Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 160–167

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019

Word2Vec Model Analysis for Semantic Similarities in English Words

Derry Jatnika[a,*], Moch Arif Bijaksana[a], Arie Ardiyanti Suryani[a]

[a] *School of Computing, Telkom University, Bandung, Indonesia, 40257*

Abstract

This paper examines the calculation of the similarity between words in English using word representation techniques. Word2Vec is a model used in this paper to represent words into vector form. The model in this study was formed using the 320,000 articles in the English Wikipedia as the corpus and then Cosine Similarity calculation method is used to determine the similarity value. This model then tested by the test set gold standard WordSim-353 as many as 353 pairs of words and SimLex-999 as many as 999 pairs of words, which have been labelled with similarity values according to human judgment. Pearson Correlation was used to find out the accuracy of the correlation. The results of the correlation from this study are 0.665 for WordSim-353 and 0.284 for SimLex-999 using the Windows size 9 and 300 vector dimension configurations.

Keywords: Word2Vec; Cosine Similarity; Pearson Correlation;

1. Introduction

Semantic similarity has an important role in the field of linguistics, especially those related to the similarity of words meaning. Semantic similarity between words is the search for similarities between two words or more. In terms of the similarity of words meaning, two words may differ syntactically but have the same meaning. For example, *Me* and *I* have the same meaning. Calculating the similarity of words meaning has been widely represented in the field of linguistics with basic rules as a result of the reasoning of human thought. This calculation can also be done through the field of computer science, namely the study of Natural Language Processing and Text Mining based on the field of linguistics.

Natural Language Processing is one of the fields of science of Artificial Intelligence that deals with the interaction between computers and natural human language[1]. Computers need to process the human language that is received

[*] Corresponding author. Tel.: +6285726567333
E-mail address: derry.jatnika@gmail.com

10.1016/j.procs.2019.08.153

first so that the intentions of humans can be understood and then provide the appropriate response. For example, a computer must know the value of how many similarities between *Me* and *I*.

The main technique for calculating the similarity and relevance of words meaning using the Word2Vec model from word embeddings. Word2Vec is a model used to represent words into vectors. Then, the similarity value can be generated using the Cosine Similarity formula of the word vector values produced by the Word2Vec model. In the construction of the Word2Vec model called the training process, there are several features that used to produce the Word2Vec model including windows size and vector dimensions configuration. Some previous study mostly used Windows size and vector dimensions configuration to produce the Word2Vec model. In this study several Windows size and vector dimensions configuration were used to compare the similarity values of each configuration of the resulting Word2Vec model. The configuration of the Word2Vec model that produces the best similarity values will be the result of this study. It is important to know the best configuration of the Word2Vec model to find the best value for similarity word meanings.

2. Related Work

In a previous study related to the Word2Vec model, trying to apply the Word2Vec model by doing some architectural configurations of Word2Vec CBOW and Skip-Gram. The research used 120,000 articles of English Wikipedia training data, the configuration of Word2Vec models with windows size 5 & vector dimension 300, and minimum preprocessing procedures: XML tag deletion in the corpus of the English Wikipedia. The study made 4 configuration models of the Word2Vec model, namely: Full English CBOW Wikipedia (FW-CBOW), Full English Wikipedia Skip-Gram (FW-SG), Simple English CBOW Wikipedia (SW-CBOW), and Simple English Wikipedia Skip-Gram (SWSG). The pre-trained Google News Skip-Gram model with windows size 5 & vector dimension 300 (GN-SG) was used as comparative material for the 4 models it made[2].

The evaluation of previous study used recall rate points to calculate system value evaluations with the gold standard WordSim-353 test set. Results of the study stated that the FW-CBOW model produced the best recall rate points with a cumulative score of 7.03, compared to other models. This result is even better than the Word2Vec model made by Google[3].

Therefore, this study will use FW-CBOW as the main benchmark, but there are several configurations of the modified Word2Vec model such as the 320,000 articles in the English Wikipedia corpus, the configuration of the Word2Vec model, namely: windows size 3, 6, 9 and vector dimension 50, 150, 300. This study combines these configurations to produce 9 Word2Vec models that will be compared as evaluation material using the Pearson Correlation with the test sets WordSim-353 and SimLex-999 to used as comparative material.

