

# Tools Based on Word Embedding to Make Easy the Analysis of Emotions in Spanish Text

Jorge Silva Pedreros, Alejandra Segura-Navarrete,  
Christian Vidal-Castro  
Information Systems Department  
Universidad del BíoBío, UBB  
Concepción, Chile  
email: jorge.silva1602@alumnos.ubiobio.cl,  
asegura@ubiobio.cl, cvidal@ubiobio.cl

Claudia Martínez-Araneda  
Computer Science Department  
Universidad Católica de la Santísima Concepción, UCSC  
Concepción, Chile  
email: cmartinez@ucsc.cl

**Abstract**— In the context of the analysis of emotions of text obtained from the web and social networks, this article aims to describe the development process of a tool to support this analysis based on the embedding of words. The main motivation has to do with the need to offer a tool for text preprocessing and emotion analysis for the Spanish language integrated into a single web application. The web application for the analysis of emotions, called fastText-embedding-viewer integrates three modules: The first corresponds to an Application Programming Interfaces (API) called corpus-preproc for text preprocessing that includes a collection of functionality, such as whitespace normalization, modifier and mark removal, lowercase folding, punctuation trimming around words and words without alphabetic characters, and content extraction main HTML and Cascading Style Sheets (CSS); the second corresponds to a novel module that allows obtaining similar words and analogies based on word embedding; and the third enables the analysis of emotions via a supervised machine learning model created using fastText for an input sentence. Promising results were obtained in the preliminary evaluation of the web application.

**Keywords**-word embedding; analysis of emotions; fastText.

## I. INTRODUCTION

Currently, there are numerous works on the area of analysis of emotions, either applying a lexicon, machine learning (ML), or a hybrid approach. It has been shown that when ML approaches are combined with the use of a lexicon, the performance of the classifiers improves [1]. The analysis of emotions in texts has several associated subtasks, such as tokenization, part of speech (POS), lemmatization, stemming, and elimination of stopwords, for continuing with the detection of emotion itself. Nevertheless, there are numerous tools for the analysis of emotions, that is, detection of polarity {positive, negative, neutral}, aimed at knowing the preferences of a customer in relation to a product or service or various related topics. Most of the time, the support tools for the pre-processing and analysis tasks are not integrated, which makes it difficult to manage them, in addition to not being available for the Spanish language, which limits their work in a diversity of Corpus. Therefore, this work shows the development of a web application that integrates various modules, among which are the complete text preprocessing

and analysis of emotions for the Spanish language using the word embedding fastText algorithm [2][3].

Sections 2 and 3 discuss the relevant concepts of word embedding and related works where applications that perform an analysis of emotions similar to that documented in this work are included. Section 4 describes the tools that have been developed, e.g., preprocessing, and the architecture of the web application for detection of emotions based on the word embedding fastText algorithm. Then, we describe an evaluation of quality the web application. The acknowledgement and conclusions close the article.

## II. BACKGROUND

The analysis of emotion corresponds to the area of subjectivity analysis in AI that has the objective of classifying a text into one of the positive, negative or neutral categories. The analysis of emotions, on the other hand, attempts to classify a text into one or more emotions according to the selected taxonomy. The taxonomies of emotions (Figure 1) have the objective of simplifying the analysis of the large number of human emotions. Emotion is a multifaceted experience, typical of humans and animals [4], that combines consciousness, bodily sensation, and behavior, reflecting the personal significance of different situations [5]. As represented in Figure 1, there are several proposals for taxonomies to classify emotions. As the categories increase, the recognition of an emotion and, consequently, its automatic detection becomes more complex. The analysis of subjectivity in texts is usually based on lexicons or ML models. These two approaches allow the classification of texts into different emotions or feelings. The first uses annotated semantic lists or hierarchies to classify texts. In simple terms, the words of the lexicon present in a comment are used to detect the predominant emotions or feelings.

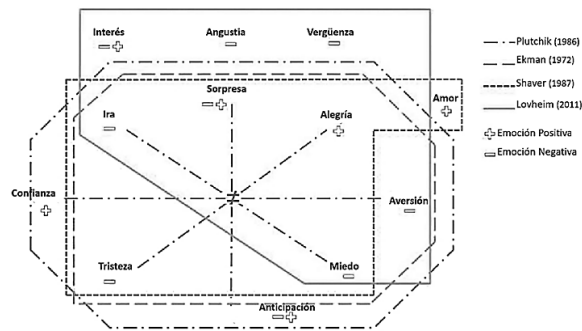


Figure 1. Representation of the four most common emotion taxonomies [6].

