Exploring Word Embeddings

Nuno Gonçalves

June 2023

1 Introduction

Different approaches can be adopted to determine vector representations of words and to measure their quality. Word embeddings is an approach that has received much attention lately because of its ability to represent similar words as nearby points in a vector space. Each word is represented by a vector containing a number of features extracted from the word and its context, considered as a part of a text corpus. A word embedding is then a contextualized vector representation of a word. By using vectors to represent words, the similarity between pairs of words can be determined by computing, for example, the cosine distance between the two vectors. Such vectorized representations can thus provide efficient generalizations when the objective is to compare lexical items.

A common approach to obtain accurate and consistent word embeddings is to use neural networks that receive as input a text corpus. The larger the input text corpus, the better the quality of the model and thus of the generated word embeddings. A conveniently word embedding trained model can accurately identify how likely it is that two words will occur simultaneously or, in other words, how likely it is that the two words can be used interchangeably. Figure 1 illustrates this concept of co-occurrence of words and the relationships that can be established between words. For example, “cup” and “cups”, that are words semantically coherent, will have a high probability of co-occurrence, indicated by a low value of the cosine distance between their embeddings or vectors.

In this report we are going to explore in detail three different models: Hyperspace Analogue to Language (HAL) - where words within a pre-defined window are recorded as co-occurring with a strength inversely proportional to the number of other words separating them within the window; Continuous Bag of Words (CBOW) - where the model predicts the current word, from a window surrounding context words, by using both the n words before and after the target word w; Skip-Gram - where instead of using the surrounding words to predict the center word, it uses the center word to predict the surrounding words.

1.1 Word embeddings and topic modelling

1.1.1 Hyperspace Analogue to Language (HAL): This method was first introduce by Lund and Burgess in 1996 [1] and it was one of the first procedure where no explicit human judgments were required and the choice of axes was at least no longer arbitrary.

The primary principle employed is the notion of a "sliding window", a specific sequence of words across the corpus where the size is pre-defined although normal ranges go from 2 to 5 words. Within this window, the occurrence of words is counted, where the association strength between two words is dictated by their proximity within the window. Words appearing side by side are assigned the highest co-occurrence value (equal to the window size), whereas the strength gradually diminishes as the distance between the words increases.

As this window advances through the corpus by one word at each step, the co-occurrence values of the word pairs in the current window are logged. This iterative process ultimately results in a co-occurrence matrix, where each row and column corresponds to a unique word in the vocabulary. Each cell in this matrix signifies the aggregated co-occurrence count of a specific word pair. One important aspect is that this matrix distinguishes between word order in a pair - the sequence "xy" and "yx" count towards separate cells - therefore we have that, the rows show the co-occurrence for words preceding the target word, while the column the data for words succeeding it.

Finally, in order to get a word representation, the corresponding row/column pair for a word it is merged to produce a co-occurrence vector of length $2n$ (given nxn is the dimension of the matrix).

Table 1 shows an example matrix computed for the sentence "O João comeu o bolo do Miguel" using a window width of five words.

Vocabulary	'bolo'	'comeu'	'do'	'João'	'Miguel'	'o'
'bolo'	0.	4.	0.	3.	0.	7.
'comeu'	0.	0.	0.	5.	0.	4.
'do'	5.	3.	0.	2.	0.	5.
'João'	0.	0.	0.	0.	0.	5.
'Miguel'	4.	2.	5.	1.	0.	3.
'o'	0.	5.	0.	4.	0.	3.

Table 1: Co-occurrence Matrix for the phrase: "O João comeu o bolo do Miguel"

1.1.2 Continuous Bag of Words (CBOW): The CBOW model, as stated in [2], is an architectures for learning distributed representations of words while minimizing computational complexity. The reduction in computational complexity is achieved by removing the non-linear hidden layer, that is present in other Neural Net Language Models, and by sharing the projection layer for all words, meaning that all the context words get projected in the same position and then the result is averaged. This model receives a context (words around a given target word), then projects each word in the context to the embedding space, after that, the context words projections are averaged into a single vector which is used to predict the target word with a linear layer followed by a logsoftmax classifier.