3. Methodology

3.1. Semantic Similarity

Semantic similarity is a concept that can measure the similarity of meaning in the context of short texts. Text that is compared can be in the form of words, short sentences, and a document[4]. Semantic similarity has an important role in several tasks from Natural Language Processing and several related fields such as text classification, document clustering, text summarization, etc[5]. Semantic similarity is metrics defined above documents or words, where ideas have located the distance between the two is based on the similarity of meaning or semantic content compared to predictable similarities regarding representation their syntax. Semantic similarity is also a mathematical tool used to estimate the strength of the semantic relationship between language units, concepts or examples, through numerical descriptions obtained according to the comparison of information that supports its meaning or describes its nature. For example, knowing the similarity between a *bicycle* and a *motorcycle* or the difference between a *car* and a *horse*. The example of semantic similarity can be seen in Table 1.

Table 1: Examples of the word pair relationships by Mikolov

Relationship	Example 1	Example 2
France - Paris	*Italy : Rome*	*Apple : Iphone*
Big - Bigger	*Small : Larger*	*Kona : Hawaii*
Miami - Florida	*Baltimore : Maryland*	*USA : Pizza*
Einstein - Scientist	*Messi : Midfielder*	*Obama : Barack*
Sarkozy - France	*Google : Android*	*Quick : Quicker*

3.2. System Overview

This study builds a system that can calculate the similarity values between words by using the Word2Vec model representation. This study was carried out by building several configurations of the Word2Vec model to find the best similarity value. The configuration is done by changing the windows size settings and the word vector dimensions. This study step can be shown in Figure 1.

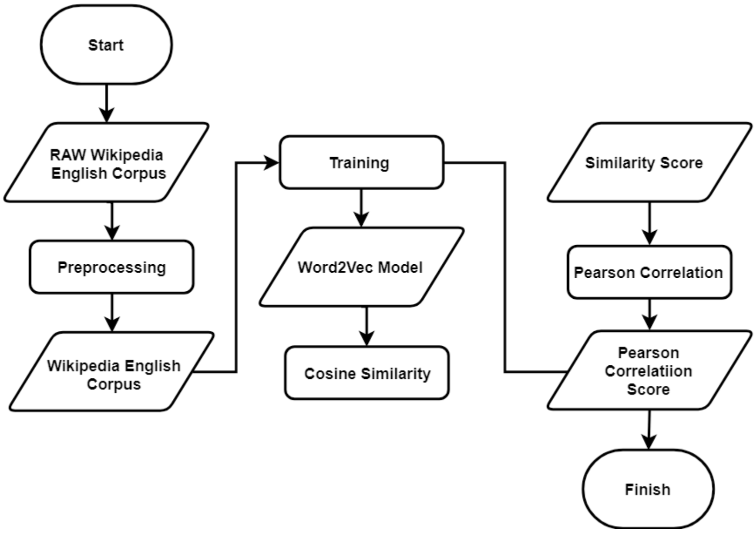

Fig. 1: study flow system.

3.3. Word Embeddings

Word Embeddings is the collective name for a set of language modelling and features of learning techniques in Natural Language Processing where words or phrases are represented in the form of real number vectors. Conceptually, Word Embeddings involves mathematical formulas. The models used in Word Embeddings are varied, one of which is the Word2Vec model. Word2Vec represents words into vector based on several features they have such as windows size and vector dimensions.

Similar words tend to have the same vector values and are grouped in the same block can be seen in Figure 2. Therefore, Word2Vec can capture the similarity value between words from the training of a large corpus. The resulting similarity value is obtained from the word vector value than calculated using the Cosine Similarity equation. The similarity value produced by Word2Vec ranges from -1 to 1 as the highest similarity value.

Word2Vec can provide an efficient implementation of architectural Continuous Bag of Words (CBOW) and Skip-Gram to calculate vector representations of words, these representations can be used for various tasks in language processing. CBOW architecture predicts current words based on context, while Skip-Gram architecture predicts words around the word currently given. The CBOW and Skip-Gram architectures can be seen in Figure 3.

