ANALYSIS OF THE CONTEXT OF WORDS IN PORTUGUESE USING WORD2VEC

Alexandre D'Elia

delialexandre@poli.ufrj.com

Cidade Universitária, 21941-901, Rio de Janeiro, RJ, Brazil

Myrian C.A. Costa

myrian@ntt.ufrj.br

Cidade Universitária, 21941-901, Rio de Janeiro, RJ, Brazil

Nelson F.F. Ebecken

nelson@ntt.ufrj.br

Cidade Universitária, 21941-901, Rio de Janeiro, RJ, Brazil

Valéria M. Bastos

valeriab@ntt.ufrj.br

Cidade Universitária, 21941-901, Rio de Janeiro, RJ, Brazil

Abstract:

Due to the wide availability of textual documents on the Web, there was an intense study on how the machine should deal with related words and contexts. The usual methods of representing words as index in a vocabulary have become obsolete for new applications. Faced with this demand was developed a new way to study words and contexts efficiently. Word2Vec, presented in [1], is a group of templates used to produce word integrations. These models are two-layered neural networks trained to reconstruct linguistic contexts of words.Word2Vec takes as input a large text corpus, producing a vector space, typically containing several hundred dimensions, with each unique word in the body being assigned to a corresponding vector in space. The word vectors are positioned in the vector space so that words that share common corpus contexts are located close to each other in space.This paper aims to make an analysis of documents in contexts in the Portuguese language. For a general study on the language, a database of 37.5 million randomly selected Web pages was used. In this way, it became possible to observe the use of words based on the context in which they are inserted in an empirical way. Finally, according to tests, the performance was higher than expected.

1.introdution

With the rise of the digital age, there has been a change in data storage from physical to digital form. This process has already resulted in over 25 million digitized books, over two billion articles published every day, and over 130 trillion pages on the Internet. Due to the presence of this huge database in the digital area, it has become possible to use computational tools to better use this data in various projects and studies. In this sense, there was a great progress in computing in the area of data science and artificial intelligence. From that, companies and scholars get better efficiency in their operations and decision making, this area is getting increasingly in demand.

Among the different forms of action of artificial intelligence, the word representation methodology is one of the most developed in recent years. The use of usual word representation methods such as indices in a vocabulary has become obsolete for the latest computer applications. In this sense, the Natural Language Processing (NLP) area has developed methods of representing words in vector form. This methodology approaches vectorially words of similar meaning making possible translation operations, dictionary, text classification, among others.

With the development of word vectorization algorithms (word embeddings), several models obtained difficulties in the applicability due to its high operating cost. This happened because the linear processing and large vectors. In order to avoid this high cost, Word2vec [1] operates log-linear form and with the previously defined vector dimension. It allows representing the words of a set of texts in a vector space, calculated from the application of neural networks. This method improves the performance of learning algorithms in natural language processing tasks by grouping similar words.

2. Word2Vec

The Word2Vec is a group of models related to linguistic contexts. These models are based on superficial neural networks that use fewer layers, which receive as input a database in text format and return each word in the form of vectors arranged according to their contextual proximity. Its processing layer operates log-linearly, which makes it simpler than other nonlinear models. This feature allows for processing with a larger database much more efficiently. Within this group of models the Skip-Gram and Continous-Bag-of-Words (CBoW) were used in the study.

The Skip-gram model presented in [1] uses a target word as input of a log-linear classifier, with continuous projection layer, and seeks to predict the words next to the target word. Thus, the algorithm seeks, from the vector representation of words, to assemble the context of the sentence and thus try to predict the adjacent words.

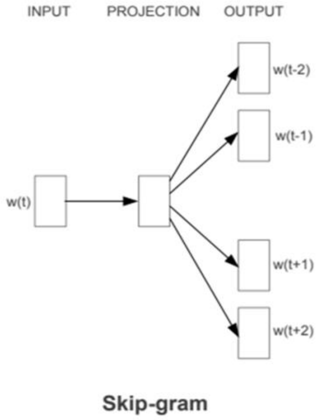


Figure 1. Skip-gram architecture
Source: Mikolov et al (2013a), [1].

The Continous Bag-of-Words (CBoW) model uses three layers (input, projection, output) to find a relationship between words. This learning method takes a window as a parameter within which it gets the words as input. Then, it makes a projection of the matrices generating the output vector and finally compares the result with the word corresponding to the position. Thus, CBoW acts contrary to Skip-Gram because it seeks, from the context, to find the word that best completes the meaning of the sentence.

