A COMPARISON OF SEMANTIC SIMILARITY APPROACHES IN DOCUMENT RANKING AND INFORMATION RETRIEVAL

Abstract

Semantic similarity measures compute the similarity between concepts/terms included in knowledge sources in order to perform estimations [1]. To improve the retrieval of documents using similarity measures, a number of techniques have been proposed but most of them do not take into consideration the semantic information of the text while calculating the similarity between the query and the documents to be retrieved. The inclusion of semantic information in any similarity measure improves the efficiency of the similarity measure and thus helps in retrieving more relevant documents [2]. The idea behind this research is to represent the documents as vectors of features, and compare documents by measuring the distance between these features. This project explores how several similarity measures work with word embedding in finding the relevant documents in a large dataset.

Table of Contents

1. Introduction

Over the years, it has been established that Natural Language Processing (NLP) is a theory-motivated range of computational techniques for the automatic analysis and representation of human language. NLP research has evolved from the era of punch cards and batch processing (in which the analysis of a sentence could take up to 7 minutes) to the era of Google and the likes of it (in which millions of webpages can be processed in less than a second) [4]. The goal is for computers to process or "understand" natural language in order to perform various human-like tasks like language translation or answering questions. With the rise of voice interfaces and Chabot's, NLP is one of the most important technologies of the 4th Industrial Revolution and become a popular area of AI.

Information Retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information. The system assists users in finding the information they require but it does not explicitly return the answers of the questions. It informs the existence and location of documents that might consist of the required information. The documents that satisfy user's requirement are called relevant documents and a perfect IR system will retrieve only relevant documents. Document similarity (or distance between documents) is a one of the central themes in Information Retrieval. Document similarity means that user's query text will be matched with the document text and on the basis on this matching user retrieves the most relevant documents [5].

With the enormous increase in recent years in the number of text databases available online, and the significant need for better techniques to access and retrieve this information, there has been a strong resurgence of interest in the research done in the area of Information Retrieval (IR). Everyone wants to access the most relevant data to their specific query within minutes, which creates the need to find techniques that ensure the documents are ranked and retrieved as per the user's need.

2. Literature Review

After reviewing research papers based on Information Retrieval and Document Similarity, we came across several efficient approaches that have already been proposed. However, it was certain that Word2Vec, a group of related models that are used to produce word embedding that was created by Google, has been largely unexplored in the field. It takes as its input a large corpus of words and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. Word2Vec is a particularly computationally-efficient predictive model for learning word embedding's from raw text.

Text Similarity is also one of the essential techniques of NLP which is used to find the closeness between two chunks of text by its meaning or by surface. As we know, computers require data to be converted into a numeric format to perform any machine learning task. In order to perform these tasks efficiently, various word embedding techniques are used to encode the text data. This allows us to perform NLP operations such as searching similar documents from the database, recommending semantically similar news articles, etc. Hence, in this research, Word2Vec is used to encode our data and then similarity measures such as Cosine Similarity, Word Mover's Distance and Jaccard Similarity Index are used to calculate the similarity between the query and all the documents in the dataset.