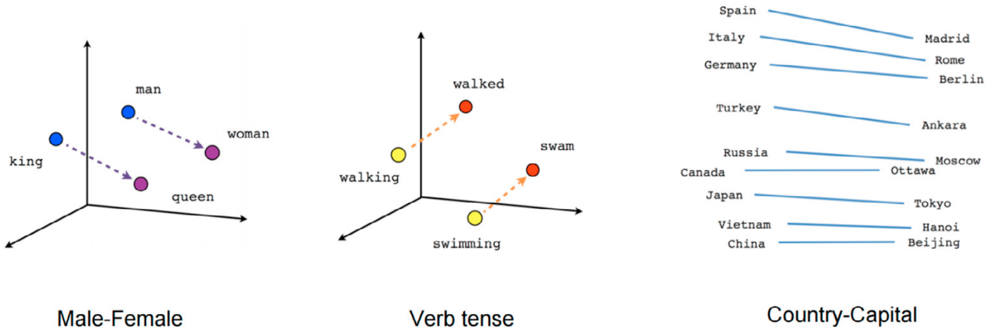

Fig. 2: Word2Vec representation[3].

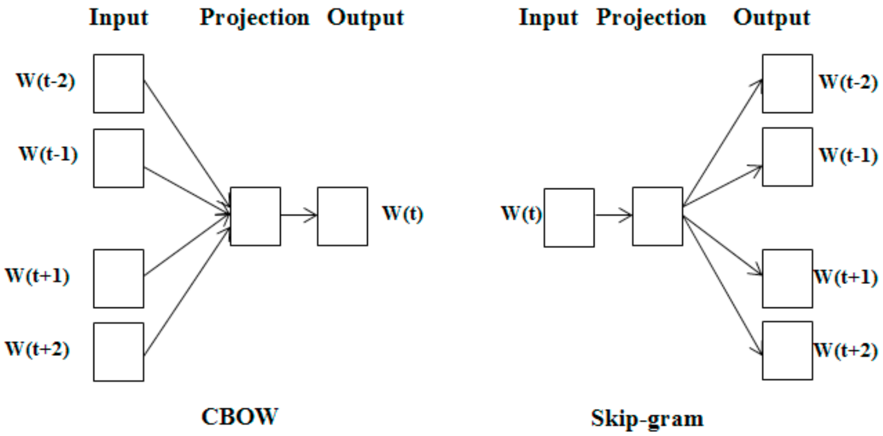

Fig. 3: Word2Vec CBOW and Skip-Gram models architecture[3].

3.4. Dataset

The dataset used for the Word2Vec model training in this study is the 320,000 English Language Wikipedia articles found in Wikipedia Database[1] in XML format. Then do simple preprocessing by deleting XML tags to produce a clean corpus in the form of a large collection of words from 320,000 articles. The test data used as testing datasets are WordSim-353[2] and SimLex-999[3]. This test set contains the similarity value of word pairs from human ratings or called the gold standard.

The WordSim-353 dataset is a dataset of 353 pairs of nouns, while each pair is presented without context and is associated with 13 to 16 human judgments about similarities and reciprocal relationships on a scale from 0 to 10[6]. The SimLex-999 dataset contains various pairs of concrete and abstract adjectives, nouns, and verbs, together with ratings independent of concrete and association strengths (free) for each pair[7].

[1] https://dumps.wikimedia.org/enwiki/
[2] http://leviants.com/ira.leviant/MultilingualVSMdata.html
[3] http://leviants.com/ira.leviant/MultilingualVSMdata.html

3.5. Preprocessing

Preprocessing is the process of converting unstructured data into structured data as needed, for further text mining processes (sentiment analysis, summarizes, document groupings, etc). Preprocessing performed on the corpus dataset is tokenizing and case folding. Tokenizing is the process of dividing text in the form of sentences, paragraphs or documents, into certain tokens. For example, the tokenizing of a sentence *they eat and laugh really hard* produce six tokens, namely: *they*, *eat*, *and*, *laugh*, *really*, *hard*. Case folding is the process of matching a case in a document. This is done to facilitate search. Not all text documents are consistent in the use of capital letters. Therefore, the case folding role is needed to convert all text in a document into a standard form (usually lowercase). For example, a case folding from *COMPUTER*, produce *computer*.