The oldest example is the General Enquirer [7], which uses a lexicon of positive and negative words to label the affective valence of texts. EmoLEX [8] is a lexicon of words labelled with the eight primary emotions of Plutchik [9]; RedPal is a semantic hierarchy developed by [10] that also uses Plutchik. ML-based analysis uses ML techniques to label emotions, e.g., the XED dataset [11], where the Support Vector Machine (SVM) technique and Bidirectional Encoder Representations from Transformers (BERT) are used to predict emotions through Plutchik.

There is also the possibility of combining the above approaches to create a hybrid model and improve the performance of automatic analysis. An example is Emo2Vec [12], where lexicons in Mandarin Chinese are used to postprocess word embedding, predicting a subset of emotions with the hourglass model [13]. Study [14] combines Global Vectors for Word Representation (GloVe) and Long Short-Term Memory (LSTM) to predict emotions with the Eckman taxonomy. It is also relevant to mention XED, which uses a corpus of manual development to develop an SVM model and fine-tuning for BERT.

In this study, we used the El Mercurio online (EMOL) unannotated Corpus for the unsupervised fastText model and the XED annotated Corpus for the supervised model.

### III. RELATED WORK

The analysis of emotions in texts has several associated subtasks, such as tokenization, POS, lemmatization, stemming, and elimination of stopwords, for continuing with the detection of emotion itself, which is a classification task using an approach based on lexicons, machine learning or hybrids. Nevertheless, there are numerous tools for the analysis of emotions, that is, detection of polarity {positive, negative or neutral}, aimed at knowing the preferences of a customer in relation to a product or service or various related topics. Among these are tools including MonkeyLearn, Lexalytics, Amazon Comprehend, Google cloud NLP, Aylien, MeaningCloud, Rosette, OdinText Text Analytics, Repustate, and BiText [15].

Within the literature review that we have carried out, we identify various tools that in addition to polarity, are capable of detecting the specific emotion that is reflected in an emotion intensity value, normally on a scale of -1 to +1, such

as Synesketch, Receptiviti, IBM Watson Natural Language, EMTk and Text2Emotion.

- **Synesketch:** Corresponds to an open-source engine for the recognition of textual emotions. Its entry corresponds to one or more sentences and analyses emotional content in terms of emotional categories based on the Ekman taxonomy, intensity, and polarity (valence). The recognition technique is based on a refined method of keyword detection that uses a word lexicon based on WordNet, a lexicon of emoticons, common abbreviations and colloquial expressions, and a set of heuristic rules. The recognition ends with a visualization of the emotions in an animated graphic. The emotional weight of each word and each emotional type is calculated as the ratio between the number of emotional synsets (of a given emotional type) and the number of all synsets to which the word belongs, decreased by the average penalty of all its emotional synsets [16]. For example, for the entry “I will not be lovesick!”, the output is a vector of weights [joy = 1,0, 0, 0, 0, 0], where negation processing and the amplifier were included.

- **Receptiviti:** API based on an emotion engine called SALLEE (Syntax-Aware Lexica L Emotion Engine) detects the emotions and feelings expressed in a text. It is designed to qualify the emotions that a person is expressing, which can include the emotions that they are feeling at present, emotions that they have felt in the past, those that they hope to feel in the future, or those that they see or assume that others are feeling. Each emotion can be defined as negative, neutral or positive. SALLEE is particularly effective in capturing emotions from social media posts, short text samples, casual language use and media such as conversations or text messages. For example, for the input “That was the best movie”, the output is a string in JSON format with the following content.

```
{
  "dictionary_measures":
  {
    "admiration": 0.2,
    "goodfeel": 0.2,
    "sentiment": 0.2
  }
}
```

where the calculation formula for one of the emotions (admiration) is described by (1):

$$\frac{\text{count}(\text{admiration})}{\text{count}(\text{goodfeel}; \text{badfeel}; \text{ambifeel}; \text{nonemotion})} \tag{1}$$

Equation (1) returns a score of 0.2 for the emotion of admiration, which means that approximately 20% of the statements were characterized by showing admiration. The goodfeel score for this sentence is

0.2, and the sentiment score is also 0.2 (on a scale of -1.0 to +1.0). This occurs because the word “best” expresses a positive emotion without amplifiers, softeners or negations. The score is relatively low because better is not a particularly strong emotion word [17].