3. Methodology and Framework

3.1 System Architecture

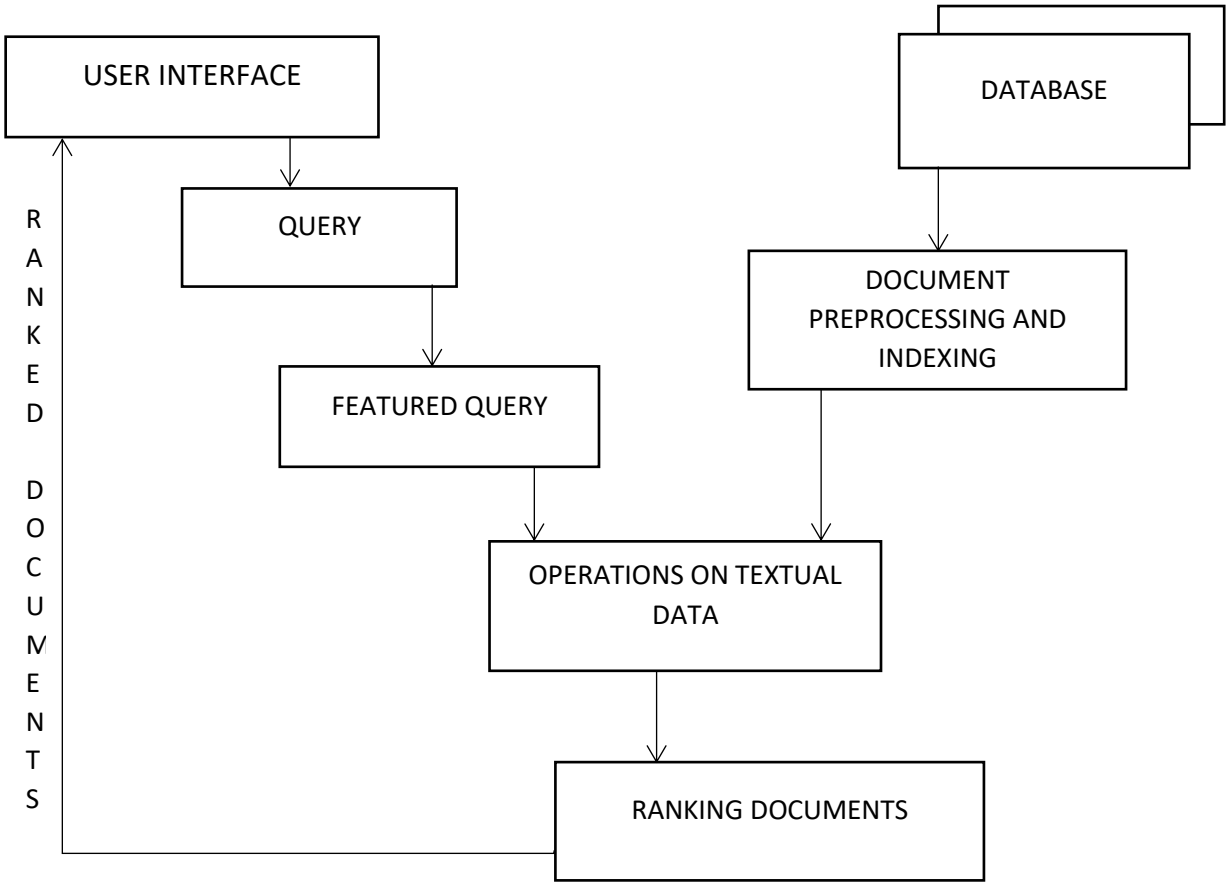


Figure 1- A Classic Information Retrieval System

A classic Information Retrieval system takes a query from the user and retrieves the most relevant documents for that query from the database. However, to do so, the system first performs operations on the textual data to convert them in a form that is easily understandable for the computer. After performing these operations, the documents from the dataset are ranked according to their relevance to the user's query and these ranked documents are then returned to the user.

3.2 Dataset Used

The dataset being used for the research project is the Classic4 Dataset. It was obtained from Cornell University's Archives. This dataset consists of 4 different document collections: CACM, CISI, CRAN, and MED. The collection being used for this research is the CISI Document Collection, to determine the most relevant documents for the given query by using different word embedding techniques and similarity measures. This document collection comprises of the abstracts of over 1000+ scientific research papers.