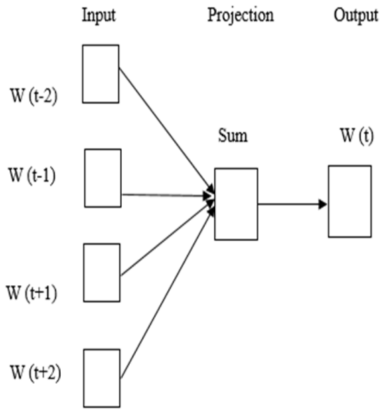


Figure 2. CBoW Architecture,

Source: Mikolov et al (2013a), [1]

2.1 Extensions models

To obtain a better result and a faster training process for both models, Negative Sampling [2] and Hierarchical Softmax [4] were presented.

2.1.1 Negative Sampling

Created from Noise Contrastive Estimation (NCE) by (Gutmann, et al, 2012) [3], Negative Sampling makes the model generation task more efficient by classifying a target word with a randomly selected context. While the model that does not use negative sampling has in its layer the weight equivalent to multiplying the number of words by the size of the vectors, Negative Sampling reduces the number of words to be multiplied, resulting in faster processing and better results.

2.1.2 Hierarchical Softmax

The Hierarchical Softmax was presented by Morin and Bengio [4] and its process works as a binary tree, as illustrated Figure 3.

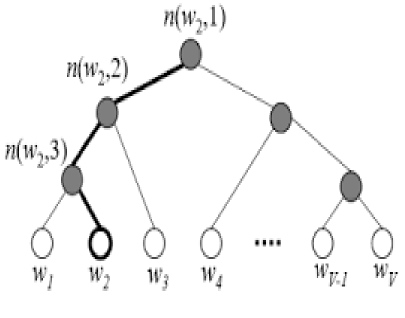


Figure 3. Binary tree

Source: https://becominghuman.ai/hierarchical-softmax-as-output-activation-function-in-neural-network-1d19089c4f49

In this structure, leaf nodes (nodes at the bottom of figure 3) represent words and inner nodes represent probabilities. In the figure example, the highlighted nodes and borders show the path from the root to an example of leaf node w2.

3 Experiment and preliminary results

3.1 Execution Environment

The machine used has 256 gigabytes of RAM, two processors with 10 core each Intel (R) Xeon (R) E5-2670 v2 CPU @ 2.70GHz and uses the operating system CentOS (6.10).

The Python 2.7 platform was used in the training stages and evaluation of the experiment results. For the training of the models the Gensim library was used.

3.2 About the code

For the study, the Python2.7 programming language was used. Within this environment we used the Gensim library [5], which is open source, and using machine learning in its processing. It is specifically designed to handle large text collections using data streaming and efficient incremental algorithms, which sets it apart from most other scientific software packages that target batch processing only.

Among the tools provided by the library are functions that enable vector space modeling using NumPy [6], SciPy [7] and, optionally, Cython [8] for performance. For the execution of the program the following functions were used:

- Word2Vec that allows various input parameters to determine the use of Skip-Gram or CBoW combined with Negative Sampling or hierachical Softmax, where the user can indicate the size of each vector and, in the case of Negative Sampling, how classification iterations must be made . In the study, it was used as a parameter for vectoring dimensions 200 words and 10 iterations where Negative Sampling was used .

-Most_Similar which receives two or more words as parameters and returns the most similar terms and the proximity between the vectors indicated by the cosine between them.

3.3 Database

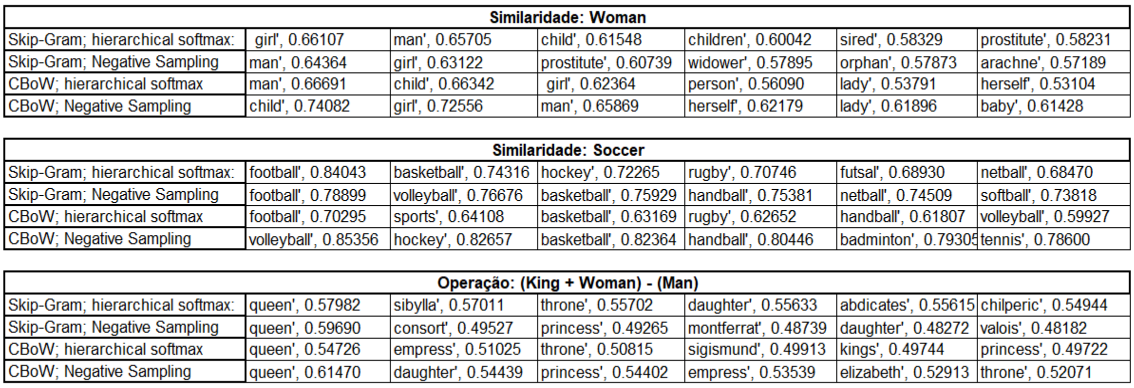
The database required for the study required the existence of an extensive vocabulary in Portuguese and data extraction done from different sources on random subjects. Meeting the requirements, Clueweb-09 [9] was chosen, which is a textual database collected between January and February 2009. The database is in use at Carnegie Mellon University (CMU), and its data in Portuguese were made available for the Never Ending Learning Language (NELL) project. Data extraction was realized through a crawler, which performed extractions from several bases guaranteeing a database of varied themes. From its extractions, Clueweb accumulates more than one billion web pages in 10 languages, approximately 40 million web pages in Portuguese. Our study used 25% of Clueweb09, with approximately 29 billion words.

