MASTER'S THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

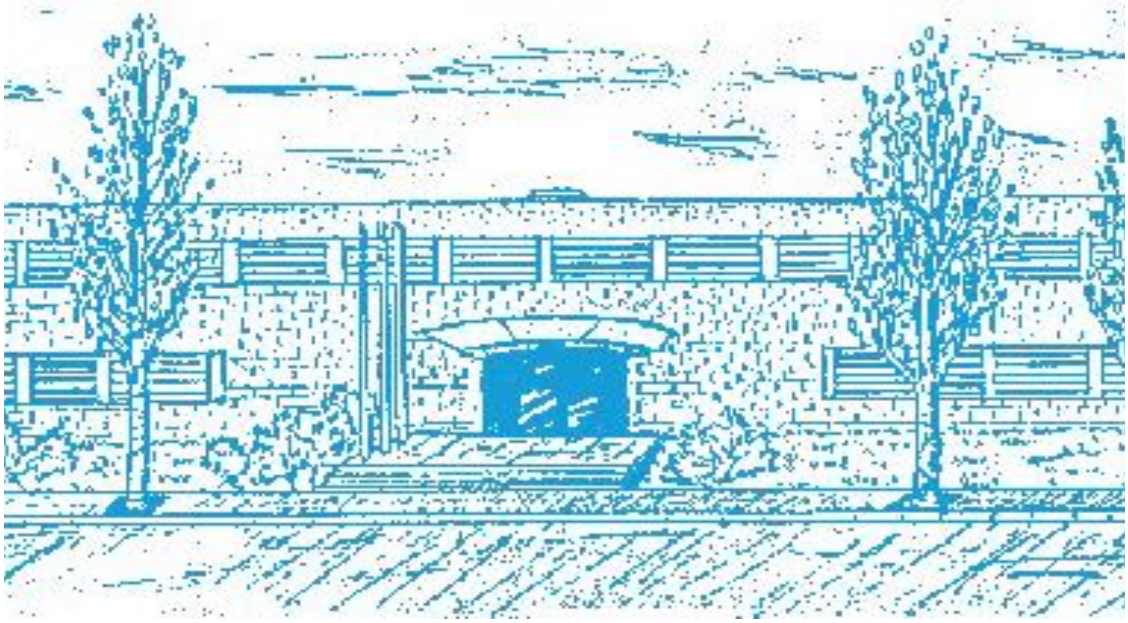
Title: An analysis of word embedding spaces and regularities

Author: Gijón Agudo, Manuel

Advisor: Cortés García, Claudio Ulises

Department: Ciències de la Computació

Academic year: 2018-2019



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya

Facultat de Matemàtiques i Estadística
Master's degree in Advanced Mathematics and Mathematical Engineering

Final Master Thesis

An analysis of word embedding spaces and regularities

Manuel Gijón Agudo

Supervised by Dario García Gasulla
and Armand Vilalta Arias
HPAI - BSC

Abstract

Keywords: Word embeddings, Embedding space, Distances, Semantic relations, WordNet
MSC2010: 68T01, 68T50

Word embeddings are widely use in several applications due to their ability to capture semantic relationships between words as relations between vectors in high dimensional spaces. One of the main problems to obtain the information is to deal with the phenomena known as the Curse of Dimensionality, the fact that some intuitive results for well known distances are not valid in high dimensional contexts. In this thesis we explore the problem to distinguish between synonyms or antonyms pairs of words and non-related pairs of words attending just to the distance between the words of the pair. We considerer several norms and explore the problem in the two principal kinds of embeddings, GloVe and Word2Vec.

Contents

Acknowledgment

I would like to thank Armand Vilalta Arias for his help during the last year, but but specially this last semester, without his advice this thesis would not have been possible . Also, I want to thanks to Dario García Gasulla and Raquel Leandra Pérez Arnall for their support during this time.

1 Introduction

Word embeddings have recently become a fundamental tool of Natural Language Processing, with application to tasks like machine translation or image annotation. The high-dimensional space defined by these embeddings is typically explored and exploited through distance-based operations. In this paper we work on the problem of finding words related between them in a text embedding. This relationship can be of different kinds, we focus in semantic relations like synonymy and antonym. We explore the idea of using the distance between norms instead of, like other authors has done before, the vector that units them. We present different norms, some of them well known in the literature and others not so widely used and we also introduce a new one with its theoretical mathematical framework. We also give an explanation of why them work properly or not and compare their performance on most used embeddings, GloVe and Word2Vec.

In recent years the use of neural networks has been increasing due to a significant improvement in the computational power of hardware resources, and to an explosion of digitised information. Deep learning methods (DL) for Natural Language Processing (NLP) are widely used, typically by representing words as vectors in a $\mathbb{R}^N$ space (the embedding space). There are different ways to obtain these embeddings, being neural networks trained with large corpus of text is the most frequent approach. Word embeddings capture different types of semantic relations between words [1], [9], [10]. These relationships are encoded in the resulting high dimensional space as geometrical relationships.
Word embeddings has many uses, from translation applications to caption image retrieval regularities. Another applications of the regularities presented in these embeddings is to evaluate the quality of these embeddings based of the relative position of words with similar meaning [8].

2 Objectives

The goals of this thesis are:

1. Prove that is possible, using a simple statistical test, to distinguish between related (synonyms or antonyms) or non related works attending just to the distance that separates them.

2. Check if it is possible to separate between synonyms or antonyms word pairs.

3. Study the performance of the most widely used norms for this task evaluated through a new proposed methodology.

4. Propose a new norm specially designed to obtain competitive performance in all considered situations.

5. Establish a comparison between the currently most widely used word embedding methodologies, GloVe and Word2Vec, based on the proposed evaluation methodology.

3 Methods

In order to archive our goals, we will be working under the following hypothesis:

1. We hypothesise that the vocabulary include in WordNet (WN) is sufficient to consistently evaluate the quality of the embedding.

2. We hypothesise that distances between pairs vectors have enough relevant information to assess semantic relations. In contrast, others authors have focused on the study of the relationship between two words attending to the characterisation of the vector (direction and relative position in the space) that unites them [9], [10].

3. We hypothesise that distances attending to individual component differences (*e.g.* Canberra, Braycurtis) are more suitable than distances that focus on the average change (*e.g.* euclidean, cosine, correlation).

The method of experimentation we use to test our hypothesis consists of the following steps:

First we gather a set of embeddings pre-trained from online resources. For each of these embeddings, we filter the corresponding vocabulary using the words present in WordNet. Once we define the relation to study, we use Wordnet to create two sets of word pairs based on their fulfilment of the relation. For each of these two sets we sample distances and then we compare them using the Kolmogorov-Smirnov (KS) statistical test. If the test is significant, we will be able to differentiate between related an unrelated pairs of words using the distance chosen. Repeating this process under different distances we are able to evaluate their usability as a measure of the semantic relationship in high dimensional spaces.

The KS test is the tool that we are going to use to compare our empirical distributions. The statistic measures a distance between the two samples giving us a way to know how different they are.

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{1}$$

We reject the null hypothesis (the two samples, $F_{1,n}$ and $F_{2,m}$, follows the same distribution) at a level of confidence α if:

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}}, \quad \text{where } c(\alpha) = \sqrt{-\frac{1}{2}\ln\alpha} \tag{2}$$

4 Distances

4.1 The curse of dimensionality

The curse of dimensionality is a term that refers to the fact that in high dimensional spaces the concept of proximity, distance or nearest neighbour may not even be qualitatively meaningful. Problems such as clustering, nearest neighbour search, and indexing suffer that phenomena and their approach must not be the same that in a low dimensional situation.

From [13] we know that the problem of finding the Nearest Neighbour to a given point is different as the dimension increase. This article shows that distance of the closest point and the farthest one tends to zero as the dimension grows, gives and explicit formula to compute the boundaries of the asymptotic behaviour and proves that this effect is important enough to be consider in dimensions greater than 15. In another work [14], the authors explore the effect of using L_p, defined in equation 3, distances applied to this

problem and find out that the value of p has a great impact in the way the dimension affects to the nearest neighbour and other similar problems.

Let $x = (x_1, x_2, \cdots, x_N)$ be a vector in $\mathbb{R}^N$. We define the L_p norm of this vector as follows:

$$L_p(x) = ||x||_p = \sqrt[p]{\sum_{i=1}^{N} |x_i|^p} \tag{3}$$

In [14] the authors prove that for resolve this problem (Nearest Neighbour) the optimal norm are the L_1 (also known as the "Manhattan distace" or "Taxicab distance"). They prove that for $p \geq 2$ the situation gets worse as p increases too. Notice that $p = 2$ gives us the Euclidean distance.

One interesting example on how the intuition seems to fail talking about high dimensional spaces is the volume of the sphere N-dimensional. The volume of a N-dimensional sphere of radius R is given by the formula:

$$V(N, R) = \frac{\pi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2} + 1\right)} R^N \tag{4}$$

Where Γ is the Gamma function, an extension of the factorial for non integers numbers. We can easily check that

$$\lim_{N \to \infty} V(N, R) = 0 \tag{5}$$

One way to study the asymptotic behaviour of the formula is using its Sterling approximation, with a small error $(1 + O(N^{-1})$ approximately), which is:

$$V(N, R) \sim \frac{1}{\sqrt{N\pi}} \left(\frac{2\pi e}{N}\right)^{\frac{N}{2}} R^N \tag{6}$$

In figure 1 we can see how the volume of a sphere n-dimensional of radio 1 decreases and tends to zero and in the next one (2) we can see how the effect of changing radio affects to this tendency.