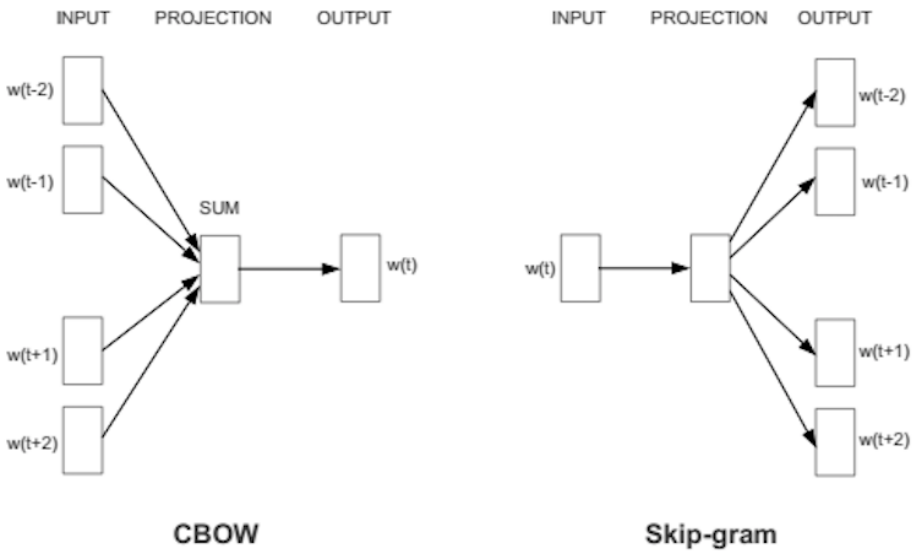
3.3 Tools Used

3.3.1 Word2Vec

Word2Vec is a group of related models that are used to produce word embedding's. It is a two-layer neural net that processes text by vectorizing words. Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. Word2Vec is a particularly computationally-efficient predictive model for learning word embedding's from raw text. Word2vec is a combination of two techniques – CBOW (Continuous bag of words) and Skip-gram model. Both of these are shallow neural networks which map word(s) to the target variable which is also a word(s).

3.3.1.1 Continuous Bag-of-Words (CBOW) - CBOW predicts target words from the surrounding context words.

3.3.1.2 Skip-Gram - It predicts surrounding context words from the target words (inverse of CBOW). It treats each context-target pair as a new observation, and this tends to do better when we have larger datasets.



3.3.2 Cosine Similarity:

It measures the cosine of the angle between two vectors projected in a multi-dimensional space. In this context, the two vectors are the query and the documents. When plotted on a multi-dimensional space, where each dimension corresponds to a word in the document, the cosine similarity captures the orientation (the angle) of the documents and not the magnitude.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

The closer the documents are by angle, the higher is the Cosine Similarity. Cosine similarity returns the score between 0 and 1 where 1 means the texts are exactly similar and 0 means the texts are nowhere similar to each other.

3.3.3 Jaccard Similarity:

The Jaccard Index, also known as Intersection over Union, is a statistic used for gauging the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between

finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

The Jaccard Similarity Index compares members for two sets to see which members are shared and which are distinct, thus it does not take into consideration any sort of data duplication. It is best used to determine the overlap between any two pieces of text; however it cannot determine the overlap in meaning between these two texts. Jaccard Similarity also does not perform well when the size of the documents are too large as even with similar words, the union of the two documents would reduce the similarity index by a significant margin.

3.3.4 Word Mover's Distance:

Word Mover's Distance (WMD) uses the word embedding's of the words in the two texts to determine the minimum distance that the words in one text need to travel in the semantic space to reach the words in the other text. If this distance is small, it means the two texts are similar to each other. This metric makes use of the word embedding's power to overcome the basic distance measurement limitations between among the words. It is able to calculate the similarity even when there are no common words, provided that similar words have similar vectors in the vector space.

3.3.5 Precision and Recall:

Precision and recall are the measures used in the information retrieval domain to measure how well an information retrieval system retrieves the relevant documents requested by a user. The measures are defined as follows:

3.3.5.1 Precision = Total number of documents retrieved that are relevant/Total number of documents that are retrieved.

3.3.5.2 Recall = Total number of documents retrieved that are relevant/Total number of relevant documents in the database.

4. Design Methodology

In this Information Retrieval system, the documents and search queries are first processed to remove punctuations, convert the text to lowercase, to lemmatize the words and convert them into tokens. The Word2Vec model is then fit into the corpus of documents so as to create word embedding's. Different similarity measures are then employed to compute the similarity between documents and the query. This system then retrieves the 100 most relevant documents based on the calculated similarity. Based on the retrieved documents, precision and recall for the model is calculated.