For testing purposes, we used an English database, Text8, which has about 240,000 Wikipedia pages and approximately 17 million words, and its results are also presented.

3.4 Results in the Text8 Database

For the demonstration of the test results, a table was set up, in which the searched word is indicated at the top, on the left the configuration used and then the closest words found by the models, followed by the cosine between the vectors, indicating the level of proximity. To obtain the results with the English language, we used the Word2Vec function from the Gensim library with the explicit settings and the Most_similar function searching for the target words. The following table shows the results obtained with the Text8 database in their respective configurations.

Table 1: Results of the models for database Text8



Similaridade: Woman						
Skip-Gram; hierarchical softmax:	girl', 0.66107	man', 0.65705	child', 0.61548	children', 0.60042	sired', 0.58329	prostitute', 0.58231
Skip-Gram; Negative Sampling	man', 0.64364	girl', 0.63122	prostitute', 0.60739	widower', 0.57895	orphan', 0.57873	arachne', 0.57189
CBoW; hierarchical softmax	man', 0.66691	child', 0.66342	girl', 0.62364	person', 0.56090	lady', 0.53791	herself', 0.53104
CBoW; Negative Sampling	child', 0.74082	girl', 0.72556	man', 0.65869	herself', 0.62179	lady', 0.61896	baby', 0.61428

Similaridade: Soccer						
Skip-Gram; hierarchical softmax:	football', 0.84043	basketball', 0.74316	hockey', 0.72265	rugby', 0.70746	futsal', 0.68930	netball', 0.68470
Skip-Gram; Negative Sampling	football', 0.78899	volleyball', 0.76676	basketball', 0.75929	handball', 0.75381	netball', 0.74509	softball', 0.73818
CBoW; hierarchical softmax	football', 0.70295	sports', 0.64108	basketball', 0.63169	rugby', 0.62652	handball', 0.61807	volleyball', 0.59927
CBoW; Negative Sampling	volleyball', 0.85356	hockey', 0.82657	basketball', 0.82364	handball', 0.80446	badminton', 0.79305	tennis', 0.78600

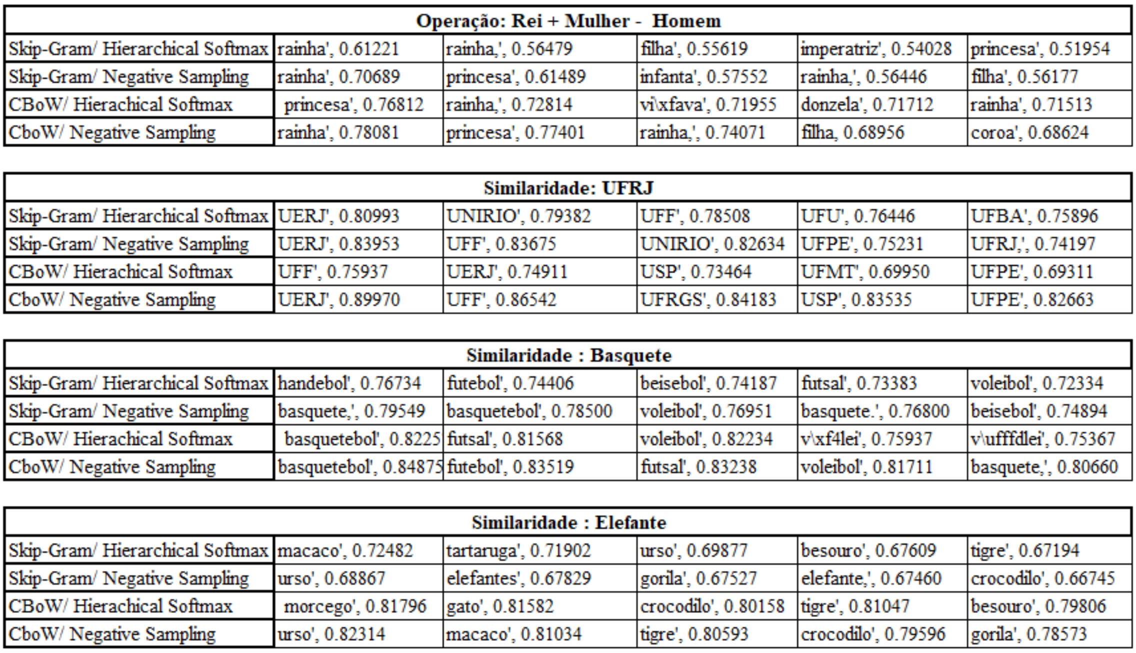
Operação: (King + Woman) - (Man)						
Skip-Gram; hierarchical softmax:	queen', 0.57982	sibylla', 0.57011	throne', 0.55702	daughter', 0.55633	abdicates', 0.55615	chilperic', 0.54944
Skip-Gram; Negative Sampling	queen', 0.59690	consort', 0.49527	princess', 0.49265	montferrat', 0.48739	daughter', 0.48272	valois', 0.48182
CBoW; hierarchical softmax	queen', 0.54726	empress', 0.51025	throne', 0.50815	sigismund', 0.49913	kings', 0.49744	princess', 0.49722
CBoW; Negative Sampling	queen', 0.61470	daughter', 0.54439	princess', 0.54402	empress', 0.53539	elizabeth', 0.52913	throne', 0.52071