In general, for L_p norms, the expression is the next one:

$$V(N, R, p) = \frac{\left(2\Gamma\left(\frac{1}{p} + 1\right) R\right)^N}{\Gamma\left(\frac{N}{p} + 1\right)} \tag{7}$$

Notice that 7 is equal to 4 in the case $p = 2$ (euclidean distance) since $2\Gamma\left(\frac{3}{2}\right) = \sqrt{\pi}$.

The main conclusion we can get from all this formulations is that no matter the radios of the sphere, as dimension grows the volume tends to zero.

The conclusions of these works (and others such that [15] and [16]) motivate the research of alternative metrics for this problem.

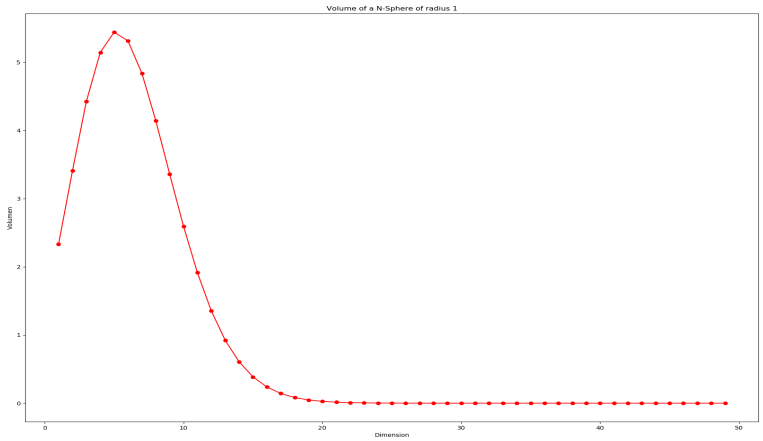


Figure 1: Volume of an N-dimensional sphere of radio 1 constant computed used the Sterling approximation. We can see how the volume tends to zero as the dimension of the sphere grows.

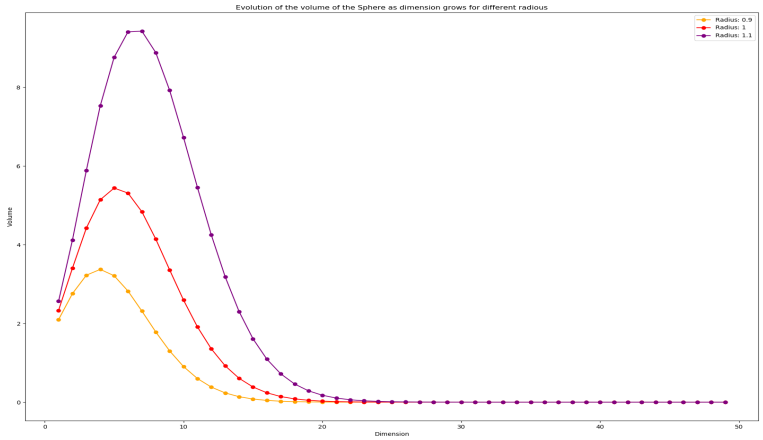


Figure 2: Volume of an N-dimensional sphere for different radio computed used the Sterling approximation. We can see how the volume tends to zero as the dimension of the sphere grows.

4.2 Well known distances

The most popular distances used in real spaces $\mathbb{R}^N$ are the euclidean (8) and the cosine (9). Beyond these we wish to explore other distance measures that are not typically evaluated in the literature. The choice of these distances is based on the last of the previous hypothesis, proposing distances than focus on the individual change of the components.

Let $u = (u_1, u_2, \cdots, u_N)$ and $v = (v_1, v_2, \cdots, v_N)$ be two vectors in $\mathbb{R}^N$, where N is the dimension of the embedding space. The well known distances that we will test are the next ones:

- Euclidean distance:

$$d_{\text{euclidean}}(u, v) = \sqrt{\sum_{i=1}^{N}(u_i - v_i)^2} = ||u - v|| \tag{8}$$

- Cosine distance:

$$d_{\text{cosine}}(u, v) = 1 - \frac{u \cdot v}{||u|| \cdot ||v||} \tag{9}$$

  ,where $|| \cdot ||$ is the euclidean norm of the vector ($||u|| = \sqrt{\sum_{i=1}^{N} u_i^2}$).

- Correlation distance:

$$d_{\text{correlation}}(u, v) = 1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{||u - \bar{u}|| \cdot ||v - \bar{v}||} \tag{10}$$

  ,where $\bar{u}$ is the mean of the components ($\bar{u} = \frac{1}{N}\sum_{i=1}^{N} u_i$).

- Canberra distance:

$$d_{canberra}(u, v) = \sum_{i=1}^{N} \frac{|u_i - v_i|}{|u_i| + |v_i|} \tag{11}$$

- Braycurtis distance:

$$d_{\text{braycurtis}}(u, v) = \sum_{i=1}^{N} \frac{|u_i - v_i|}{|u_i + v_i|} \tag{12}$$

4.3 Proposed distance

Attending to the difference between coordinates, we define this new distance.

$$d_{\text{proposed}}(u, v) = \text{Number of coordinates with different sign} \tag{13}$$

The mathematical definition of distance demands that $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$. In the case of the proposed distance this does not happen since two different vectors can be at distance 0 (*i.e.* take $x = (1, 1, \cdots, 1)$ and $y = (2, 2, \cdots, 2)$,

$p_{\text{proposed}}(x, y) = 0$). To fix that and make this mathematically rigorous we can define a relation of equivalence like this:

$$u\mathcal{L}v \iff sing(u_i) = sing(v_i) \;\; (\forall i = 1, , N) \tag{14}$$

where the sing function is defined as follows:

$$sing(x) = \begin{cases} 1 & , x > 0 \\ 0 & , x = 0 \\ -1 & , x < 0 \end{cases} \tag{15}$$

This equivalence classes will have this structure:

$$[x] = \{(x_1, ..., x_N) : x_i \in \{-1, 0, 1\} \; (\forall i = 1, ..., N)\}$$

where for a vector $u = (u_1, \cdots, u_N) \in \mathbb{R}^N \longmapsto [x]$, $u\mathcal{L}x$ in this way:

$$x_i = \begin{cases} 1 & , u_i > 0 \\ -1 & , u_i \leq 0 \end{cases}$$

Observe that we are dividing the $\mathbb{R}^N$ space in a total of 3^N equivalence classes.
Observation: we can define the distance proposed as follows, for $u\mathcal{L}x$, $v\mathcal{L}y$:

$$d_{\text{proposed}} : \mathbb{R}^N \times \mathbb{R}^N \longrightarrow [0, N] \subset \mathbb{N}$$

$$d_{\text{proposed}}(u, v) = d_{\text{proposed}}([x], [y]) = \begin{cases} 0 & , \sum_{i=1}^N x_i \cdot y_i \leq 0 \\ \sum_{i=1}^N x_i \cdot y_i & , \text{otherwhise} \end{cases}$$

Now we define the distance proposed not as a distance between vectors, otherwise as a distance between the equivalence classes this vectors belongs to under this relation of equivalence.

Result: distance proposed is a distance between the equivalence classes defined as above.

Proof:

We have to check four properties:

- $d_{\text{proposed}}(x, y) \geq 0$, this is true by the definition of norm.
- $d_{\text{proposed}}(x, y) = d_{\text{proposed}}([x], [y]) = 0 \iff [x] = [y] \iff x\mathcal{L}y$ ie. x and y belongs to the same class, so they are the same in this sense.
- $d_{\text{proposed}}(x, y) = d_{\text{proposed}}(y, x)$, immediate from the symmetry of the product.
- $d_{\text{proposed}}(x, z) \leq d_{\text{proposed}}(x, y) + d_{\text{proposed}}(y, z)$

$$0 \leq \sum_{i=1}^N x_i \cdot z_i \leq \sum_{i=1}^N x_i \cdot y_i + \sum_{i=1}^N y_i \cdot z_i = \sum_{i=1}^N y_i(x_i + z_i)$$

and observe that $x_i, y_i, z_i \in \{-1, 1\}$.

■

5 Resources

5.1 WordNet

WordNet [2] is one of the most used resources in Natural Language Processing and Representation Learning. The English version of this database includes information over 117,000 different synsets and their semantic relations(hyponymy, hypernymy, etc.). A synset is a set of terms that represent an unique idea or concept. All the words included in a synset are considered synonymous.

In the next figures we can see examples of how synsets are. In the first one, 3, we can see the information that the synset corresponding to the word "Hamburger" contains. In the next one, 4, we can appreciate how the set of synsets are related with the synset of the word "animal".

Hamburger

- Hamburger (an inhabitant of Hamburg)
  - direct hypernym:
    - German (a person of German nationality)
  - sister term
    - German (a person of German nationality)
      - East German (a native/inhabitant of the former GDR)
      - Bavarian (a native/inhabitant of Bavaria)
  - derivationally related form
    - Hamburg (a port city in northern Germany on the Elbe River that was founded by Chalemagne in the…)

Figure 3: Hamburguer synset.