3.6. Word2Vec Configuration Setup

This Word2Vec model is built on the results of training from a large corpus of English Wikipedia. The Gensim library of the Python programming language has an important role in this study because the Gensim library provides all the features to produce a Word2Vec model. The making of the Word2Vec model was built based on the configuration of windows size and different vector dimensions. In this study, the window size 3, 6, 9 and vector dimensions 50, 150, 300.

The Word2Vec model can be calculated using the value of word vectors obtained using the Cosine Similarity equation. Cosine Similarity is the calculation of the similarity between two n-dimensional vectors by looking for a cosine value from the angle between the two and is often used to compare documents in text mining[8]. The Cosine Similarity formula is as follows:

$$similarity = \cos\theta = \frac{\overline{x} \bullet \overline{y}}{\| \overline{x} \| \| \overline{y} \|} \tag{1}$$

where :

$\overline{x} \bullet \overline{y}$: Vector dot product from x and y. $\sum_{k=1}^{n} x_k y_k$
$||x||$: Long vector x. $\sum_{k=1}^{n} x_k^2$
$||y||$: Long vector y, $\sum_{k=1}^{n} y_k^2$

The results of Equation (1) are then recalculated using the Pearson Correlation Equation to find out how much accuracy the value generated by the system with the gold standard WordSim-353 and SimLex-999. Pearson Correlation is used to evaluate the results of similarity calculations. Pearson Correlation produces a correlation value between the range 0 to 1[9]. The formula of the Pearson Correlation can be seen in Equation (2) and criteria for correlation values can be seen in Table 2.

$$corr = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \tag{2}$$

where:
n : Number of pair words,
x : Value of the system,
y : value of the gold standard.

Table 2: Criteria for correlation

r	Correlation Criteria
0	No correlation
0-0.5	Weak correlation
0.5-0.8	Moderate correlation
0.8-1	Strong correlation
1	Perfect correlation

4. Evaluation

4.1. Testing

Tests on this study were carried out by combining windows size 3, 6, 9 and vector dimensions 50, 150, 300 so that the tests were carried out 9 times with different configurations. Configurations are used which aim to find the best configuration of the Word2Vec model for the Wikipedia corpus in English. Configuration details can be seen in Table 3.

Table 3: Word2Vec study setup

study	Word2Vec Configuration	
	Window Size	Vector Dimension
1	3	50
2	3	150
3	3	300
4	6	50
5	6	150
6	6	300
7	9	50
8	9	150
9	9	300

4.2. Analysis of the Test Results

The test results of this study are the Pearson Correlation score of the similarity value generated by the system which is correlated with the gold standard values of WordSim-353 and SimLex-999. Pearson Correlation results from WordSim-353 and SimLex-999 can be seen in Table 4 & Table 5. Some parts of the results from the similarity score produced can be seen in Table 6 & Table 7.

Table 4: correlation result for test set WordSim-353

Windows Size	WordSim-353		
	Vector Dimension		
	50	150	300
3	0.6005	0.6262	0.6225
6	0.6336	0.6463	0.6484
9	0.6478	0.6628	**0.6653**

In Table 4, it can be seen that the higher the windows size and vector dimension configuration applied to the Word2Vec model, the higher Pearson Correlation value will be. Therefore for the WordSim-353 test set, the best configuration for this study is Windows size 9 and vector dimension 300.

Table 5: correlation result for test set SimLex-999

Windows Size	**SimLex-999**		
	Vector Dimension		
	50	**150**	**300**
3	0.2369	0.2723	0.2633
6	0.2281	0.2560	0.2651
9	0.2279	0.2540	**0.2845**

Similar to the test set of WordSim-353 in Table 4, in the SimLex-999 test set the best configuration for the Word2Vec model is windows size 9 and vector dimension 300. Therefore, it can be concluded that the best configuration of Word2Vec model in this study is windows size 9 and vector dimension 300.

This study also shows that the higher window size and vector dimensions can produce a high correlation value and the similarity value will also be better. But window sizes and vector dimensions that are too large also do not always produce good values because larger window size value, the more context of words will cause the value of the similarity to be weak.