- **Emotion Mining Toolkit (EMTk):** Is a set of modules and datasets that offers a comprehensive solution to extracting feelings and emotions from a text. The toolkit is written in Java, Python and R and is published under an open-source licence from MIT. The modules are based on previously developed tools, Senti4SD and EmoTxt [18], whose codes have been refactored and optimized, respectively. The emotion detection module includes classification models trained in our datasets and used for the detection of emotions from text. EmoTxt identifies emotions in a corpus of input in CSV format, with one text per line, preceded by a unique identifier. The result is a CSV file that contains the text ID and the predicted label for each element of the input collection. EmoTxt can also be used to train a new classifier from a set of data, for which it must have a collection of texts labelled with emotions [19].
- **IBM Watson™ Natural Language Understanding:** This web service uses linguistic analysis and machine learning to extract meaning and metadata from unstructured text data. It explores text data using text analysis to extract categories, classifications, entities, keywords, sentiments, emotions, relationships, and syntax [20]. For example, for the input in JSON format:

```
{ "html":
  "<html><head><title>Fruits</title></head><body
  ><h1>Apples and Oranges</h1><p>I love apples! I
  don't like oranges.</p></body></html>",
  "features": { "emotion": { "targets": [ "apples",
  "oranges" ]
  }
  }
}
```

The output in the same format is:

```
{ "language": "en",
  "emotion": {
    "targets": [{
      "text": "apples",
      "emotion": {
        "sadness": 0.028574,
        "joy": 0.859042,
        "fear": 0.02752,
        "disgust": 0.017519,
        "anger": 0.012855
      }
    }
  ],
  {
    "text": "oranges",
    "emotion": {
      "sadness": 0.514253,
      "joy": 0.078317,
      "fear": 0.074223,
      "disgust": 0.058103,
      "anger": 0.126859
    }
  }
}
] "document": {
  "emotion": {
    "sadness": 0.32665,
    "joy": 0.563273,
    "fear": 0.033387,
    "disgust": 0.022637,
    "anger": 0.041796
  }
}
```

- **Text2emotion:** This is a Python package to detect emotions from textual data. It processes any textual data, recognizes the emotions embedded in it, and provides the output in the form of a dictionary. It is well suited, with five basic emotion categories: happy, angry, sad, surprise, and fear [21].

A summary of all these tools is provided in Table I.

TABLE I. TOOLS FOR ANALYSIS OF EMOTIONS IN WRITTEN TEXTS

Features	Tools				
	<i>Synesketch</i>	<i>Receptiviti</i>	<i>EMTk</i>	<i>IBM Watson NL Understanding Text Analysis</i>	<i>Text2Emotion</i>
<b>Preprocessing</b>	No	Partially	Yes, it uses Stanford CoreNLP	Yes	Yes
<b>API available</b>	No	Yes	No	Yes	Yes
<b>Licence</b>	GNU	Commercial	MIT open source	Commercial	MIT open source
<b>Output</b>	Emotion intensity and visualization	Emotion intensity and valence	CSV (Text_id, emotion_label)	Emotion intensity	Emotion Intensity
<b>Analysis base on</b>	Word lexicon, emoticon lexicón, heuristic rules	Dictionaries	Emotion Lexicon from WordNet affect and supervised machine learning model	Linguistic and machine learning approach	Machine learning approach
<b>Languages supported</b>	English	English	English	English, French	English
<b>Authors</b>	[16]	[17]	[18][19]	[20]	[21]
<b>URL</b>	<a href="https://www.krca dinac.com/work/projects/synesket ch/">https://www.krca dinac.com/work/projects/synesket ch/</a>	<a href="https://www.receptiviti.com/emotions">https://www.receptiviti.com/emotions</a>	<a href="https://hub.docker.com/r/collabuniba/emtk">https://hub.docker.com/r/collabuniba/emtk</a>	<a href="https://www.ibm.com/demos/live/natural-language-understanding/self-service/home">https://www.ibm.com/demos/live/natural-language-understanding/self-service/home</a>	<a href="https://text2emotion.herokuapp.com/">https://text2emotion.herokuapp.com/</a>

Table I shows that the main motivation of this work is justified given the scarcity of preprocessing and emotion analysis tools existing for the Spanish language.