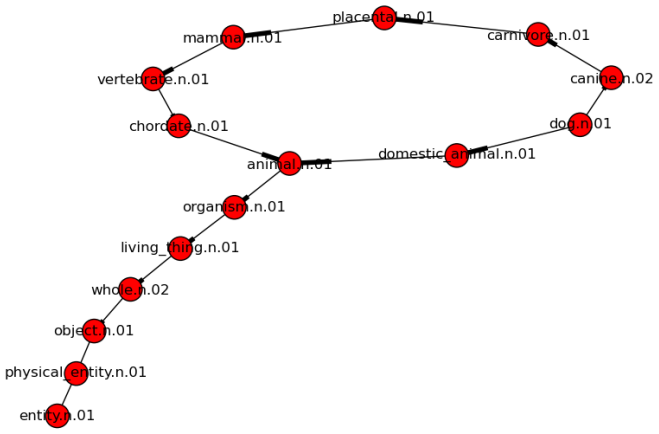


Figure 4: In this image we can see how the synsets in WordNet are related with the synset corresponding to the word "animal".

5.2 GloVe

The GloVe model (Global Vectors for Word Representation) was developed and introduced in 2014 [4] by researchers of Stanford University. In our case, we are using the embeddings `glove.6B.50d.txt`, `glove.6B.100d.txt`, `glove.6B.200d.txt` and `glove.6B.300d.txt` with embedding space dimensions 50, 100, 200 and 300 respectively and trained with the corpus Wikipedia 2014 and Gigaword-5[1].

In our experiments we use the embedding `GoogleNews-vectors-negative300.bin`. It is a model of dimension 300 and trained with the corpus "Google News dataset"[2]..

5.3 Word2Vec

Word2Vec methodology, created by Mikolov et al. [7], [8] and [9], automatically creates word embeddings from a corpus of text. Two algorithms are described that produce embeddings. The first one, Continuous Bag-of-Words [5], [6] is trained for predicting a target word given the words around it . On the other hand, the second one, Skip-Gram [5], [6], is to predict the words surrounding a given word.

5.4 WER (Word Embedding Research)

WER (Word Embedding Research) is a python package that I have created to solve the problem of replicate easier experiments in GloVe and Word2Vec embeddings. GLoVe and Word2Vec are very different in the sense that they are saved in different format and the ways to access to the information is specific for each. On one hand, GloVe embeddings are saved as text (`.txt`) files and we can work with them using the python basic functions, on the other hand to work with Word2Vec we have to use functions of the package `gensim`.

Our package WER provides a common framework for both embeddings. This package provides the funcionalities required for all the experiments in the present work.

6 Experiments

We are going to use the vocabulary and embeddings obtained from previously described resources. For all the GloVe embeddings, we have the same vocabulary set, compose of $400,000$ terms of which, $55,666$ terms are also present in WordNet (13.92%, figure 5). .In this case, this results in $28,763$ sets of synonyms, that include a total of $30,439$ different words.

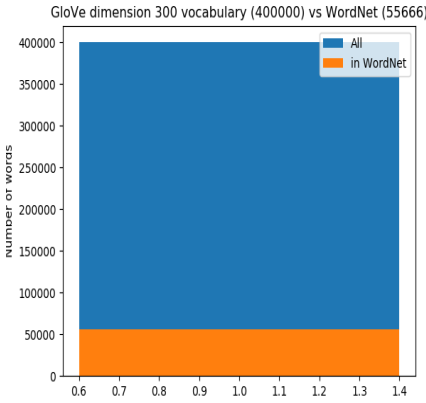
For the Word2Vec case, we have a total of 3 millions of words of which, only $54,586$ of them are in WordNet (1.82%, figure 5). In the case of Word2Vec embedding, there are a total of $31,886$ sets of synonyms, with a total of $33,822$ different words.

As we can see, the number of synonyms available in both embeddings is quite similar. For the antonyms we follow a similar procedure.

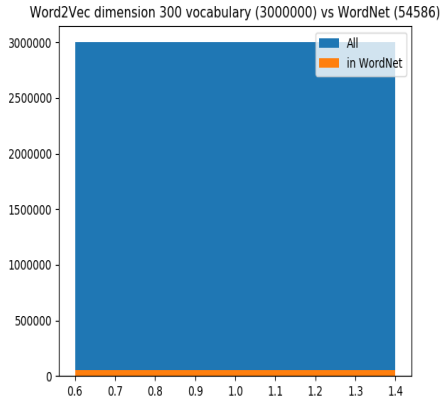
The set of non related words can include or not the words that are not present in WordNet. Both possibilities are studied. The size of the sample of random pairs taken from the non-related ones is limited to $5,000$ for both, filtered and non-filtered vocabularies.

For each of the embeddings considered we do two rounds of experiments: synonyms and antonyms against vocabulary filtered by WordNet, and synonyms or antonyms against

[1]These embedings are available in the web adress: `https://nlp.stanford.edu/projects/glove/`
[2]This embeding is available in the web adress: `https://code.google.com/archive/p/word2vec/`



(a) Proportions of the words presented in the vocabulary of GloVe embeddings (dimensions 50, 100, 200 and 300) that are included in WordNet



(b) Proportions of the words presented in the vocabulary of Word2Vec embedding that are included in WordNet

Figure 5: GloVe dimension 300. Synonyms and non related words. All Vocabulary. Best and worst distance distributions (excluding Euclidean distance).

all the vocabulary. In each round of experiments, we test each one of the six distances defined previously.

We evaluate this results based on the KS value and the p-values associated. We use $\alpha = 0.05$ as significance value for the test in all the cases.

7 Results

7.1 About the optimal norms in dimension 300

Due to the level of significance and the results, we must reject the null hypothesis in all the cases (*i.e.* the two distributions are not identical).

The tables 1 and 2 contains results of experiments comparing synonyms with random words for the GloVe and Word2Vec embeddings respectively. The results of the experiments for the antonyms in GloVe and Word2Vec embeddings are included in tables 3 and 4.

The first result is that the Euclidean distance is the less capable to distinguish between synonyms or antonyms from random pairs of words. This result is align with the literature. In general, Cosine distance is among the best candidates in all experiments except GloVe non-filtered. Given this poor results, euclidean distance will not be considered in the rest of the comments in this section.

We can see that in general, Word2Vec embedding is better in the task of discriminating between related and non related pairs of words for all the considered distances in both filtered and non filtrated schemes. The difference is important up to the point that the best performing distance in GloVe is worse than the worst performing distance in Word2Vec.

Over the results, we can see that for Word2Vec the best performing distances are Cosine and Correlation while the distance we proposed achieve slightly lower KS (approx. 0.05). The best norm (considering only words in WordNet) is the Cosine with a $KS = 0.5368$. In the case of the GloVe filtrated, best options are Cosine, Correlation and Braycurtis, again

our proposed distance is slightly inferior (approx. 0.05). In the case of GloVe unfiltered, best options are Canberra and Braycurtis, archiving our proposed distance similar results (approx. 0.05). Within this results we can see that our proposed distance is robust across embeddings filtered or not, archiving competitive results in all the cases.

Finally, in figure 6 we can see the values of the KS statistic between synonyms and antonyms distributions for every considered distance in both embeddings. We can see that in the GloVe embedding the value of KS statistics is almost the double than the statistic in the Word2Vec embedding. There are small values compared to the previous experiments (maximum of 0.12 compare to a range between 0.26 and 0.77 in the previous experiments).

Table 1: Synonyms: Results for GloVe (dimension 300). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.2073	$6.0208e^{-170}$	0.4469	0.0
Cosine	0.2623	$1.1153e^{-271}$	0.5368	0.0
Correlation	0.2629	$4.4642e^{-271}$	0.5358	0.0
Canberra	0.4692	0.0	0.4993	0.0
Braycurtis	0.4270	0.0	0.5359	0.0
Proposed	0.4619	0.0	0.4883	0.0

Table 2: Synonyms: Results for Word2Vec. The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.2926	0.0	0.4153	0.0
Cosine	0.7224	0.0	0.6502	0.0
Correlation	0.7234	0.0	0.6509	0.0
Canberra	0.6287	0.0	0.6170	0.0
Braycurtis	0.7073	0.0	0.6575	0.0
Proposed	0.6564	0.0	0.6026	0.0

7.2 The curse of dimensionality in studied norms (GloVe)

In this section we study the performance of every considered norm in the task of distinguish between pairs of synonyms and pairs of non-related words.. The complete results for every distance are included in the appendix section.

In the first table (5) we see the result of the KS test using the distance Euclidean to distinguish between synonyms and non related word pairs in different GloVe embeddings, from dimension 50 to 300. We use both, considering all vocabulary and only WordNet words cases. The next tables correspond respectively to the distances Cosine (6), Correlation (7), Canberra (8), Braycurtis (9) and the proposed distance (10).

In order to see better how this distances behave when we are considering all vocabulary or just the words in WordNet, we include the next figures, one for distance considered:

Table 3: Antonyms: Results for GloVe (dimension 300). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.2692	$2.7072e^{-90}$	0.5286	0.0
Cosine	0.3745	$2.4274e^{-174}$	0.6352	0.0
Correlation	0.3759	$1.2959e^{-175}$	0.6347	0.0
Canberra	0.5801	0.0	0.5954	0.0
Braycurtis	0.5496	0.0	0.6435	0.0
Proposed	0.5794	0.0	0.5998	0.0

Table 4: Antonyms: Results for Word2Vec. The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.3387	$2.3391e^{-155}$	0.4677	$9.8481e^{-296}$
Cosine	0.7939	0.0	0.7035	0.0
Correlation	0.7695	0.0	0.7035	0.0
Canberra	0.7074	0.0	0.6667	0.0
Braycurtis	0.7580	0.0	0.7075	0.0
Proposed	0.7140	0.0	0.6505	0.0

Euclidean (7), Cosine (8), Correlation (9), Canberra (10), Braycurtis (11) and proposed (12).