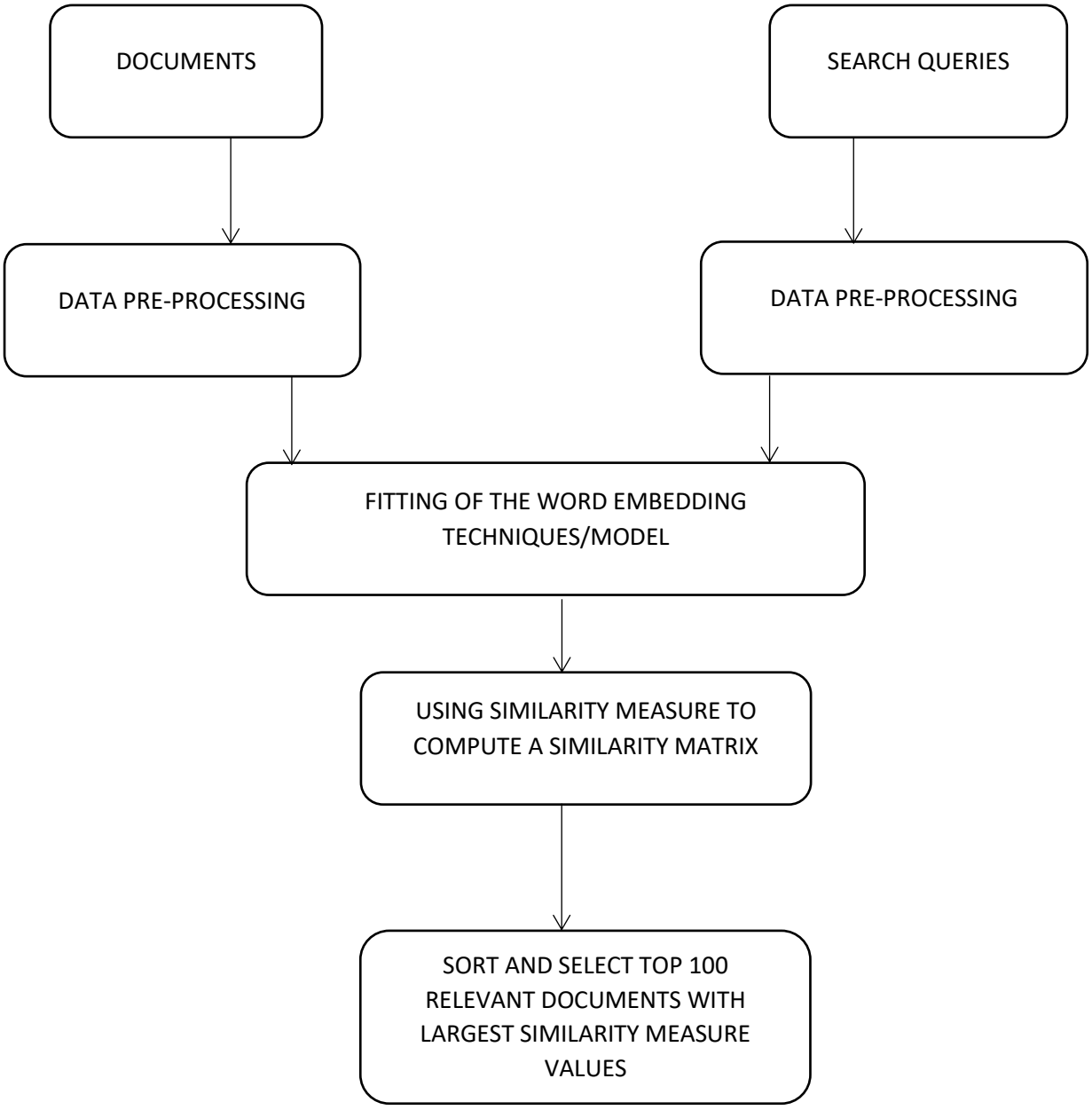


Figure 2- Information Retrieval Model as used in this Research

5. Results

To evaluate these approaches, we compare the retrieved documents with the relevant documents. The intersection between these two sets is used to determine the precision for the system. Precision is defined as the ratio of the retrieved documents that are relevant to the user, to the total retrieved documents by the system. Recall is defined as the ratio of the retrieved documents that are relevant to the user, to the total relevant documents in the dataset. However, to evaluate any IR system, Precision and Recall are looked at together through a measure called F-Score. In this measure, Precision and Recall are evenly weighted and thus it gives a better picture of the IR system.