#### IV. TOOLS DEVELOPMENT

Next, the development of a text preprocessing API and a web application for the analysis of emotions and detection of analogies based on the fastText model is described. The web application operates as a client of the API.

##### A. Development of preprocessor module (API)

A common problem in Natural Language Processing (NLP), machine learning and any specialty that works with large volumes of plain text is the great variety of forms in which these are represented and expressed. The development of an API to execute the ad hoc preprocessing task allows the integration of several functionalities to solve the problem of cleaning and normalizing the corpus. Among the functionalities are the following:

- Parallel processing of files in a directory.
- Text is automatically converted to UTF-8 if the original encoding is in the Encoding Standard.
- NKFC and whitespace normalization.

- Removal of modifiers and marks.
- Lower-case folding.
- Trimming of punctuation around words.
- Replace words with <unk> placeholder if they meet any of the following criteria:
  - Word has an at sign @.
  - Word lacks alphabetic characters.
  - Word has two punctuation chars in a row, such as http://.
- HTML code is parsed, and CSS selectors can be used to:
  - Remove undesired elements.
  - Insert newlines after paragraphs and line breaks.
  - Extract the main content of an HTML document.

Table II summarizes the functionalities and shows some examples, excluding coding conversion, since it only applies to the internal representation of the text:

TABLE II. PREPROCESSING TASKS

Task	Input	Output
Unicode normalization <sup>a</sup>	Ḥf0lA muNdó	H0lA muNdo
Parsing and cleaning HTML code	<p>uno<p> dos <script>3===2	uno\ndos
Tokenization	Lorem ipsum. Dolor sit	Lorem ipsum\nDolor sit
Removal punctuation marks	y foo dice, ... cómo (es posible)	y foo dice cómo es posible
Replacement of custom tokens <sup>b</sup>	El reporte 3442 de foo@example.com está en http://example.com/report	El reporte de está en
Lower-case folding	Zenda sobrevivió al COVID-19	zenda sobrevivió al covid-19

<sup>a</sup> This stage also includes the normalization of word and sentence separators.

<sup>b</sup> A replacement token is inserted to preserve the distance between words.

To complete this preprocessing phase, an ad-hoc tool was developed for the preprocessing of text and HTML corpus called corpus-preproc available from [22], this tool will be a module of the web application described below.

*B. Development of web application*

The architecture of the available application is based on the controller view model, as shown in Figure 2. Docker platform on Ubuntu 20.04, whose host has the following hardware.

- Processor: Intel®Xeon® CPU E3-1220 v3 @ 3.10 GHz.
- RAM: 16 Gb.
- Disk memory: 285 Gb.

1) *Back-end of application:* For back-end of application, the microservices architecture is used. Figure 2 shows that to guarantee a level of security, Cloudflare Access services are used, which allow authenticating users by sending an OTP key by email, in this case, only accepting the domain ubiobio.cl. We also use Cloudflare Tunnel, a service that hosts sites without opening ports, which allows hosting a site with a standard port using the infrastructure provided by the university. In addition, it is observed within the security layer that the supervised word embedding server (ServerfastText) has been launched.