Notice that the values of the KS statistic are higher in the case that we are just considering just the words in WordNet than if we consider all the words in the embedding. There is just one exception to this behave, the Euclidean norm in dimensions 50 and 100.

In figure 13 we can see a comparative of the performance of every considered norm in both scenarios, considered all vocabulary or just the words in WordNet.

Table 5: Euclidean distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.4193	0.0	0.4430	0.0
100	0.5296	0.0	0.3318	0.0
200	0.4221	0.0	0.3737	0.0
300	0.2073	$6.0208e^{-170}$	0.4469	0.0

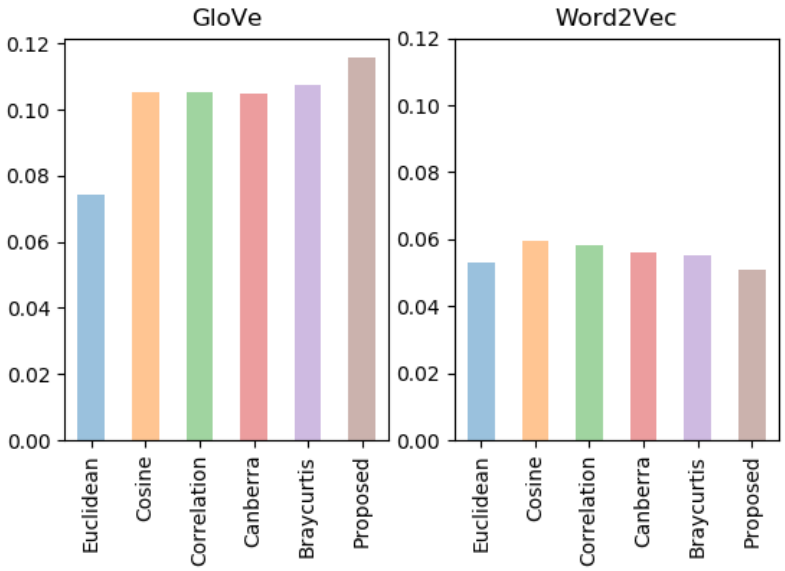


Figure 6: Kolmogorov-Smirnov statistic for the comparison between distances taken from pairs of synonyms and pairs of antonyms in GloVe and Word2Vec scenarios

Table 6: Cosine distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.2630	0.0	0.5160	0.0
100	0.2876	0.0	0.5365	0.0
200	0.3019	0.0	0.5486	0.0
300	0.2629	$1.1153e^{-271}$	0.5368	0.0

8 Conclusions

1. We provided a methodology to evaluate the quality of a word embedding based on comparing distances between known synonyms and pairs of random words.

2. The distributions of distances between synonyms and between antonyms are different but quite similar. It seems not possible to differentiate them in most of the cases based on their embedding distance. We understand that these results are aligned with the intuitive idea that two antonyms have actually a very similar meaning compared to unrelated words.

3. About the norms in the GloVe:

   (a) Vocabulary filtered by WordNet: the most effective distances attend to the norm of the vector that join the two words.

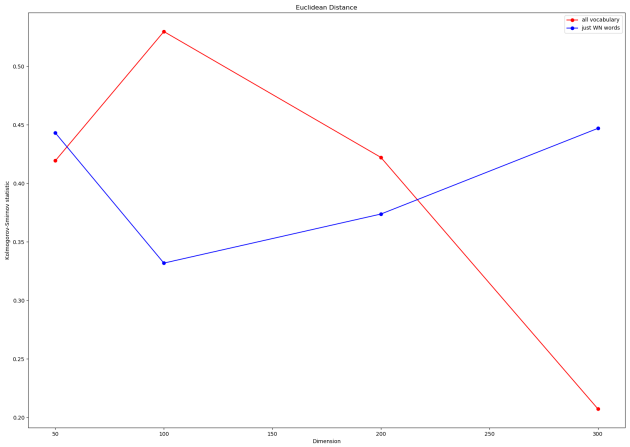


Figure 7: Euclidean distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 7: Correlation distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.2566	0.0	0.5123	0.0
100	0.2841	0.0	0.5349	0.0
200	0.3026	0.0	0.5470	0.0
300	0.2629	$4.4642e^{-271}$	0.5358	0.0

(b) All the vocabulary: the most effective norms attend to difference between components (Canberra, Braycurtis and the proposed one).

(c) The result that norms attending mostly to component differences perform better would support the hypothesis that if two words are synonyms (or antonyms), they are very similar in almost all components.

(d) The words not present in WordNet behave similarly to unrelated words.

About the norms in the Word2Vec context: there is no big difference between the performances in any context. That indicates that the synonyms and antonyms are related in terms of norm and components and this relation is of a nature such that allows us to distinguish them between other words of the whole vocabulary of the embedding.

4. Word2Vec embedding performs significantly better than GloVe embedding representing semantically related words (synonyms or antonyms) closer than unrelated. Being this a desirable characteristic of a word embedding, the present work can establish a new criteria for word embedding selection.

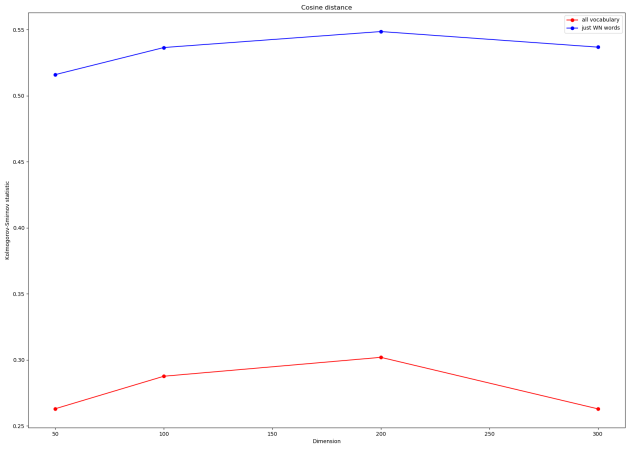


Figure 8: Cosine distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 8: Canberra distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.3989	0.0	0.4476	0.0
100	0.3603	0.0	0.4741	0.0
200	0.4374	0.0	0.5076	0.0
300	0.4692	0.0	0.4993	0.0

9 Future Works

This work showed the potential of a word embedding analysis based on the identification of semantic relations from embedding geometrical properties. Possible lines of future work are:

- Development new norms based on the features of the word embedding, including the use of different weights for each one optimized for having words with similar meaning closer under the proposed distance.
- The creation of a embedding trained with the proposed distance.
- Study more in depth the effect of the Curse of Dimensionality in order to create new distances that works better than the more used ones (L_p, Cosine, Correlation, Canberra, Braycurtis) in the task of distinguish between related (under a specific relation) or non-related pairs of words.
- Study the following hypothesis: if two words belongs to a same synset, each of the features of their vector representations are similar.

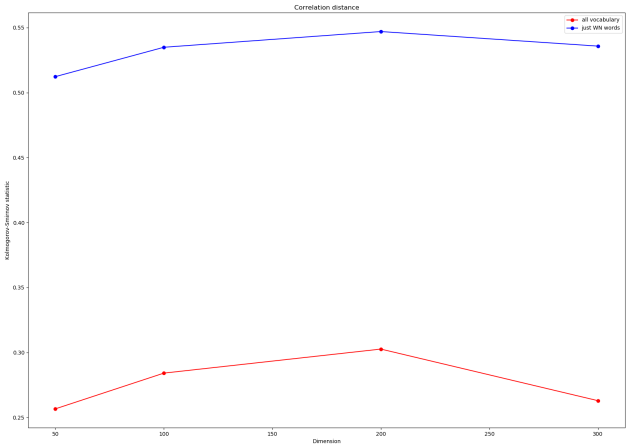


Figure 9: Correlation distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 9: Braycurtis distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.3808	0.0	0.5082	0.0
100	0.3552	0.0	0.5286	0.0
200	0.4216	0.0	0.5471	0.0
300	0.4270	0.0	0.5359	0.0

10 Appendix

10.1 Tables

10.1.1 GloVe dimension 50

In this section we present the complete results of the experiments done with GloVe embeddins of dimension lower than 300. The results for the task of distinguish synonyms from non-related word pairs are in tables: 11 for dimension 50, 13 for dimension 100 and 15 for dimension 200. In the tables 12, 14 and 16 are presented the results for dimensions 50, 100 and 200 respectively of the KS coefficient between the distributions of distances in antonym pairs of words and distances between non-related pairs of words.

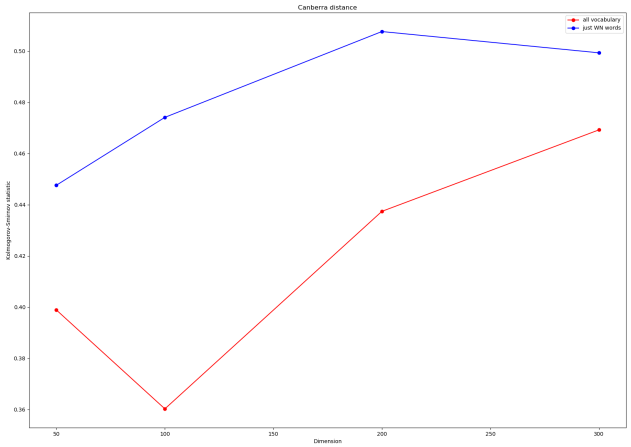


Figure 10: Canberra distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 10: Proposed distance performance (distinguish between synonyms or non related words task) for dimensions 50, 100, 200 and 300. The first column include the results using all the vocabulary in the embedding while the second one use the vocabulary restricted to WN.