1.1.3 Skip-Gram The Skip-Gram model is a type of word embedding technique, similar to Continuous Bag of Words (CBOW), but with a different approach to context prediction. While CBOW predicts a target word based on the surrounding context, Skip-Gram instead uses each current word as input and aims to predict the surrounding words within a certain range in the same sentence.

This model is essentially a log-linear classifier with a continuous projection layer. Its objective is to maximize the likelihood of predicting the correct context words. It has been observed that by increasing the range of words taken into account for prediction, the quality of the resulting word vectors can be improved (this fact will be important later on). This means the Skip-Gram model can generate better quality word embeddings when provided with a wider contextual scope.

Here is a schematic of both CBOW and Skip-Gram showing the differences between the two:

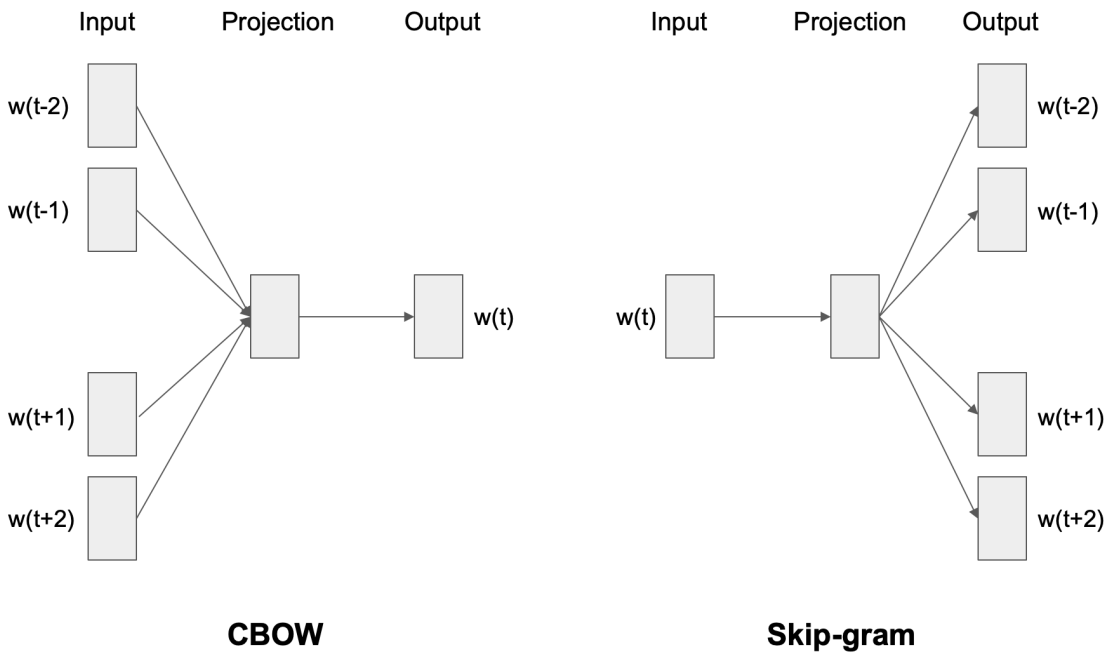


Figure 1: CBOW and skip-gram architectures.

2 Datasets

2.1 Training Corpus: The Portuguese dataset [3] used in this study was constructed through a detailed process involving web scraping, cleaning, and tokenizing text data from six reputable Portuguese online news websites. The collected data includes article titles, headlines, and the main body of the articles. It's important to emphasize that no text casing normalization was performed, which allows the model to differentiate words with case-sensitive meanings, such as "Porto" (a city in Portugal) and "porto" (meaning seaport in English). In addition the process included tokenization, removal of non-alphanumeric characters, and the elimination of Portuguese stopwords (like "mas", "e", etc..). The stopwords list was extended with overly frequent words in the dataset, particularly verbs and adverbs, which lacked significant definitional value.

Finally, the team[3] manually inspected the dataset for incorrectly formatted words resulting from HTML formatting errors. With all of this done the cleaned and processed data from the various sources were merged to form a comprehensive corpus consisting of 394,825,480 tokens or 33,372,416 phrases.