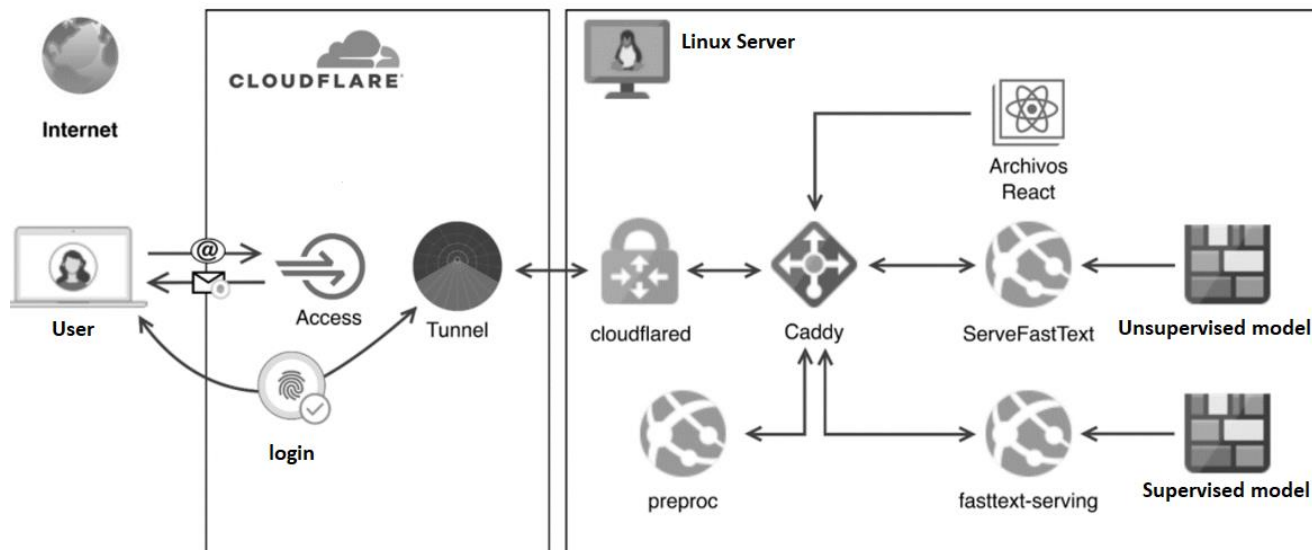


Figure 2. Web application architecture (back-end).

The reverse proxy and server for React assets (Caddy) and the Cloudflare client (Cloudflared) are static binaries. This makes provisioning very easy, requiring only the binaries, the compiled React files, the models and the configuration of Cloudflare and Caddy in a GNU/Linux distribution.

2) *Front-end of application:* As shown in Figure 3, the fastText-embedding-viewer application has three modules. The first is a preprocessor client (described in Section 4.A), the second is the word embedding module, where similar words and analogies are obtained, and the third module, oriented to the analysis of emotions, is where the emotional profile of a sentence, based on Plutchik’s taxonomy, is obtained. The view of front-end can be seen in Figure 3 available from [23].

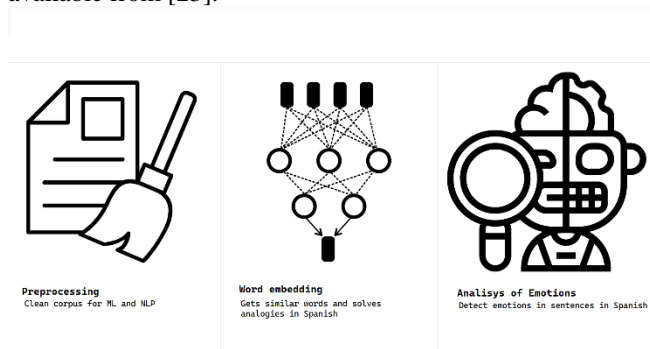


Figure 3. fastText-embedding-viewer application home page (front-end).

This interface offers to users preprocessing, similarities and analogies and analysis of emotions modules.

a) *Module for preprocessing:* This module allows uploading a corpus in TXT or HTML format with any acceptable coding within the web standard for preprocessing. The functions of Unicode normalization \*, cleaning HTML code, segmentation of grammatical sentences, removal of punctuation marks surrounding words, replacement of personalized tokens, normalization of uppercase and lowercase letters are available (see Figure 4).

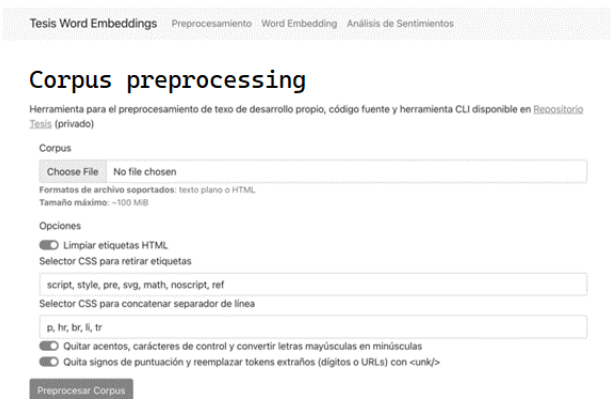


Figure 4. fastText-embedding-viewer preprocessor client module.

Word embedding module for similarities and analogies: This module allows the analysis of words using word embedding (see Figure 5), making it possible to perform similarity and analogy analyses, these functionalities are implemented by the unsupervised model generated from fastText. After entering the data for similarity, the fastText find the most related words a given word. The similarity of a word can support the search for synonyms during the composition of texts, while analogies can allow the discovery of new relationships between words.