Dimension	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
50	0.4037	0.0	0.4254	0.0
100	0.3709	0.0	0.4711	0.0
200	0.4371	0.0	0.4951	0.0
300	0.4619	0.0	0.4883	0.0

Table 11: Synonyms: Results for GloVe (dimension 50). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.4193	0.0	0.4430	0.0
Cosine	0.2630	0.0	0.5167	0.0
Correlation	0.2566	0.0	0.5123	0.0
Canberra	0.3989	0.0	0.4476	0.0
Braycurtis	0.3808	0.0	0.5082	0.0
Proposed	0.4037	0.0	0.4254	0.0

10.1.2 GloVe dimension 100

10.1.3 GloVe dimension 200

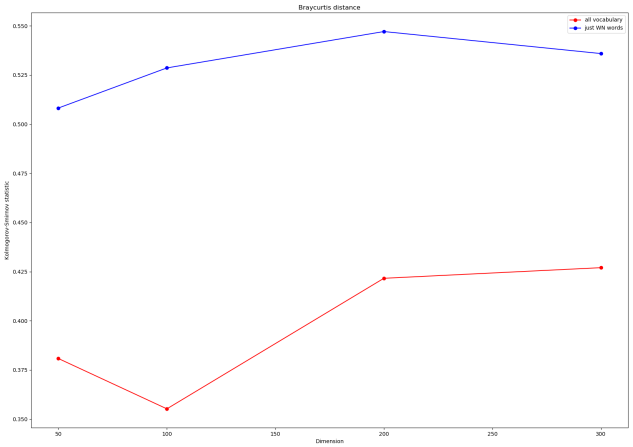


Figure 11: Braycurtis distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 12: Antonyms: Results for GloVe (dimension 50). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.4146	$1.6427e^{-213}$	0.5062	$3.8194e^{-318}$
Cosine	0.2579	$7.1882e^{-83}$	0.6315	0.0
Correlation	0.2487	$4.0653e^{-77}$	0.6303	0.0
Canberra	0.4784	$3.7431e^{-284}$	0.5509	0.0
Braycurtis	0.4368	$5.0910e^{-237}$	0.6277	0.0
Proposed	0.4961	$1.7107e^{-305}$	0.5513	0.0

Table 13: Synonyms: Results for GloVe (dimension 100). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.5296	0.0	0.3318	0.0
Cosine	0.2876	0.0	0.5365	0.0
Correlation	0.2841	0.0	0.5349	0.0
Canberra	0.3603	0.0	0.4741	0.0
Braycurtis	0.3552	0.0	0.5286	0.0
Proposed	0.3709	0.0	0.4711	0.0

10.2 Graphics

10.2.1 GloVe dimension 50

The figure 14 shows the best and the worst distances (based on their KS value) in the case of considering all the words in the embedding, the next one (15) corresponds to the case

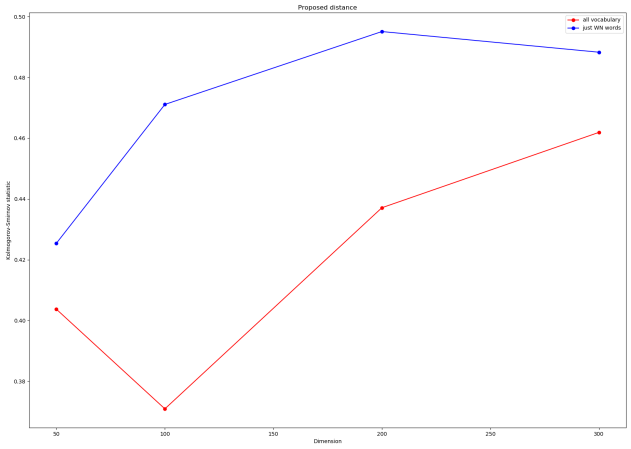


Figure 12: Proposed distance. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 14: Antonyms: Results for GloVe (dimension 100). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.5234	0.0	0.3777	$2.5237e^{-177}$
Cosine	0.2562	$7.9427e^{-82}$	0.6266	0.0
Correlation	0.2545	$9.7082e^{-81}$	0.6256	0.0
Canberra	0.4021	$6.5039e^{-201}$	0.5892	0.0
Braycurtis	0.3702	$2.6892e^{-170}$	0.6303	0.0
Proposed	0.4240	$2.7790e^{-223}$	0.5701	0.0

Table 15: Synonyms: Results for GloVe (dimension 200). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.4221	0.0	0.3737	0.0
Cosine	0.3019	0.0	0.5486	0.0
Correlation	0.3026	0.0	0.5470	0.0
Canberra	0.4374	0.0	0.5076	0.0
Braycurtis	0.4216	0.0	0.5471	0.0
Proposed	0.4371	0.0	0.4951	0.0

of considering just the words in WordNet.

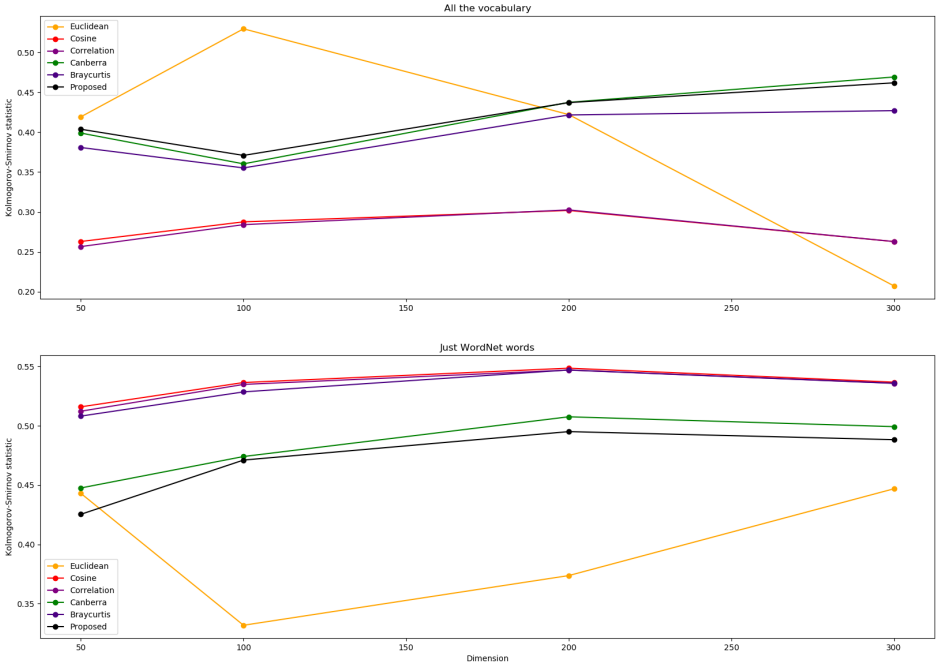


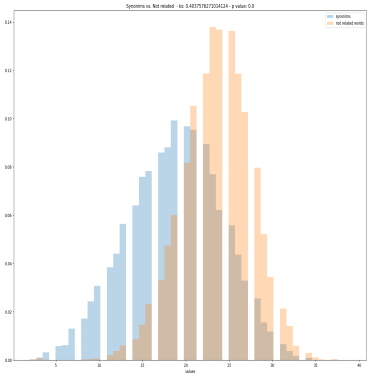
Figure 13: All considered distances. Kolmogorov-Smirnov statistic for dimensions 50, 100, 200, 300 in GloVe, compassion between sets of synonyms and non related words.

Table 16: Antonyms: Results for GloVe (dimension 200). The first two columns include the results using all the vocabulary in the embedding while the later two columns use the vocabulary restricted to WN. For each of them we report the KS statistic and the associated p-value

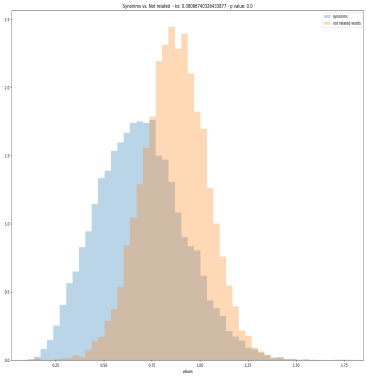
Distance	KS all vocabulary	p-value all vocabulary	KS only WN	p-value only WN
Euclidean	0.4087	$1.4616e^{-207}$	0.4520	$1.1566e^{-253}$
Cosine	0.2719	$4.0749e^{-92}$	0.6493	0.0
Correlation	0.2759	$7.7850e^{-95}$	0.6470	0.0
Canberra	0.5021	$5.7287e^{-313}$	0.6018	0.0
Braycurtis	0.4620	$4.6878e^{-265}$	0.6546	0.0
Proposed	0.5003	$9.9522e^{-311}$	0.5969	0.0

10.2.2 GloVe dimension 100

The figure 16 shows the best and the worst distances (based on their KS value) in the case of considering all the words in the embedding, the next one (17) corresponds to the case of considering just the words in WordNet.