Table 1- Precision for different Similarity Measures when used with Word2Vec

S. No.	Word Embedding Technique	Similarity Measure	Precision	Recall	F-Score
1	Word2Vec - CBOW Model	Cosine Similarity	7.0	25.0	10.94
2	Word2Vec - CBOW Model	Jaccard Similarity	7.5	26.785	11.72
3	Word2Vec - SkipGram Model	Word Movers' Distance	15.07	58.93	24.0

Word Mover's Distance performs significantly better than the other two measures when used with Word2Vec. This could be due to the fact that it overcomes the synonym problem and works only on the basis of the semantic meaning of the text.

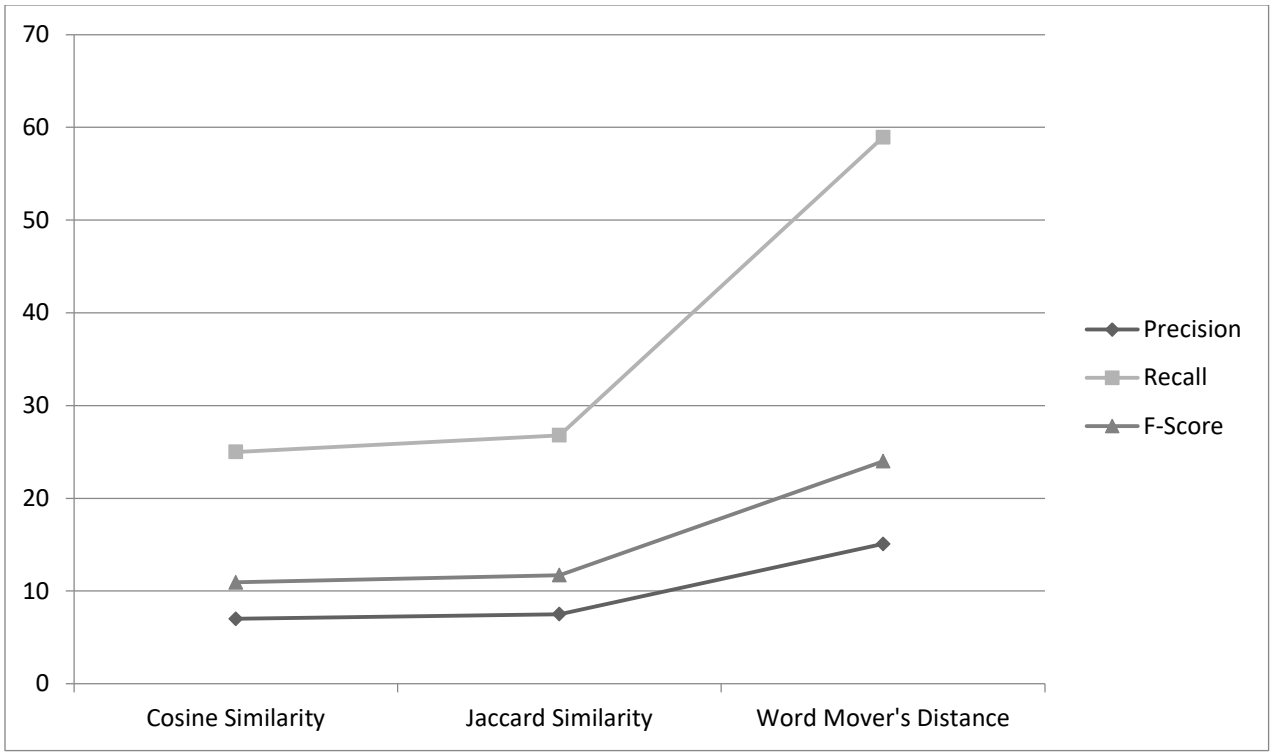


Figure 3- Comparison of different Semantic Similarity Approaches

6. Conclusion and Future Work

Information Retrieval provides a means to get information that already exists in electronic formats, which can be done using multiple approaches. In this application, it is essential to see the context of the query while retrieving documents. Hence, several new approaches to retrieve the most relevant documents were explored- by using Word2Vec to create word embedding's that represent the data numerically and then computing similarity of the query with the documents in the corpus. These similarity measures have shown great promise and are capable of retrieving relevant documents for the users.

These results were obtained when the research was done on the Classic4 Dataset, so for further analysis these approaches should also be applied to other, larger datasets.

References:

1 Slimani, Thabet. “Description and Evaluation of Semantic Similarity Measures Approaches.” International Journal of Computer Applications, vol. 80, no. 10, 2013, pp. 25–33., doi:10.5120/13897-1851.

2 Sitikhu, Pinky, et al. “A Comparison of Semantic Similarity Methods for Maximum Human Interpretability.” 2019 Artificial Intelligence for Transforming Business and Society (AITB), 2019, doi:10.1109/aitb48515.2019.8947433.

3 Sieg, Adrien. “Text Similarities: Estimate the Degree of Similarity between Two Texts.” Medium, Medium, 13 Nov. 2019, medium.com/@adriensieg/text-similarities-da019229c894.