Table 6: part of similarity test set WordSim-353

Word1-Word2	**Gold Standard WordSim-353**	**WS**	**Similarity**		
			Vector Dimension		
			50	**150**	**300**
coast-shore	9.10	3	0.8010	0.6577	0.6128
		6	0.7900	0.6651	0.6374
		9	0.7954	0.6787	0.6140
book-paper	7.46	3	0.5667	0.4807	0.4104
		6	0.5102	0.4520	0.3974
		9	0.4899	0.3955	0.3593
train-car	6.31	3	0.6665	0.5611	0.4596
		6	0.6647	0.5431	0.4767
		9	0.6375	0.3955	0.4634
⋮	⋮	⋮	⋮	⋮	⋮

5. Conclusion and Future Work

Based on the results of tests and analyses that have been carried out, conclusions can be drawn as follows:

1. Based on the similarity score generated by the Word2Vec model created, the Pearson Correlation value for WordSim-353 is 0.665 which means that it is moderately correlated, while for SimLex-999 it is 0.284 which means weak correlation. This study produces a poor value for the SimLex-999 dataset compared to WordSim353. This is because the Wordsim353 dataset is a pair of related or association, therefore different pairs of words (different materials, functions etc.) get high similarity values, for example that *clothes* are not similar to *closets* but they are very much related. While the Simlex-999 dataset is a dataset to capture similarity, rather than relatedness or association, so as to obtain a low similarity value.

Table 7: part of similaruty test set SimLex-999

Word1-Word2	**Gold Standard SimLex-999**	**Similarity**			
		WS	**Vector Dimension**		
			50	**150**	**300**
fast-rafid	9.85	3	0.5847	0.5092	0.3910
		6	0.5481	0.4515	0.4168
		9	0.5038	0.4753	0.4134
happy-glad	9.39	3	0.7375	0.6629	0.6371
		6	0.7286	0.6675	0.6188
		9	0.7339	0.6779	0.5958
illegal-immoral	4.70	3	0.5774	0.5379	0.4784
		6	0.5890	0.5066	0.4754
		9	0.6557	0.5265	0.4746
⋮	⋮	⋮	⋮	⋮	⋮

2. Factor affecting the Word2Vec model's similarity value is the number of occurrences of words in the corpus based on the window size and vector dimensions used. If the window size and dimensions vector size used is too small, the context of the resulting word is also less. As for the bigger window size and vector dimension used, the more context the word is produced, the greater the likelihood that the pair will appear.

As a future work for the next study, using a large corpus takes a long time for the process of training the Word2Vec model. Therefore, it is recommended to use parallel programming to overcome this.

References

1. Handler, A.. An empirical study of semantic similarity in WordNet and Word2Vec. In: *University of New Orleans Theses and Dissertations.* (2014), .
2. Jin, L., Schuler, W.. A Comparison of Word Similarity Performance Using Explanatory and Non-explanatory Texts. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, p. 990–994.
3. Mikolov, T., Le, Q.V., Sutskever, I.. Exploiting Similarities among Languages for Machine Translation. *arXiv preprint arXiv:13094168* 2013;.
4. Agirre, E., Diab, M., Cer, D., Gonzalez-Agirre, A.. Semeval-2012 Task 6: A Pilot on Semantic Textual Similarity. In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics; 2012, p. 385–393.
5. Mihalcea, R., Corley, C., Strapparava, C., et al. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In: *AAAI*; vol. 6. 2006, p. 775–780.
6. Kliegr, T., Zamazal, O.. Antonyms are Similar: Towards Paradigmatic Association Approach to Rating Similarity in Simlex-999 and WordSim-353. *Data & Knowledge Engineering* 2018;**115**:174–193.
7. Hill, F., Reichart, R., Korhonen, A.. Simlex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics* 2015;**41**(4):665–695.
8. Lai, S., Liu, K., He, S., Zhao, J.. How to Generate a Good Word Embedding. *IEEE Intelligent Systems* 2016;**31**(6):5–14.
9. Rong, X.. Word2Vec Parameter Learning Explained. *arXiv preprint arXiv:14112738* 2014;.