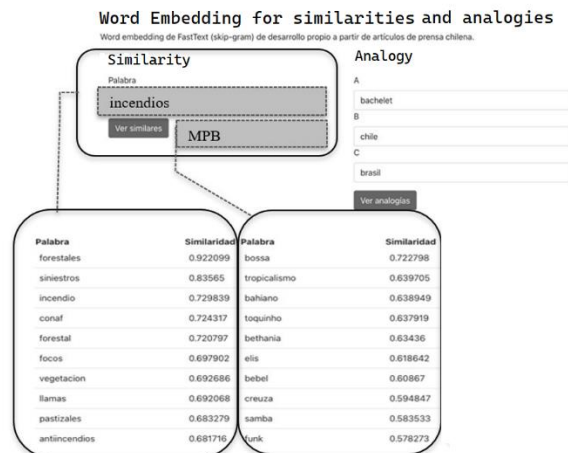


Figure 5. Word embedding module for similarities (Examples 1, 2 and 3).

From the input data of Figure 5, three examples are explained:

**Example 1:** When entering fires, similar words are obtained, such as forest or sinister, CONAF also arises, which is an organization dedicated to the control of forest fires.

**Example 2:** With MPB (Brazilian Popular Music), related musical genres such as Bossa Nova and prominent artists of the genre such as Elis Regina are observed.

**Example 3:** Writing Bachelet shows the name of the former president, the position of president and other presidents chilenos, such as Ricardo Lagos and Eduardo Frei.

In case of analogies, three parameters are entered in the analogy section, called A, B and C, which are read in the rhetorical form: B is to A as C is to prediction. The denomination arises from the vector form used:  $\underline{A} - \underline{B} + \underline{C} = \underline{prediction}$

From the input data of Figure 6, the following analogies emerge:

**Country-Celebrity:** Chile is to Bachelet as Brazil is to Rousseff. The most similar word is Dilma, with the second name being Rousseff.

Other outputs obtained were for the country-capital and position-gender analogies:

**Country-Capital:** Chile is to Santiago as Portugal is to Lisbon. The most similar word is Lisbon, also having a high similarity to the names associated with the population of Portugal.



Word Embedding for similarities and analogies

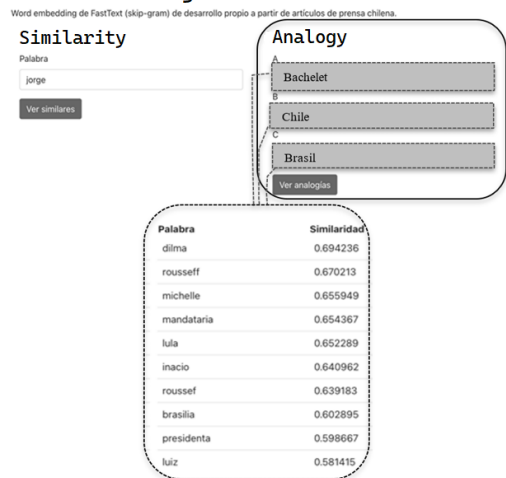


Figure 6. Screenshot with results for Country-Celebrity analogy.

b) *Word embedding module for Analisis of emotions using fastText:* It is possible to automatically determine the emotions associated with one or more sentences with word embedding. This functionality was implemented by the supervised model generated from fastText. The interface is illustrated in Figure 7. Note that the sentences can be entered by line or as a paragraph because the tool is responsible for separating paragraphs and cleaning the text by calling the preprocessing API to become an appropriate input to the fastText model. The output obtained corresponds to each sentence preprocessed, accompanied by a graph with the profile of predicted emotions and their percentage distribution.

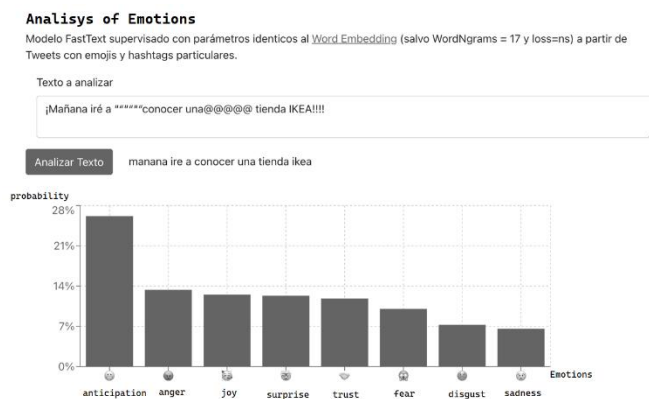


Figure 7. Module for the detection of emotions.