From the results obtained by the test in the English language database, it is possible to observe proximity between the results obtained by each configuration, being the main result similar in most cases. It is also possible to observe a greater accuracy CBoW with Negative Sampling, which the results presented, owned a cosine value significantly higher than the others.

4. Results

For the demonstration of results with the Portuguese database, the assembled table has the same format as the one presented in the test results. The code used with the Portuguese database was the same as the code used in the tests. Table 2 shows the results obtained.

Table 2: Results of the models for database Clueweb9



Operação: Rei + Mulher - Homem					
Skip-Gram/ Hierarchical Softmax	rainha', 0.61221	rainha,', 0.56479	filha', 0.55619	imperatriz', 0.54028	princesa', 0.51954
Skip-Gram/ Negative Sampling	rainha', 0.70689	princesa', 0.61489	infanta', 0.57552	rainha,', 0.56446	filha', 0.56177
CBoW/ Hierachical Softmax	princesa', 0.76812	rainha,', 0.72814	vi\xfava', 0.71955	donzela', 0.71712	rainha', 0.71513
CboW/ Negative Sampling	rainha', 0.78081	princesa', 0.77401	rainha,', 0.74071	filha, 0.68956	coroa', 0.68624

Similaridade: UFRJ					
Skip-Gram/ Hierarchical Softmax	UERJ', 0.80993	UNIRIO', 0.79382	UFF', 0.78508	UFU', 0.76446	UFBA', 0.75896
Skip-Gram/ Negative Sampling	UERJ', 0.83953	UFF', 0.83675	UNIRIO', 0.82634	UFPE', 0.75231	UFRJ,', 0.74197
CBoW/ Hierachical Softmax	UFF', 0.75937	UERJ', 0.74911	USP', 0.73464	UFMT', 0.69950	UFPE', 0.69311
CboW/ Negative Sampling	UERJ', 0.89970	UFF', 0.86542	UFRGS', 0.84183	USP', 0.83535	UFPE', 0.82663

Similaridade : Basquete					
Skip-Gram/ Hierarchical Softmax	handebol', 0.76734	futebol', 0.74406	beisebol', 0.74187	futsal', 0.73383	voleibol', 0.72334
Skip-Gram/ Negative Sampling	basquete,', 0.79549	basquetebol', 0.78500	voleibol', 0.76951	basquete.', 0.76800	beisebol', 0.74894
CBoW/ Hierachical Softmax	basquetebol', 0.8225	futsal', 0.81568	voleibol', 0.82234	v\xf4lei', 0.75937	v\ufffdlei', 0.75367
CboW/ Negative Sampling	basquetebol', 0.84875	futebol', 0.83519	futsal', 0.83238	voleibol', 0.81711	basquete,', 0.80660

Similaridade : Elefante					
Skip-Gram/ Hierarchical Softmax	macaco', 0.72482	tartaruga', 0.71902	urso', 0.69877	besouro', 0.67609	tigre', 0.67194
Skip-Gram/ Negative Sampling	urso', 0.68867	elefantes', 0.67829	gorila', 0.67527	elefante,', 0.67460	crocodilo', 0.66745
CBoW/ Hierachical Softmax	morcego', 0.81796	gato', 0.81582	crocodilo', 0.80158	tigre', 0.81047	besouro', 0.79806
CboW/ Negative Sampling	urso', 0.82314	macaco', 0.81034	tigre', 0.80593	crocodilo', 0.79596	gorila', 0.78573

