audio Analysis*, pages 59 – 103. Academic Press, Oxford. DOI: 10.1016/B978-0-08-099388-1.00004-2.

Goldberg, Y. and Hirst, G. (2017). *Neural Network Methods in Natural Language Processing*. Morgan and Claypool Publishers. Book.

Horne, B. D. and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news.

ISO/IEC (2002). *Information Technology - Multimedia Content Description In- terface - part 4: Audio.* ISO/IEC, Moving Pictures Expert Group, 1st edition. Available at:https://www.iso.org/standard/34231.html.

Jang, B., Kim, M., Harerimana, G., Kang, S.-u., and Kim, J. (2020). Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, 10:5841. DOI: 10.3390/app10175841.

Jeronimo, C., Campelo, C., Marinho, L., Sales, A., Veloso, A., and Viola, R. (2020). Computing with subjectivity lexicons. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association. Available at:https://aclanthology.org/2020.lrec-1.400/.

Jeronimo, C., Marinho, L., Campelo, C., Veloso, A., and Melo, A. (2019). Fake news classification based on subjective language. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*. DOI: 10.1145/3366030.3366039.

Jin, P., Zhang, Y., Chen, X., and Xia, Y. (2016). 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. Available at:https://frcchang.github.io/pub/ijcai16.peng.pdf.

Khadhraoui, M., Bellaaj, H., Ammar, M. B., Hamam, H., and Jmaiel, M. (2022). Survey of bert-base models for scientific text classification: Covid-19 case study. *Applied Sciences*, 12(6). DOI: 10.3390/app12062891.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. SpringerLink : Bücher. Springer New York. Book.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). 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. Available at:https://proceedings.mlr.press/v37/kusnerb15.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). 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. Morgan Kaufmann Publishers Inc. Available at:https://dl.acm.org/doi/10.5555/645530.655813.

Le, Q. and Mikolov, T. (2014). 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. Available at:https://proceedings.mlr.press/v32/le14.html?ref=https://githubhelp.com.

Lee, S.-W., Lee, J.-T., Song, Y.-I., and Rim, H.-C. (2010). High precision opinion retrieval using sentiment-relevance flows. pages 817–818. DOI: 10.1145/1835449.1835631.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., and He, L. (2022). A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2). DOI: 10.1145/3495162.

Li, T. and Ogihara, M. (2005). Music genre classification with taxonomy. volume 5, pages v/197 – v/200 Vol. 5. DOI: 10.1109/ICASSP.2005.1416274.

Liang, Q., Mu, J., Wang, W., and Zhang, B. (2017). *Communications, Signal Processing, and Systems: Proceedings of the 2016 International Conference on Communications, Signal Processing, and Systems*. Springer Publishing Company, Incorporated, 1st edition. Book.

Liu, Z., Wang, Y., and Chen, T. (1998). Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI Signal Processing*, 20. DOI: 10.1023/A:1008066223044.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.. DOI: 10.48550/arXiv.1705.07874.

Maharjan, S., Kar, S., Montes, M., González, F. A., and Solorio, T. (2018). 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. Association for Computational Linguistics. DOI: 10.18653/v1/N18-2042.

Mao, Y. and Lebanon, G. (2007). Isotonic conditional random fields and local sentiment flow. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 961–968. MIT Press. Available at:https://proceedings.neurips.cc/paper/2006/hash/32cbf687880eb1674a07bf717761dd3a-Abstract.html.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157. DOI: 10.1007/BF02295996.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. Available at:https://www.khoury.northeastern.edu/home/vip/teach/DMcourse/4_TF_supervised/notes_slides/1301.3781.pdf.

Mitrovic, D., Zeppelzauer, M., and Breiteneder, C. (2006). Discrimination and retrieval of animal sounds. In *2006 12th International Multi-Media Modelling Conference*, pages 5 pp.–. DOI: 10.1109/MMMC.2006.1651344.

Mitrović, D., Zeppelzauer, M., and Breiteneder, C. (2010a). 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. DOI: 10.1016/S0065-2458(10)78003-7.

Mitrović, D., Zeppelzauer, M., and Breiteneder, C. (2010b). 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. DOI: 10.1016/S0065-2458(10)78003-7.

Mutinda, J., Mwangi, W., and Okeyo, G. (2023). Sentiment analysis of text reviews using lexicon-enhanced bert embedding (lebert) model with convolutional neural network. *Applied Sciences*, 13(3). DOI: 10.3390/app13031445.

Muñoz, S. and Iglesias, C. A. (2022). A text classification approach to detect psychological stress combining a lexicon-based feature framework with distributional representations. *Information Processing Management*, 59(5):103011. DOI: 10.1016/j.ipm.2022.103011.

Pawar, A. and Mago, V. (2019). Challenging the boundaries of unsupervised learning for semantic similarity. *IEEE Access*, 7:16291–16308. DOI: 10.1109/ACCESS.2019.2891692.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project. Available at:http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Available at:https://aclanthology.org/D14-1162.pdf.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). 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. Association for Computational

Linguistics. DOI: 10.18653/v1/N18-1202.

Rabiner, L. and Schafer, R. (1978). *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall. Book.

Rabiner, L. and Schafer, R. (2010). *Theory and Applications of Digital Speech Processing*. Prentice Hall Press, USA, 1st edition. Available at:https://dl.acm.org/doi/abs/10.5555/1841670f.

Rabiner, L. R. and Schafer, R. W. (2007). Introduction to digital speech processing. *Found. Trends Signal Process.*, 1(1):1–194. DOI: 10.1561/2000000001.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. Available at:https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

Ramalingam, A. and Krishnan, S. S. (2006). Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting. *IEEE Transactions on Information Forensics and Security*, 1:457–463. DOI: 10.1109/TIFS.2006.885036.

Recasens, M., Danescu-Niculescu-Mizil, C., and Jurafsky, D. (2013). 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. Association for Computational Linguistics. Available at:https://aclanthology.org/P13-1162.pdf.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. In *PloS one*. DOI: 10.1371/journal.pone.0118432.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47. DOI: 10.1145/505282.505283.

Seo, J. and Jeon, J. (2009). 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. ACM. DOI: 10.1145/1571941.1572082.

Shannon, C. E. (2001). A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55. DOI: 10.1145/584091.584093.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). 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. Association for Computational Linguistics. Available at:https://aclanthology.org/D13-1170.pdf.

Tumitan, D. and Becker, K. (2013). Tracking sentiment evolution on user-generated content: A case study on the brazilian political scene. In *SBBD*. Available at:https://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/sbbd_shp_24.pdf.

Vasconcelos, L., Campelo, C., and Jeronimo, C. (2020). Aspect flow representation and audio inspired analysis for texts. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1469–1477, Marseille, France. European Language Resources Association. Available at:https://aclanthology.org/2020.lrec-1.183/.

Wachsmuth, H. and Stein, B. (2017). A universal model for discourse-level argumentation analysis. *ACM Trans. Internet Technol.*, 17(3):28:1–28:24. DOI: 10.1145/2957757.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). 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. Association for Computational Linguistics. DOI: 10.3115/1220575.1220619.

Wu, C., Wu, F., Liu, J., Huang, Y., and Xie, X. (2019). 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. Association for Computing Machinery. DOI: 10.1145/3357384.3357973.

Yu, D. and Deng, L. (2015). *Automatic Speech Recognition: A Deep Learning Approach*. Signals and Communication Technology. Springer, London. DOI: 10.1007/978-1-4471-5779-3.

Zhang, T. and Kuo, C.-C. (2001). Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4):441–457. DOI: 10.1109/89.917689.