The output obtained corresponds to each sentence preprocessed, accompanied by a graph with the profile of predicted emotions and their percentage distribution.

V. EVALUATION

The evaluation of the tool involved the determination of the quality of the web application by considering aspects of

performance, accessibility, good practices, and Search Engine Optimization (SEO). For this, Google lighthouse tool was used, which produced the results (1-100 score) shown in Table III.

TABLE III. EVALUATION OF WEB APPLICATION

Module	Performance	Accessibility	Best Practices	SEO
Preprocessing	93	87	100	92
Word embedding for Similarities and Analogies	96	89	100	100
Word embedding for Analysis of emotions	96	90	100	100

Metrics shown in Table III include Performance related to first content paint, speed index, first idle CPU, estimated input latency, blocking time; Accessibility related to universal design; Best practices are related to the security of the web application and its documentation and SEO related to how optimized the site is for search engines.

Table III shows that the best results of the quality assessment were obtained for the word embedding similarity and analogies and analysis of emotions modules, which were slightly lower in the preprocessing module.

Making a comparison between our tool and those included in Table I, in addition to being oriented to the analysis of emotions for the Spanish language, it is based on the embedding of words through the fastText algorithm and integrates the functionality to identify synonyms and analogies through the similarity detection features offered by the algorithm.

VI. CONCLUSION AND FUTURE WORK

The tools that we have developed are useful for the tasks of researchers in the area of subjectivity analysis, specifically in the analysis of emotions in written texts. It is necessary to highlight the functionality of determining similarities and analogies as a novel aspect of our work.

The first module of the web application supports the area of subjectivity analysis by integrating a broad list of preprocessing tasks and considering input from HTML, Twitter or another source. The second, with the option of obtaining similarities, can support the search for synonyms during the process composition of texts, while analogies can allow the discovery of new relationships between words, all this through based on word embedding and unsupervised fastText model. Finally, the analysis of emotions module predicts an emotional profile of a written text, which could guide a commercial strategy based on the opinions of buyers or regulate the intensity of the statements on social networks, among other applications in the current world, all this through based on word embedding and supervised fastText model.

Both the API for text preprocessing corpus-preproc and the web application for analysis of emotions based on fastText are available from [22] and [23]. It is intended to add experimentation with users, since the tool is aimed at them. A benchmarking against other tools could be added.

It would be of interest to create a web browser extension to provide feedback on the predominant emotions in the text that a user writes or to hide comments or news articles in which negative emotions predominate. The incorporation of polysemy within the model will also be explored, such as that which appears in the representation of contextualized words to enhance the tool. Another aspect that could be incorporated into future work would be how a recommendation system [24] could be nourished from the results of the similarity module and analogies such as the analysis of emotions.

#### ACKNOWLEDGMENT

The authors thank the Department of Information Systems at the University of BioBío and the Research Directorate at the Universidad Católica de la Santísima Concepción.