2.2 Word analogy evaluation (Test Set): It has been demonstrated that word embeddings created by models such as CBOW and Skip Gram exhibit an additional linear structure that captures the relation between pairs of word, thus, the use of simple vector arithmetic allows solving analogy queries such as "man is to king as woman is to?" In this example, "queen" happens to be the word whose vector $V(queen)$ is the closest approximation to the vector $V(woman)$ - $V(man)$ + $V(king)$.

Therefore, for the evaluation of the semantic reliability of the model, I have used the only publicly available word analogies for Portuguese (Rodrigues 2016)[4] which contains 17,558 questions in the form a is to b as c is to d and it is divided into semantic and syntactic sections. Therefore, in the test task, a well-performing model is expected to estimate the correct word d given vectors of words a, b and c, obtained from the linear operation $Wb + Wc$ - Wa, by estimating the most similar word vector to that operation. Some examples of this word analogies dataset are shown in table 2.

a	b	c	d
Lisboa	Portugal	Londres	Inglaterra
pai	mãe	avô	avó
alegre	alegremente	feliz	felizmente
agradável	desagradável	certo	incerto
Alemanha	Alemão	Portugal	Português
cão	cães	mão	mãos

Table 2: Examples of word analogies in the dataset

3 Model Exploration

3.1 Hyperspace Analogue to Language (HAL) In the HAL implementation, a window size of 7 and a minimum word occurrence threshold of 200 were utilized. Given the considerable memory requirements of HAL (due to its necessity to construct an $n \times n$ matrix and transform it into a $2n$ vector), only 1 million sentences were utilized in the analysis. Increasing the corpus size beyond this limit significantly hampered not only the training speed but also the inference capabilities, such as performing word arithmetic or identifying closest word neighbors. Despite its simplicity, the model exhibited impressive capabilities, particularly in terms of identifying word similarity. However, it demonstrated poor performance word arithmetic task, though that capability is beyond its original design scope. Some of the results obtained can be seen in Table 3.

3.2 HAL with Singular Value Decomposition (SVD) Due to the inherent memory and time limitations of the HAL model, I looked into alternative methods to scale the algorithm more effectively. Following the work developed in the first project, Singular Value Decomposition (SVD) with a target dimension of 300 was applied to try to capture the most significant relationships and mitigate the problems I was having. That allowed us to train on 20M sentences and scale the algorithm quite a bit. Though, to manage the considerable increase in unique words due to the larger corpus, the minimum word occurrence threshold was raised to 500. The results were very satisfactory, not only did we get better word representation by analyzing the similar words but the time and memory problems were in a way solved. Some word similarities obtained from this model variation are present in the Table 3.

5 Most Similar Words to 'Eurodeputados'					
HAL	emigrantes	empresários	bancos	artistas	filmes
SVD	deputados	hemiciclo	votação	estrasburgo	parlamentos

5 Most Similar Words to 'aquecimento'					
HAL	pacto	votação	montante	temperatura	limitar
H-SVD	poluição	fenómenos	desflorestação	eficaz	padrões

Table 3: Comparison of 5 most similar words to 'Eurodeputados' and 'Aquecimento' using HAL and H-SVD

The integration of SVD with HAL (H-SVD) improved word representation compared to HAL alone (Table 3). For instance, for 'Eurodeputados' and 'aquecimento', H-SVD returned more contextually relevant words.

HAL is straightforward to implement but is computationally demanding and ineffective in word arithmetic tasks. Although H-SVD mitigates some issues, providing better word representations and computa-

tional efficiency, both models fall short in detecting more complicated relationships compared to what one would expect from more complex models like CBOW and Skip-Gram. Though with even all of this, HAL and H-SVD showed to be useful for capturing semantic similarity in word embeddings.

3.3 CBOW I selected 10M sentences from the Portuguese dataset, which contained 114M words and 564k unique words, without differentiating between upper and lower case. Since I didn't have a lot of computational resources to train the model, I used a small vocabulary size of 20k words, choosing the most common words in my dataset sample. To build the context and target data, I discarded sentences containing words outside of the vocabulary. I ended up with 13M context/target pairs and created a test set with 20% of them.

I chose to use a window size of 4 words, both before and after the target word. I set the vocabulary size to 20k, the embedding dimension to 300, and the initial learning rate to 0.001, and trained the model for 5 epochs. I also added a maximum norm of 10 to the embedding dimension. This restriction acted as a regularization term, which helped prevent overfitting and limited the size of vectors in the embedding space.

