Text Representation Models

Miguel Ángel Banda Del Valle [1*], Jose Ángel Pertuz Montes [2*]

[1, 2] Department of Engineering, Universidad Tecnológica de Bolívar, Parque Industrial y Tecnológico Carlos Vélez Pombo Km 1 Vía Turbaco, Cartagena, Colombia
[*]mbanda@utb.edu.co
[*]pertuzj@utb.edu.co

Abstract: In Natural Language Processing (NLP) the text inputs to be processed need to go through certain transformation processes that lead to a representation that is understandable to machines. The quality of this representation depends to a great extent on the models used, since each of these models focuses on different assumptions that, when correctly implemented in NLP, result in the best use of NLP. This paper describes the most widely used text representation models and certain details to be considered when implementing them.

1. Introduction

Nowadays, the amount of information that is produced and stored daily in a digital format, in social networks, educational entities, informative organizations, among others, is of great significance. The content of this information represents data of interest to the world population in general and, to a greater extent, to companies and governments. Within this same area, the emergence of Text Mining has provided tools that lead to a better interpretation in decision making based on the available information.

This field addresses problems such as categorization and grouping of texts, identification of patterns, selection of information of interest, whose solution addresses the extraction and analysis of connections between concepts.

The structuring of the content of the texts in question is achieved by means of a representation model, on which the algorithms and identification processes applied depend.

In this paper, we will describe the most used text representation models, the principles on which each one is based and the possible deviations in the resulting interpretation.

2. Text Representation Models

Within the process known as text mining, which consists of analyzing and deriving information from texts, models are described that, with the use of computer tools, allow the extraction of information and the transformation of the text input into multiple characteristics that can be useful in subsequent tasks of this process; these are defined as Text Representation Models.

These models are currently used in different areas of computer science, such as for the development of algorithms for text classification, identification of relevant information, Machine Learning, Natural Language Processing (NLP), etcetera.

The performance or level of effectiveness based on the objective for which these models are used depends largely on inherent characteristics of the type of text being represented, given that the text inputs coming from each of these areas differ in semantic, syntactic, and lexical forms, bearing in mind that it is possible the presence of grammatical errors and semantically incorrect sentences that represent noise in the representation [1].

In the development of applications capable of analyzing sentiments, detecting topics, classifying documents according to their content, among others, it is of utmost importance to select a textual representation model that considers both syntactic and semantic information.

These are mainly divided into three groups, those based on vector space models, those based on graphs and those based on probabilistic models.

Vector space models (VSM) represent text entries through vectors of terms, so that each entry is identified as a vector of features in a space in which each dimension corresponds to different indexed terms, and in these, each component has a numerical value representing its importance.

Although there are several methods that try to study the syntactic and semantic structure of the text, most text mining applications assume that a text document can be represented by the set of words it contains.

3. Bag of Words

Bag Of Words is a natural language processing technique of text modeling that exposes the appearance of words in a document, it allows to obtain characteristics of the documents in a flexible and simple way thanks to the fact that it does not matter the order in which the words are or the grammatical form in which they are written, it only matters if the words appear and how many times they appear in the document.

The Bag of Words Model is used in algorithms since it makes it possible to convert texts of variable size into a vector of fixed size, i.e., it converts a text into its equivalent vector of numbers. This facilitates machine learning algorithms that work best when the inputs are of fixed size, well defined and structured.

It is a classification model whose strong point is its simplicity, since it demands little programming cost, and it is used in Natural Language Processing for text classification, in occasions where within a set of texts there is contextual information that is not relevant, simplicity turns out to be the best tool within this type of analysis.

4. Bag of N-Grams

The N-gram is one of the easiest notions to understand in the machine learning space. An N-gram is an order of N words. For example, "Bad comment", is a bigram, "The bad comment" is a trigram, and "It was a bad comment" is a 5-gram.

The use of N-gram features can be useful for improving classification performance of natural language processing, whether for automatic spell checking, sentence auto-completion, and takes field in the analysis of expressed sentiments, where words can be negated. For example. "No, I like your blouse" would represent a positive sentiment, but if we take it this way, "I don't like your blouse" would already be a negative sentiment.

After determinate the N-Grams, unigrams can be added to increase the data set.

5. TF-IDF (Term Frequency / Inverse Document Frequency)

One of the most widely used representation models is the frequency of occurrence of terms, which is a representation scheme of relative weights of a certain characteristic associated with a text entry. This relative value is calculated with the expression:

$$TF - IDF = TF(w) * IDF(w)$$

Where:

$$IDF(w) = \log\left(\frac{N}{df(w)}\right)$$

With TF(w) being the frequency of the term, IDF(w) the inverse frequency of text entries, i.e., the number of entries in which each word appears but in inverse form, df(W) is the frequency of text entries containing the word and N the number of entries in the corpus.