#### REFERENCES

- [1] M. Lepe-Faúndez, A. Segura Navarrete, C. Vidal-Castro, C. Martínez-Araneda, and C. Rubio-Manzano, "Detecting aggressiveness in tweets: A hybrid model for detecting cyberbullying in the Spanish language," *Applied Sciences*, 11(22), 10706, 2021.
- [2] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," arXiv preprint arXiv:1607.01759, 2016.
- [3] P. Bojanowski, P. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, 5, pp. 135–146, 2017, doi: [https://doi.org/10.1162/tacl\\_a\\_00051](https://doi.org/10.1162/tacl_a_00051)
- [4] M. Bekoff. "Animal Emotions: Exploring Passionate Natures Current interdisciplinary research provides compelling evidence that many animals experience such emotions as joy, fear, love, despair, and grief—we are not alone". *BioScience* 50(10), pp. 861–870, 2000.
- [5] R. C. Solomon, and B. Duignan. Emotion [Encyclopedia]. *Encyclopædia Britannica*, Available from <https://www.britannica.com/science/emotion>, last accessed March 2023.
- [6] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, "Current state of text sentiment analysis from opinion to emotion mining," *ACM Computing Surveys (CSUR)* 50(2), pp. 1–33, 2017.
- [7] P.J. Stone, D.C. Dunphy, and M. S. Smith. "The general inquirer: A computer approach to content analysis," M.I.T Press.
- [8] S. Mohammad, and P. Turney. "Crowdsourcing a word–emotion association lexicon". *Computational intelligence* 29(3), pp. 436–465, 2013.
- [9] R. Plutchik. "A general psychoevolutionary theory of emotion". In *Theories of emotion*, pp. 3–33. Academic Press, 1980.
- [10] R. Varela. *Affective Lexical Resource to Spanish Language (Recurso Léxico Afectivo para idioma Español: REDPAL)*. Master Thesis. Universidad del Bío-Bío. [Online]. Available from [http://mcc.ubiobio.cl/docs/tesis/roberto\\_varela\\_medina-2020\\_varela\\_roberto.pdf](http://mcc.ubiobio.cl/docs/tesis/roberto_varela_medina-2020_varela_roberto.pdf), last accessed March 2023.
- [11] E. Öhman, M. Pàmies, K. Kajava, and J. Tiedemann, "XED: A Multilingual Dataset for Sentiment Analysis and Emotion Detection," In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6542–6552, 2020. <https://doi.org/https://doi.org/10.18653/v1/2020.coling-main.575>
- [12] S. Wang, A. Maolinyazi, X. Wu, and X. Meng, "Emo2Vec: Learning emotional embeddings via multi-emotion category," *ACM Transactions on Internet Technology (TOIT)* 20(2), pp. 1–17, 2020.
- [13] E. Cambria, A. Livingstone, and A. Hussain, "The hourglass of emotions," In *Cognitive behavioural systems*, pp. 144–157. Springer, Berlin, Heidelberg, 2012, doi: [https://doi.org/10.1007/978-3-642-34584-5\\_11](https://doi.org/10.1007/978-3-642-34584-5_11)
- [14] P. Gupta, R. Inika, B. Gunnika, and A. K. Dubey, "Decoding emotions in text Using GloVe embeddings," In *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 36–40. IEEE, 2021, doi: <https://doi.org/10.1109/ICCCIS51004.2021.9397132>
- [15] B. Saju, J. Siji, and A. Amal, "Comprehensive Study on Sentiment Analysis: Types, Approaches, Recent Applications, Tools and APIs," In *Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA 2020)*, pp. 186–193. IEEE, 2020.
- [16] U. Krcadinac, P. Pasquier, J. Jovanovic, and V. Devedzic, "Synesketch: An open-source library for sentence-based emotion recognition," *IEEE Transactions on Affective Computing* 4(3), pp. 312–325, 2013.
- [17] Receptiviti, Inc. Receptiviti's Emotions engine. [Online]. Available from <https://docs.receptiviti.com/frameworks/emotions>, last accessed March 2023.
- [18] F. Calefato, F. Lanubile, and N. Novielli, "EmoTxt: A toolkit for emotion recognition from text," In *2017 seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 79–80. IEEE, 2017.
- [19] F. Calefato, F. Lanubile, N. Novielli, and L. Quaranta, "EMTK: The Emotion Mining Toolkit," *SEmotion'19*, pp. 34–37, 2019, doi: <https://dl.acm.org/doi/10.1109/SEmotion.2019.00014>
- [20] IBM. IBM Watson Natural Language Understanding Text Analysis. [Online]. Available from <https://www.ibm.com/demos/live/natural-language-understanding/self-service/home>, last accessed December 2022.
- [21] A. Band. Text2emotion: Python package to detect emotions from textual data. [Online]. Available from <https://pypi.org/project/text2emotion/>, last accessed March 2023.
- [22] J. Silva. Corpus-preproc v0.1.0: A preprocessor for text and HTML corpora. [Online]. Available from <https://crates.io/crates/corpus-preproc>, last accessed March 2023.
- [23] J. Silva, A. Segura Navarrete. FastText-embedding-viewer: An emotion detect tool based on fastText [Online]. Available from <http://tesis.cuac.dev>, last accessed March 2023.
- [24] A.M.J. Skulimowski, "Cognitive Content Recommendation in Digital Knowledge Repositories – A Survey of Recent Trends," In: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, J. Zurada (eds) *Artificial Intelligence and Soft Computing. ICAISC 2017. Lecture Notes in Computer Science*, vol 10246. Springer, Cham, 2017. [https://doi.org/10.1007/978-3-319-59060-8\\_52](https://doi.org/10.1007/978-3-319-59060-8_52)