The final training loss on the training set was 5.947 and on the test set was 6.150. The results obtained could still be improved by using the complete Portuguese dataset, by choosing a bigger embedding dimension size and a bigger vocabulary, but it would also take a lot more time to train the model.

To visualize and observe the relations that were preserved by the embedding space I used the cosine similarity metric to find words that were close to other words or groups of words.

"crime"	Similarity	"ronaldo"	Similarity	"mãe"-"mulher"+"homem"	Similarity
crime	1.000	ronaldo	1.000	homem	0.711
crimes	0.625	jogador	0.413	mãe	0.690
homicídio	0.401	avançado	0.380	**pai**	0.396
roubo	0.363	futebolista	0.348	bebé	0.395
mp	0.361	seleção	0.309	jovem	0.394
arguido	0.355	compatriota	0.308	filho	0.384

"futebol" + "encarnados"	Similarity	"futebol" + "leões"	Similarity	"ensino" + "jovens"	Similarity
encarnados	0.79	futebol	0.814	jovens	0.769
futebol	0.79	leões	0.812	ensino	0.766
benfica	0.647	**sporting**	0.643	**alunos**	0.544
leões	0.641	benfica	0.625	**estudantes**	0.535
sporting	0.595	encarnados	0.578	crianças	0.372
dragões	0.547	dragões	0.557	professores	0.369

The results were quite good when looking near a single words or near a group of two words, but more complex arithmetic operations were still not very well defined in the embedding space. With a larger model and dataset sample the quality of the arithmetic operations would also be expected to improve. Nonetheless, it was still able to capture some good results, for instance, for "mãe" - "mulher" + "homem".

To visualize the embedding space, in a 2 or 3 dimensional space, I used PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding). In figure 2, I reduced the dimensionality using PCA and projected the words "ps", "futebol", "mãe", "carne", "ensino" and for each word its respective 10 most similar words. In figure 3, instead of using PCA I used t-SNE to embed the embedding space in a two dimensional space.

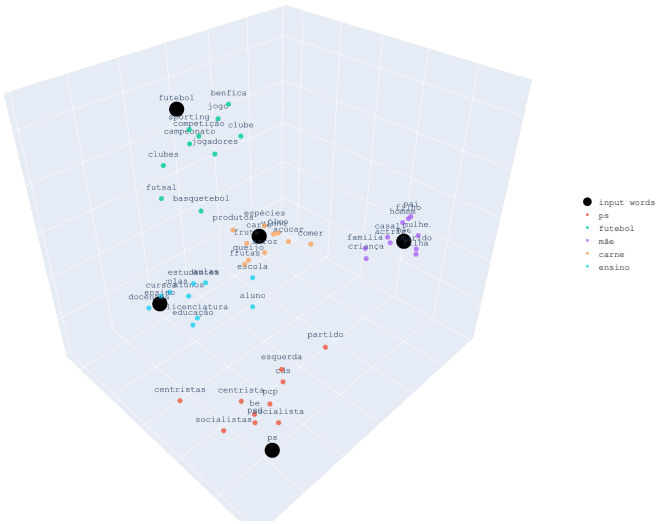
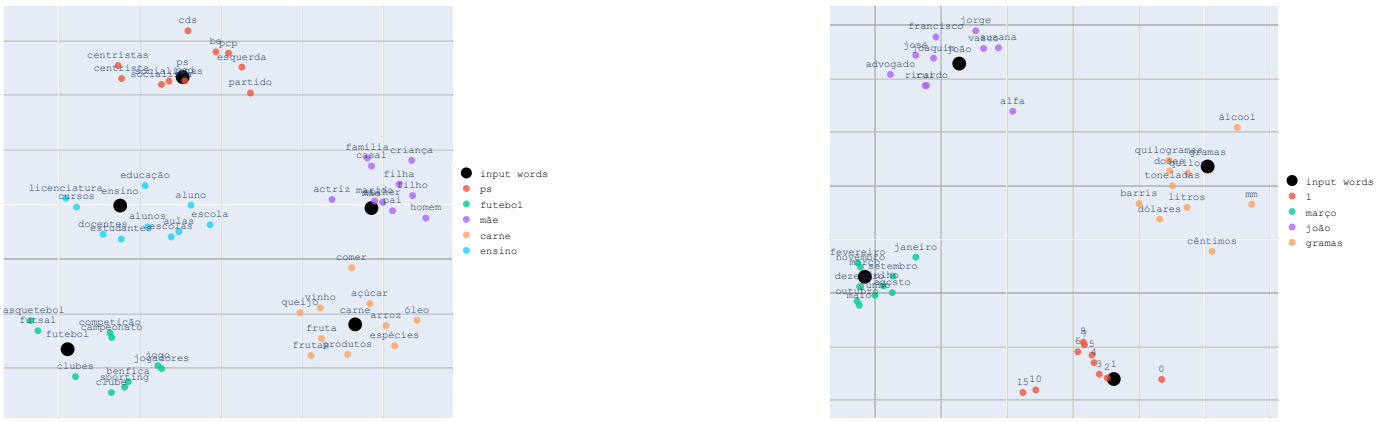