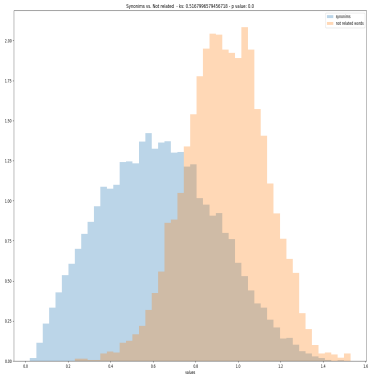


(a) Best distance: Proposed

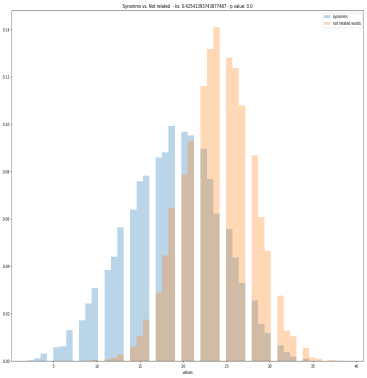


(b) Worst distance: Braycurtis

Figure 14: GloVe dimension 50. Synonyms and non related words. All vocabulary. Best and worst distance distributions.

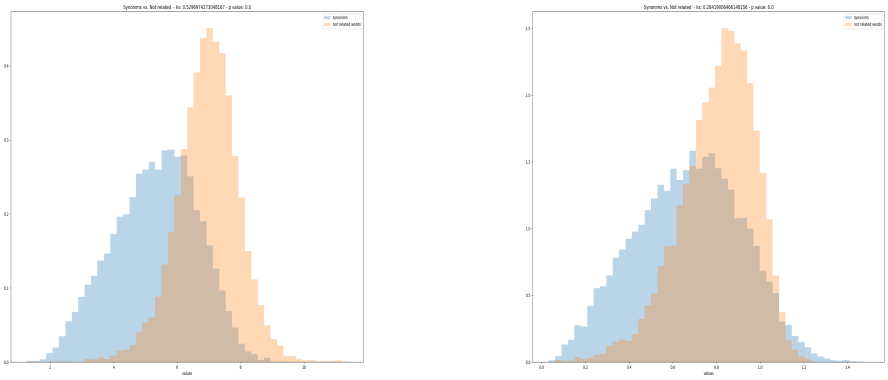


(a) Best distance: Cosine



(b) Worst distance: Proposed

Figure 15: GloVe dimension 50. Synonyms and non related words. Just WordNet. Best and worst distance distributions.



(a) Best distance: Euclidean

(b) Worst distance: Correlation

Figure 16: GloVe dimension 100. Synonyms and non related words. All vocabulary. Best and worst distance distributions.

10.2.3 GloVe dimension 200

The figure 18 shows the best and the worst distances (based on their KS value) in the case of considering all the words in the embedding, the next one (19) corresponds to the case of considering just the words in WordNet.

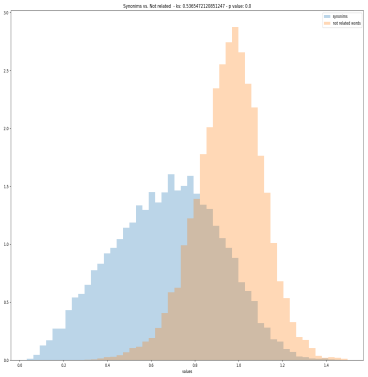
10.2.4 GloVe dimension 300

In the case of distinguish pairs of synonyms in GloVe dimension 300, the figure 20 shows the best and the worst distances (based on their KS value) in the case of considering all the words in the embedding, the next one (21) corresponds to the case of considering just the words in WordNet.

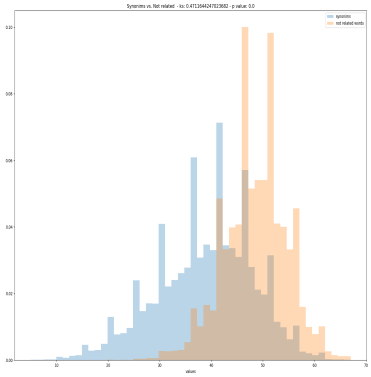
For Word2Vec, the figure 22 shows, in the case of distinguish pairs of synonyms, the best and worst distances (in terms of KS value) if we consider all the vocabulary in the embedding. The figure 23 shows the best and worst distance but considering just words in WordNet.

In the case of distinguish pairs of antonyms in GloVe dimension 300, the figure 24 shows the best and the worst distances (based on their KS value) in the case of considering all the words in the embedding, the next one (25) corresponds to the case of considering just the words in WordNet.

For Word2Vec, the figure 26 shows, in the case of distinguish pairs of antonyms, the best and worst distances (in terms of KS value) if we consider all the vocabulary in the embedding. The figure 27 shows the best and worst distance but considering just words in WordNet.

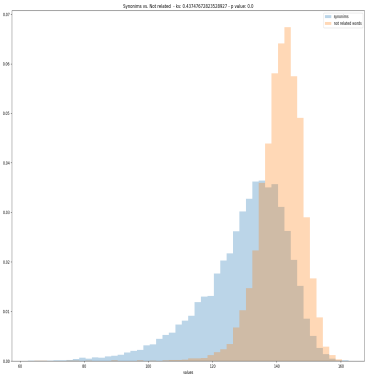


(a) Best distance: Cosine

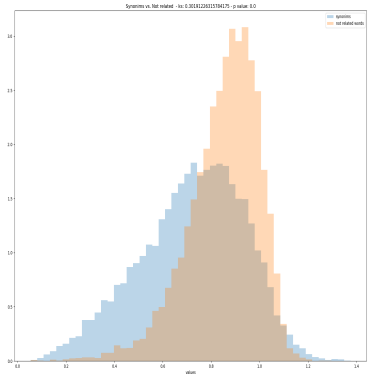


(b) Worst distance: Proposed

Figure 17: GloVe dimension 100. Synonyms and non related words. Just WordNet. Best and worst distance distributions.

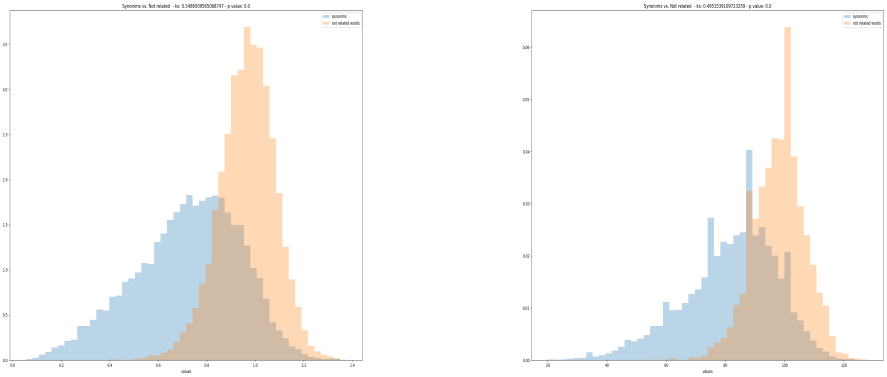


(a) Best distance: Canberra



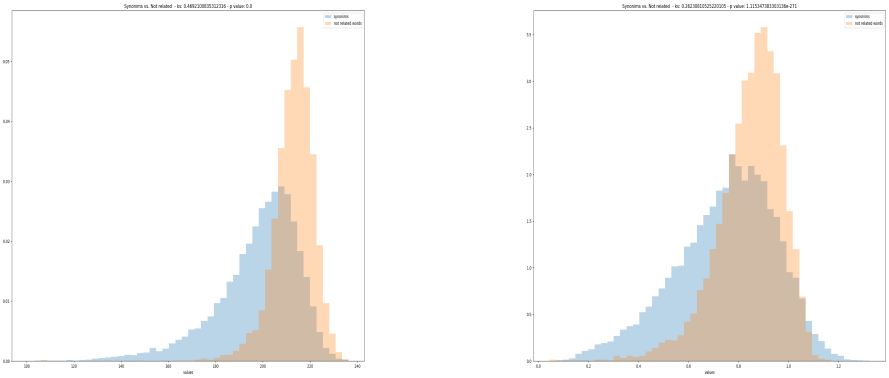
(b) Worst distance: Cosine

Figure 18: GloVe dimension 200. Synonyms and non related words. All vocabulary. Best and worst distance distributions.



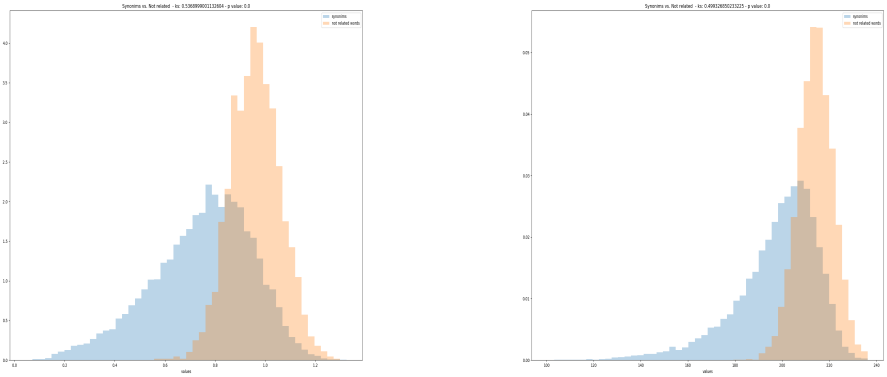
(a) Best distance: Cosine (b) Worst distance: Proposed

Figure 19: GloVe dimension 200. Synonyms and non related words. Just WordNet. Best and worst distance distributions.