The TF measure indicates how important a term is within a text entry considering that all terms have equal importance. IDF indicates the relevance of a term within the whole corpus.

Thus, together, TF-IDF assigns a weight to each term, giving less value to terms of low relevance. Some research indicates that the correlation of frequency of occurrence of terms does not correctly determine the importance of terms within a document. [2]

There is variability in the extent to which TF and IDF values and analysis points are considered, but they all start from the idea that the value of a term within a document reflects the importance of that term.

6. Co-Occurrence Matrix

The Co-Occurrence matrix is a square matrix that describes the co-occurrence of two terms in a context, i.e., it analyzes the text in its context. Its purpose is to present the number of times that each specified entity in rows (ER) appears in the same context as each specified entity in columns (EC). [3]

Thanks to its good performance, this model is still popular today, having the ability to provide links between notions. [4]

There are two approaches by which Co-Occurrence matrix can be followed:

- *Term-context matrix e.g*. Where each sentence is represented as a context and if two terms appear in the same context, it means that they have occurred in the same context of occurrence.
- *K-skip-n-gram approach e.g*. Where a sliding window will consider the k+n words and this will serve as a context, terms that co-occur within this context are said to have co-occurred.

7. Word2vec

This model is used to produce word embeddings, which are an unsupervised learning tool, capable of capturing the context of a word within a text together, its semantic and syntactic similarity, relationship with other words, among other features [5].

This model provides a dense vector representation, and is shown to have high semantic significance, making it useful in NLP applications and network flow data analysis.

There are two main architectures for building Word2vec:

- *Continuous Bag-of-Words:* (CBoW), in which it seeks to predict words according to the context in which it appears.
- *Skip-Gram*: which seeks to perform prediction in the opposite direction to the previous architecture, starting from a keyword.

8. Transformer

What Transformers does mainly is to symbolize documents as vectors, it converts a text representation that is opaque into a representation that is clearer, i.e., compact, and abstract. For this it is necessary that the vector presents a constant size to make it easier to compare with other vectors obtained from other documents, having a more effective result than comparing words as normally does the full text search.

Transformers allows to adjust the statistical relationships between words in natural language, thanks to the families of neural networks that make it up. [6]

9. ELMO/BERT

1. *ELMO (Embedding from Language Models)*

It is a model that symbolizes words in vectors or embeddings, these are helpful in achieving state-of-the-art (SOTA) results in several NLP [7].

2. *BERT (Bidirectional Encoder Representations from Transformers):*

In Machine Learning algorithms, BERT can apply Transformer's bidirectional training, resulting in a model that has a deeper sense of context. It consists of only using Transformer's encoder mechanism because its main goal is to generate a language model.

It differs from models that read text inputs sequentially in that it can read all words at once, having more of a non-directional sense, thus the model can learn the context of a word based on the environment in which it is uttered.

10. References

[1] G. Giannakopoulos, P. Mavridi , G. Paliouras, G. Papadakis and K. Tserpes, "Representation Models for Text Classification:a comparative analysis over three Web document types," Athens, 2012.

[2] M. W. Berry, Ed., Survey of Text Mining, Clustering, Classification, and Retrieval, 2nd ed., New York: Springer-Verlag New York, 2004.

[3] Merours, "Stack Overflow," 6 June 2014. [Online]. Available: https://stackoverflow.com/questions/24073030/what-are-co-occurence-matrixes-and-how-are-they-used-in-nlp. [Accessed September 2021].

[4] A. Sehgal, "Medium," 11 July 2020. [Online]. Available: https://medium.com/@imamitsehgal/nlp-series-distributional-semantics-co-occurrence-matrix-31283629951e. [Accessed September 2021].

[5] R. Guerrero Álvarez, "Aplicación del modelo word2vec para el análisis de sentimientos en tweets en idioma inglés," UCLV, Santa Clara, 2019.

[6] T. Bach, "Codegram," 9 July 2020. [Online]. Available: https://www.codegram.com/blog/finding-similar-documents-with-transformers/. [Accessed September 2021].

[7] P. Joshi, "Analytics Vidhya," 11 March 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/. [Accessed September 2021].

[8] S. Alias, S. K. Mohammad, G. K. Hoon and T. T. Ping, "A text representation model using Sequential Pattern-Growth method," 2017.

[9] A. Rodríguez Blanco, A. Simón Cuevas, E. Guevara Martínez and W. Hojas Mazo, "Modelo de representación de textos basado en grafo para la minería de texto," *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal,* vol. 46, pp. 63 - 71, 2015.