Figure 2: Part of the Embedding Space projected using 3 Principal Components



(**a**) (**b**)

Figure 3: Parts of the Embedding Space projected using t-SNE

In 3a there are lots of close word pairs like "competição","campeonato"; "socialista","socialistas"; "filha", "filho"; "cursos", "licenciatura"; etc. Near the word "carne" we can see that "comer" even though is related to "carne", it is a verb and for that reason it might be further away than other types of food like "arroz" and "fruta".

In 3b I projected the words "1", "março", "joão", "gramas" and their closest words. In this plot, we can see that all male first names, the months the numbers and the units of measure had, respectively, similar embedding vectors. Also, the numbers seemed to be ordered, from 1 up to number 8 they almost appear to make a ordered line, and the bigger numbers were to the left, "15, "10", while "0" was to the right.

Overall, the model seems to be able to capture relationships between similar types of words, like months, names, gender, sports, similar activities, numbers, related subjects, etc. Although not perfect, it is possible to observe some arithmetic in the embedding space working, especially when averaging two word embedding vectors to find common words to both of them.

4 Comparing CBOW and Skip Gram:

I then decided to go a little further and to compare the performance on a bigger scale of both CBOW and Skip Gram. For that I took advantage of the library *Gensim*, as it is flexible and intuitive to use, and enables handling large text files without having to load the entire file in memory which was a problem since the goal is to use the whole dataset.

One big difference that the *Gensim* library allows us to apply is the concept of Negative Sampling. This technique is used to reduce the computational burden during training by updating only a small percentage of the model's weights by treating the learning task as a binary classification problem, where the model learns to distinguish the target word from randomly chosen 'negative' words. This tehcnique not only allows for a better computational efficiency but also improved model performance on various semantic tasks.

For both models the following parameters were used: window distance of 5; a vector size of 300; an initial learning rate of 0.025; a threshold of 1e-5; a negative sampling of 15; and a total word frequency lower than 200 (minimum count). For these parameters, a vocabulary of 72,757 unique words was obtained. For a fair comparison and because the time the models take to sweep the whole dataset is very high, I trained the models for a total of 3 epochs. Though running for more epochs doesn't improve as much as one would expect, as reported by Mikolov et al., 2013[2], where they showed that training a model on twice as much data using one epoch gives comparable or better results than iterating over the same data for three epochs.

Both models were evaluated along four different categories in the same dataset. The first was unrestricted evaluation, considering all the words, the second is unrestricted but accepting if the correct word is in the top five closest words (the reason for this is that sometimes the word that appears first is simply a synonym of the correct word); a third - vocabulary restricted evaluation -, which ignores all questions containing a word not found in the top 30.000 words and finally the same but accepting if it appears in the first five words. The results are shown in the table 4.

Model	Unrestricted	Top 5 Unrestricted	Restricted	Top 5 Restricted
CBOW	39.6%	60.8%	46.2%	66.3%
Skip Gram	**46.7%**	**69.2%**	**52.7%**	**74.4%**

Table 4: Performance measure in accuracy (%) of CBOW and Skip Gram on Different Evaluation Categories

From the results shown in Table 4, it is evident that the Skip Gram model outperforms the CBOW model in all evaluation categories. One reason for Skip Gram's superior performance, particularly noticeable in the restricted vocabulary evaluations, could be its stronger focus on less frequent words during training. The Skip Gram model treats each context-target pair as a new observation, which allows it to learn better representations for less frequent words from larger datasets.