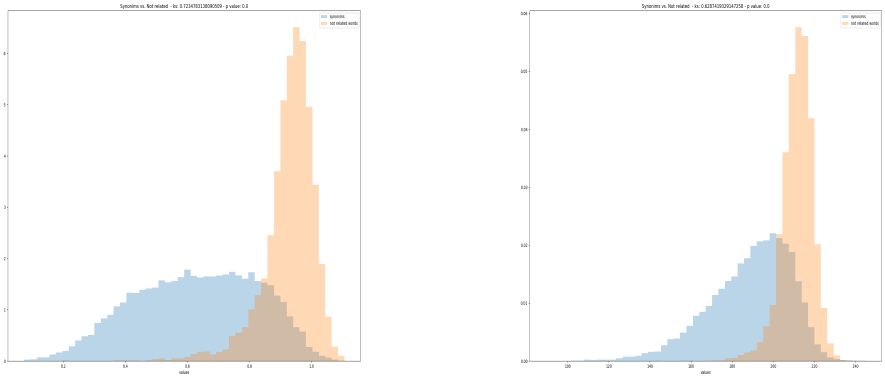
(a) Best distance: Canberra (b) Worst distance: Cosine

Figure 20: GloVe dimension 300. Synonyms and non related words. All Vocabulary. Best and worst distance distributions (excluding Euclidean distance).



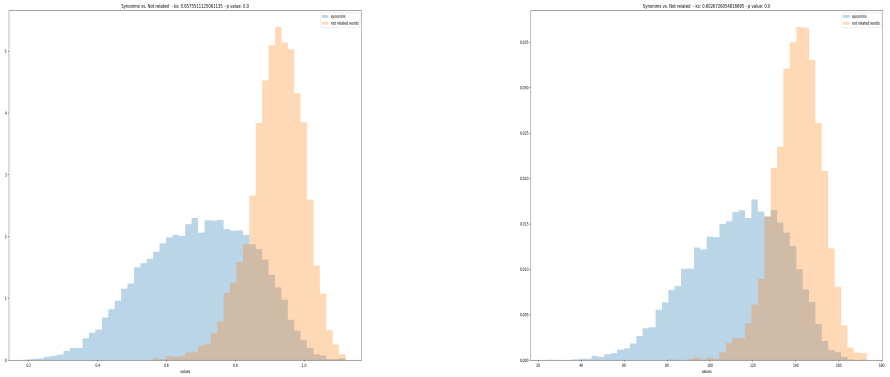
(a) Best distance: Cosine (b) Worst distance: Canberra

Figure 21: GloVe dimension 300. Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).



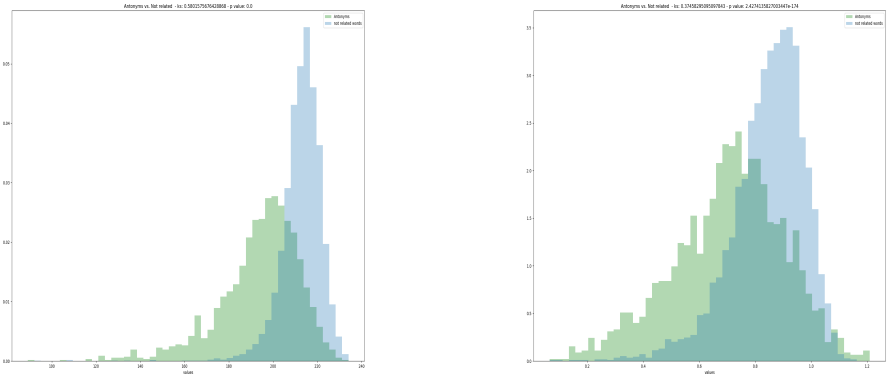
(a) Best distance: Correlation (b) Worst distance: Canberra

Figure 22: Word2Vec (dimension 300). Synonyms and non related words. All Vocabulary. Best and worst distance distributions (excluding Euclidean distance).



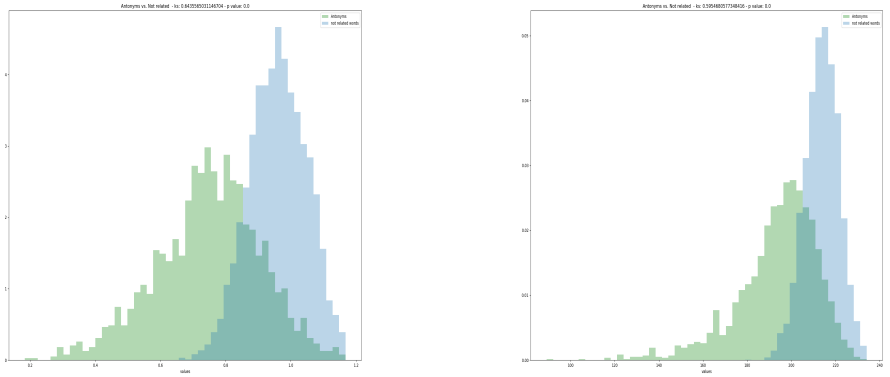
(a) Best distance: Braycurtis (b) Worst distance: Proposed

Figure 23: Word2Vec (dimension 300). Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).



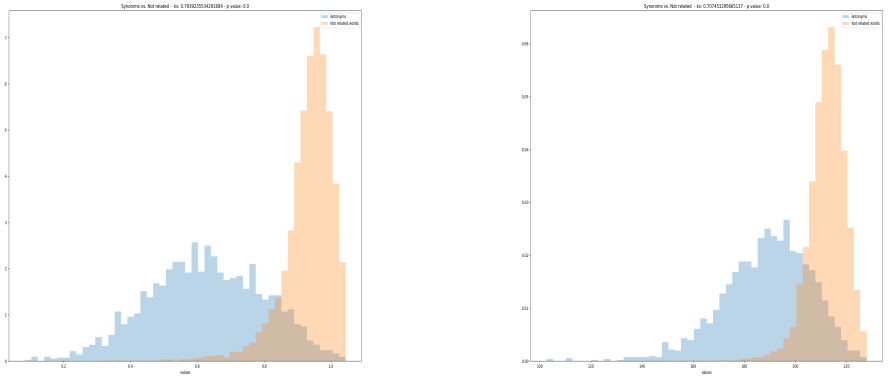
(a) Best distance: Canberra (b) Worst distance: Cosine

Figure 24: GloVe (dimension 300). Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).



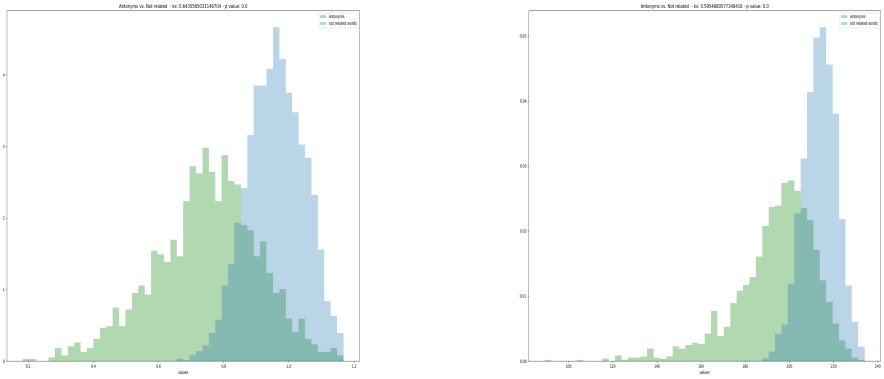
(a) Best distance: Braycurtis (b) Worst distance: Canberra

Figure 25: GloVe (dimension 300). Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).



(a) Best distance: Cosine (b) Worst distance: Canberra

Figure 26: Word2Vec (dimension 300). Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).



(a) Best distance: Braycurtis (b) Worst distance: Canberra

Figure 27: Word2Vec (dimension 300). Synonyms and non related words. Just words in WordNet. Best and worst distance distributions (excluding Euclidean distance).

10.3 WER: Word Embedding Research

10.3.1 Installation

The package is available in `pip` (the standard python package manager). In order to install it just type in the console `$pip install wer`.

Notice that due to the fact that this package has already a lot of dependencies, maybe it will be necessary to install additional packages.

Another way to download the package, is in the website `https://github.com/MGijon/WER`. You can also contribute to the package (for example, finding possible bugs or implementing new functions) and contact me directly through Github.

10.3.2 Documentation

Constructor:

The constructor of the class has the next parameters:

- `path` (*string*): directory where the embedding we want to use is saved. Must include at the end the name of the embedding (the name of the file we want to load).
- `embedings_size` (*integer*): the dimension of the embedding.
- `type` (*string*): it admits two possible values, "GloVe" and "Word2Vec".
- `log` (*string*): name of the logging file.

Each instance of the class that we create has the next variables declared by default (so we can access to any of them anytime). If we want to declare another variable we have to do as a environmental variable:

- `path` (*string*): introduced by the user when an instance of the class is created.
- `embeding_size` (*integer*): introduced by the user when an instance of the class is created.

- `type` (*integer*): the user enters a *string*, but this is saved as a *integer*, 1 for "GloVe" and 2 for "Word2Vec".
- `embeddings_index` (dictionary, *float*): in the case of GloVe embeddings, it keeps the value of the representation of each word in the vocabulary.
- `words` (array, *string*): this is the list of all words presented in the vocabulary of the embedding.
- `filtered_words` (array, *string*): subset of `words`. It contains the words that are present in WordNet.
- `synonims` (array, *string*): it keeps sets of words that are synonyms between them together.
- `synonimsDistribution` (array, *float*): it keeps the value of the distances between synonyms.
- `random_words_pairs` (array, *string*): pairs of words taken by random.
- `randomDistribution`: distances between the pairs of words kept in `randomDistribution`.
- `antonims`:
- `antonimsDistribution` (array, *float*): it keeps the value of the distances between antonyms.
- `wordSynset` (array, *string*): this is prepared to keep arrays of two values, one a word (*string*) and the other one the name of the synset it belongs to (*string*). If a single word belongs to different synsets there will be one array for each one of these synsets.
- `logger`: this is an instance of the `logger` class. It is save as a class variable in order to be able to access from the methods of the class.