Even with the different settings, the return of the tool was related to the item of each search. In the first example, only the CBoW with Hierearchical Solftmax configuration did not achieve the expected result with the operation (Rei + Mulher – Homem), in which the intended return is the female of "Rei", and thus the word "Rainha". However, it is possible to note that in all configurations there was the occurrence of “Rainha,” because the character ',' was not excluded from the base, so the tool considered that “Rainha,” was different from “Rainha”.

In the search for words similar to the term “UFRJ”, both configurations with Skip-gram returned with the three closest results, containing several names of universities in the state of Rio de Janeiro. On the other hand, as with the Text8 database test, the CBoW configuration with Negative Sampling obtained, on average, better accuracy in calculating the similarity between the terms, presenting the names of public educational institutions related to UFRJ.

In the evaluation of the term “basketball”, CBoW with Negative Sampling again presents a better representation, since in this case there was an identification of terms that are exactly a representation of the searched term, such as “basketball,“ E ”basketball "But, as "," follows, it was considered to be a different element.

Another case that influenced the results was the occurrence of “v \ xf4lei”, notably as the representation of the term “vôlei”, which due to its accentuation, despite being considered as a corresponding term by the algorithm, was expressed in the output through of a coding different from the accent applied.

It is worth noting that the algorithm was first executed in various combinations to evaluate the tool, and it was concluded that the relationships between the search terms and the results presented were satisfactory. However, database preprocessing is already underway to remove special characters and punctuation.

5. Conclusion

Through the proposed experiment, it was possible to observe, regardless of the configuration and the absence of a preprocessing the results remained syntactically and semantically close to the searched terms. This feature demonstrates that it is possible to use large databases, but the processing has a high machine cost, consuming a lot of time to perform the whole process.

However, as noted in the results analysis, running Word2Vec without preprocessing resulted in the identification of similar terms, but due to the presence of a punctuation mark next to the word were not considered to be a single representation of the term. . This error interfered with the degree of similarity found by the model, since by separating the occurrence of the same word in two or more different situations (with and without the punctuation mark) the program understood that the terms were close and not the same. Although the preprocessing was not done at the data, for a first context evaluation the results were considered valid, being the task of cleaning the database considered of great importance to obtain more accurate results.

From obtaining similarity results of the various models, studies of various nature in various areas develop finding references never seen before due to the inability to process the data. Using the processing of study data, it is possible to increase the efficiency of translators, online and offline, by analyzing the context in which a term is inserted and allowing the replacement by another similar term of better understanding. Another way to use word vectoring in a Portuguese language context is to conduct studies with Portuguese as a database, so that researchers can observe materials or other terms never before related to the studied context. As an example of the importance and efficiency of this tool in academic projects is word2vec's prediction of new thermoelectric arrays through the relationship of materials to context [11].

Thus, PLN is extremely important in processing large volumes of data, enabling efficient results to be obtained, both in the area of research and development, as well as in the usual processes of companies and industries. Using Word2Vec, it can be concluded that this study demonstrates the viability of the tool for several applications in Portuguese language texts.

6. References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.Efficient estimation of word representationsin vector space.ICLR Workshop, 2013.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, Ilya Sutskever and Jeffrey Dean. Distributed Representations of Words and Phrasesand their Compositionality. NIPS 2013.

[3] Michael U Gutmann and Aapo Hyv ̈arinen. Noise-contrastive estimation of unnormalized statistical mod-els, with applications to natural image statistics.The Journal of Machine Learning Research, 13:307–361,2012

[4] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. InPro-ceedings of the international workshop on artificial intelligence and statistics, pages 246–252, 2005.

[5] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. ELRA, 2010

[6] Oliphant TE. A guide to NumPy. Vol. 1. Trelgol Publishing USA; 2006.

[7] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001

[8] Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn DS, Smith K. Cython: The best of both worlds. Computing in Science &amp; Engineering, 2011

[9] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya, "FACC1: Freebase annotation of ClueWeb corpora, Version 1 (Release date 2013-06-26, Format version 1, Correction level 0)", June 2013. (‘http://lemurproject.org/clueweb09/’)

[10] "NELL: Never-Ending Language Learning", Carnegie Mellon University, 2010

[11] Vahe Tshitoyan, John Dagdelen, Leigh Weston,Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder & Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. Nature News, 2019.