I also compared the performance of the best model - in this case Skip Gram - with the model presented in the original paper that introduced the analogy test set[4]. The results were the following:

Our model and Rodrigues 2016[4] were both evaluated using a Portuguese dataset, however, they differ significantly in size, with Rodrigues 2016 having a substantially larger corpus (about 1.7 billion tokens) compared to ours (approximately 394 million tokens).

Despite the smaller dataset, the results were competitive and in some aspects superior. In the unrestricted evaluation, where I considered the entire vocabulary, the model outperformed Rodrigues 2016[4], achieving

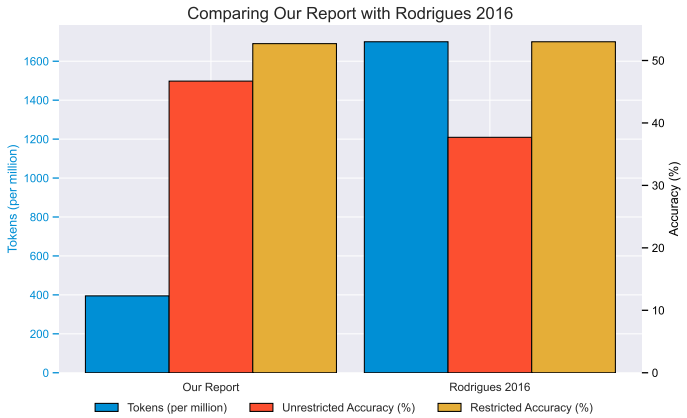


Figure 4: Comparison between our report and Rodrigues 2016[4].

an accuracy of 46.7% compared to their 37.7%. Though, this can probably be largely attributed to the selective approach in question evaluation: I only considered questions that included words present in the vocabulary, ensuring the model was evaluated in a context where it could draw upon its trained knowledge.

However, in the restricted evaluation, which was limited to the most frequent words, Rodrigues 2016[4] had a slight edge, scoring an accuracy of 53% versus my 52.7%. This similar performance suggests that both models are competent in representing high-frequency words. Despite the very close results in accuracy, it's important to note that my model achieved these competitive results with a significantly smaller training corpus. This highlights the value of a carefully curated dataset, which can lead to competitive results even with a smaller size, which goes to show the importance of quality over quantity in the dataset preparation.

In order to showcase some impressive capabilities I leave here some analogies I achieved using the Skip Gram model:

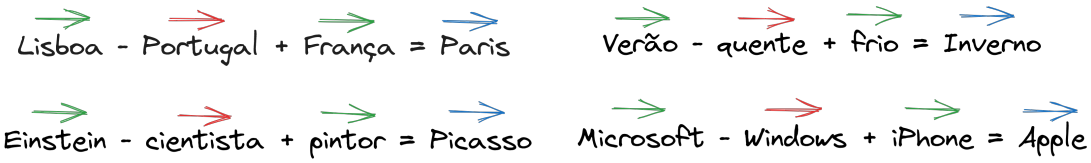


Figure 5: Analogies obtained from Skip-Gram model. Each analogy is represented by a mathematical formula: Word1 - Word2 + Word3 = Word4, where words are replaced by their corresponding vector representations in the word embedding space. The results show the model's ability to capture semantic relationships between words.

References

[1] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28:203–208, 1996.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[3] José Pedro Pinto, Paula Viana, Inês Teixeira, and Maria Andrade. Improving word embeddings in portuguese: increasing accuracy while reducing the size of the corpus. *PeerJ Comput Sci*, 8:e964, July 2022.

[4] João Rodrigues, António Branco, Steven Neale, and João Silva. Lx-dsemvectors: Distributional semantics models for portuguese. In *Computational Processing of the Portuguese Language: 12th International Conference, PROPOR 2016, Tomar, Portugal, July 13-15, 2016, Proceedings 12*, pages 259–270. Springer, 2016.