4 Cambria, Erik, and Bebo White. “Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article].” IEEE Computational Intelligence Magazine, vol. 9, no. 2, 2014, pp. 48–57., doi:10.1109/mci.2014.2307227.

5 Pradhan, Nitesh, et al. “A Review on Text Similarity Technique Used in IR and Its Application.” International Journal of Computer Applications, vol. 120, no. 9, 2015, pp. 29–34., doi:10.5120/21257-4109.

6 International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume-4, Issue-7, Jul-2017, ”Query based Document Ranking for Enhanced Information Retrieval”, Tanuj Sharma, Kanika Mittal, Smriti Khurana, Anusha Chhabra

7 “Web Document Clustering and Ranking using Tf-Idf based Apriori Approach” Rajendra Kumar, Roul Omanwar, Rohit Devanand, S. K. Sahay

8 Singh, Jaswinder, et al. “A Study of Similarity Functions Used in Textual Information Retrieval in Wide Area Networks .” International Journal of Computer Science and Information Technologies, vol. 5, 2014, doi:10.1107/s0108768104025947/bm5015sup2.hkl

9 Intellica.AI. “Comparison of Different Word Embeddings on Text Similarity - A Use Case in NLP.” Medium, Medium, 4 Oct. 2019, medium.com/@Intellica.AI/comparison-of-different-word-embeddings-on-text-similarity-a-use-case-in-nlp-e83e08469c1c.

10 Gilyadov, Julian. “Word2Vec Explained.” Hacker's Blog, israelg99.github.io/2017-03-23-Word2Vec-Explained/.

11 Shperber, Gidi. “A Gentle Introduction to Doc2Vec.” Medium, Wisio, 5 Nov. 2019, medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e.

12 Prabhakaran, Selva. “Cosine Similarity - Understanding the Math and How It Works? (with Python).” Machine Learning Plus, 28 Apr. 2020, www.machinelearningplus.com/nlp/cosine-similarity/

13 Ma, Edward. “Word Distance between Word Embeddings.” *Medium*, Towards Data Science, 6 Sept. 2018, towardsdatascience.com/word-distance-between-word-embeddings-cc3e9cf1d632.

14 https://www.sciencedirect.com/topics/computer-science/jaccard-similarity

15 Ting, Kai Ming. “Precision and Recall.” *SpringerLink*, Springer, Boston, MA, 1 Jan. 1970, link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_652.

16 Classic4 Dataset: https://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/ [Referred On: 08 January 2020]