Note that any of the variables are protected, this is willfully done.
Main functions:

- `Name`: norm
  - `Description`: computes the indicate distance between the vectors $x = (x_1, \cdots, x_N)$ and $y = (y_1, \cdots, y_N)$.
  - `Input`:
    - `vector` (array, *float*): vector x.
    - `vector2` (array, *float*): vector y.
    - `norm` (*string* or *integer*): number of the distance we want to use or its name (lowercase).
  - `Output`:
  - `Comments`: at this moment, it implements the next norms:
    1. Euclidean

$$d_{\text{euclidean}}(x, y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$$

2. Cosine
$$d_{\text{cosine}}(x, y) = 1 - \frac{x \cdot y}{||x|| \cdot ||y||}$$
,where $||x|| = \sqrt{\sum_{i=1}^{N} x_i}$, *i.e.* the euclidean norm of the vector x.
3. Cityblock
$$d_{\text{cityblock}}(x, y) = \sum_{i=1}^{N} |x_i - y_i|$$
4. L1
5. Chebyshev
$$d_{\text{chebyshev}}(x, y) = \max_i |x_i - y_i|$$
6. Minkowski: it has a third parameter, p, by default equal to two.
$$d_{\text{minkowski}}(x, y, p) = \Big(\sum_{i=1}^{N} |x_i - y_i| \Big)^{\frac{1}{p}}$$
7. Sqeuclidean
$$d_{\text{sqeuclidean}}(x, y) = \Big(d_{\text{euclidean}}(x, y) \Big)^2$$
8. Correlation
$$d_{\text{correlation}}(x, y) = 1 - \frac{(x - \bar{x}) \cdot (y - \bar{y})}{||x - \bar{x}|| \cdot ||y - \bar{y}||}$$
,where $\bar{x}$ is the mean of the components ($\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$).
9. Braycurtis
$$d_{\text{braycurtis}}(x, y) = \sum_{i=1}^{N} \frac{|x_i - y_i|}{|x_i + y_i|}$$
10. Canberra
$$d_{canberra}(x, y) = \sum_{i=1}^{N} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$
11. Kulsinski
12. Maximum 5
$$a = (a_1, \cdots, a_N), \quad a_i = |u_i - v_i|$$
We take the 5 greater coefficients: $a^* = (a_1^*, \cdots, a_5^*)$.
$$d_{\text{maximum 5}} = \sum_{i=1}^{5} a_i^*$$
13. Maximum 10
$$a = (a_1, \cdots, a_N), \quad a_i = |u_i - v_i|$$
We take the 10 greater coefficients: $a^* = (a_1^*, \cdots, a_{10}^*)$.
$$d_{\text{maximum 10}} = \sum_{i=1}^{10} a_i^*$$

14. Maximum 25

$$a = (a_1, \cdots, a_N), \quad a_i = |u_i - v_i|$$

We take the 25 greater coefficients: $a^* = (a_1^*, \cdots, a_{25}^*)$.

$$d_{\text{maximum 25}} = \sum_{i=1}^{35} a_i^*$$

15. Maximum 50

$$a = (a_1, \cdots, a_N), \quad a_i = |u_i - v_i|$$

We take the 50 greater coefficients: $a^* = (a_1^*, \cdots, a_{50}^*)$.

$$d_{\text{maximum 50}} = \sum_{i=1}^{50} a_i^*$$

16. Maximum 100

$$a = (a_1, \cdots, a_N), \quad a_i = |u_i - v_i|$$

We take the 100 greater coefficients: $a^* = (a_1^*, \cdots, a_{100}^*)$.

$$d_{\text{maximum 100}} = \sum_{i=1}^{100} a_i^*$$

17. Distance proposed

$$d_{\text{proposed}}(u, v) = \text{Number of coordinates with different sign}$$

· `Name`: filterWN

  · `Description`: take the words saved in the variable called "words" and copy them in the variable "filtered_words" only if they belongs also to WorNet.

  · `Input`: None.

  · `Output`: None.

· `Name`: randomDistances

  · `Description`: Given an array, takes a number of randomly pairs and return the distances between the elemtens of this pairs.

  · `Input`:

    - `words` (array, *string*): array of words.
    - `number` (*integer*): number of pairs to take.
    - `norm` (*integer*): number corresponding to the norm.

  · `Output`: array of distances

· `Name`: randomDistancesList

  · `Description`: similar to *randomDistances*. The only difference is that recives an array of arrays of words.

    - `words` (array, *string*): array of arrays of words.

- `number` (*integer*): number of pairs to take.
- `norm` (*integer*): number corresponding to the norm we want to use.

· `Output`: array of distances

· `Comments`: the one to use with synonyms and antonyms list of words.

· `Name`: synonymsFilteredWords

· `Description`: Construct the set of synonyms and compute their distances.

· `Input`:

- `norm` (*integer*): number corresponding to the norm we want to use.

· `Output`: None.

· `Comments`: it fills the *synonimsDistribution* and *synonims* arrays (variables of the class).

· `Name`: antonymsFilteredWords

· `Description`: Construct the set of antonyms and compute their distances.

· `Input`:

- `norm` (*integer*): number corresponding to the norm we want to use.

· `Output`: None.

· `Comments`: it fills the *antonimsDistribution* array (variable of the class).

· `Name`: randomFilteredWords

· `Description`: Construct the set of random pairs of words and compute their distances.

· `Input`:

- `norm` (*integer*): number corresponding to the norm we want to use.
- `number` (*integer*): number of randomly chosen pairs of words.

· `Output`: None.

· `Comments`: it fills the *randomDistribution* array (variable of the class).

· `Name`: wordSynsetConstruct

· `Description`: it filters the words in the embedding and then create a list with information about each word the synsets it belongs to.

· `Input`: None.

· `Output`: None.

· `Comments`: it fills the array *wordSynset*, a variable of the class.

Other Functions:

· `Name`: returnVector

· `Description`: given a list of words, it returns their representation in the embedding space.

  - `Input`:
    - `setOfWords` (array, *string*): list of words.
  - `Output`: `vectorsArray` (array, *float*): vectorial representations of the words.

- `Name`: pureSynonyms
  - `Description`: from the vocabulary of the embedding, creates an array of families of synonyms (an array for each word, it includes its synonyms).
  - `Input`: None.
  - `Output`: None.
  - `Comments`: fills the *synonims* variable of the class.

- `Name`: pureAntonyms
  - `Description`: from the vocabulary of the embedding, creates an array of pairs of antonyms.
  - `Input`: None.
  - `Output`: None.
  - `Comments`: fills the *antonyms* variable of the class.

- `Name`: returnSinonyms
  - `Description`: given a word, it returns an array with its synonyms.
  - `Input`:
    - `word` (*string*): words.
  - `Output`: list of synonyms words.

- `Name`: KolmogorovSmirlov
  - `Description`: computes the Kolmogorov-Smirlov statistic.
  - `Input`:
    - `data1` (array, *float*): array of distances corresponding to the first distribution.
    - `data2` (array, *float*): array of distances corresponding to the first distribution.
  - `Output`: the value of the statistic (*float*) and the p-value (*float*).
  - `Comments`: in this first version, it requires the package `stats`.

There are other functions in the package, these are the main ones. Full documentation available in the package.

References

[1] O. Levy, Y. Goldberg. "Linguistic Regularities in Sparse and Explicit Word Representations" (2014).

[2] G.A. Miller. "WordNet: A Lexical Database for English" (1995).

[3] J. Camacho-Collados, M.T. Pilehvar. "From Word to Sense Embeddings: A Survey on Vector Representations of Meaning" (2018).

[4] P. Jeffrey, R. Socher, and C. Manning. "Glove: Global vectors for word representation". Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). (2014).

[5] Y. Goldbert and O. Levy. "Word2Vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". (2014)

[6] X. Rong. "Word2Vec Parameter Learning Explained" (2016).

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. "Distributed Representations of Words and Phrases and their Compositionality" (2013).

[8] T. Mikolov, K. Chen, G. Corrado, J. Dean. "Efficient Estimation of Word Representations in Vector Space" (2013).

[9] T. Mikolov, W. Yih, G. Zweig. "Linguistic Regularitues in Continuous Space Word Representations" (2013).

[10] T. Bolukbasi, K.W. Chang, J. Zou, V. Saligrana, A. Kalai. "Man is to Computer Programmer as Woman is to Homemaer? Debiasing Word Embeddings" (2016).

[11] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. "When Is "Nearest Neighbour" Meaningful?" (1998).

[12] P. Indyk, R. Motwani. "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality" (1999).

[13] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. "When Is "Nearest Neighbor" Meaningful?" (1999).

[14] C.C. Aggarwal1, A. Hinneburg, D.A. Keim. "On the Surprising Behavior of Distance Metrics in High Dimensional Space" (2001).

[15] P. Indyk, R. Motwani. "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality" (1999).

[16] M. Verleysen, D. François. "The Curse of Dimensionality in Data Mining and Time Series Prediction" (2005).