**Saif Ullah (19L-0953)**

**Abdullah Asim (19L-1160)**

**Word2vec**

Word2vec is a technique for natural language processing published in 2013. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector.

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. 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.

* **Training algorithm**

A Word2vec model can be trained with hierarchical softmax and/or negative sampling. To approximate the conditional log-likelihood a model seeks to maximize, the hierarchical softmax method uses a Huffman tree to reduce calculation.

* **Sub-sampling**

High-frequency words often provide little information. Words with a frequency above a certain threshold may be subsampled to speed up training

* **Dimensionality**

Quality of word embedding increases with higher dimensionality. But after reaching some point, marginal gain diminishes. Typically, the dimensionality of the vectors is set to be between 100 and 1,000.

* **Context Window**

The size of the context window determines how many words before and after a given word are included as context words of the given word. According to the authors' note, the recommended value is 10 for skip-gram and 5 for CBOW

**Literature Review**

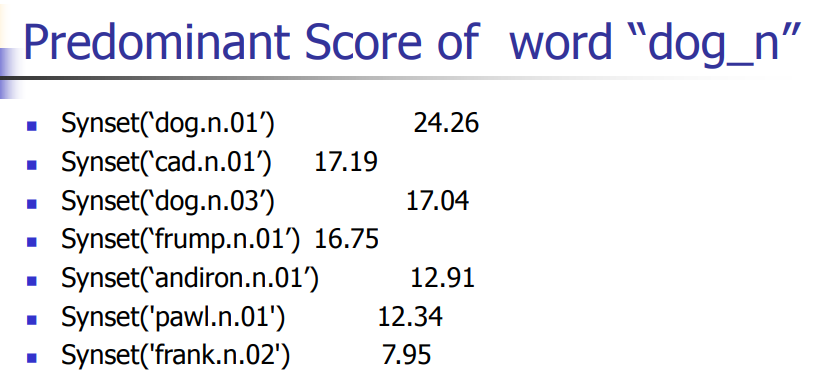
**Applications of Word2vec**

One can train the Word2vec on any data corpus e.g. English Corpus, French Corpus, German Corpus and so on. Following are some applications of Word2vec.

* **Finding Predominant Word Senses in Untagged text**

****

It can be seen that a score has been assigned to the word dog

****

* **Applications in Machine Translation**

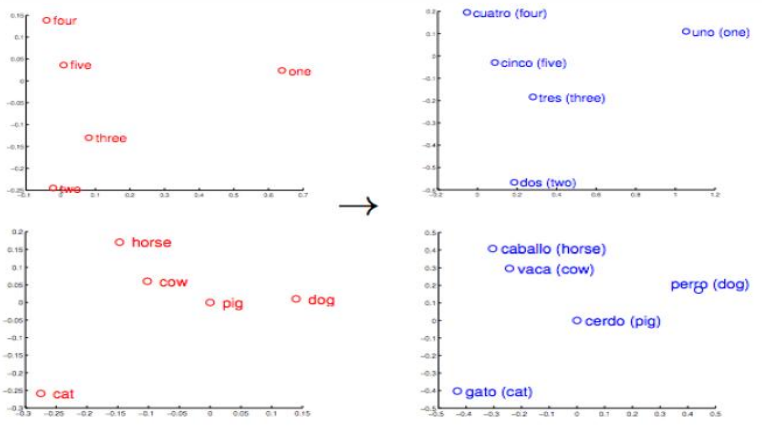
**1 -** Train word representations for e.g. English and Spanish separately

**2 -** The word vectors are similarly arranged!

**3 -** Learn a linear transform that (approximately) maps the word

**4 -** Vectors of English to the word vectors of their translations in Spanish

**5 -** Same transform for all vectors



**Paper Review**

**Word2Vec Model Analysis for**

**Semantic Similarities in English Words**

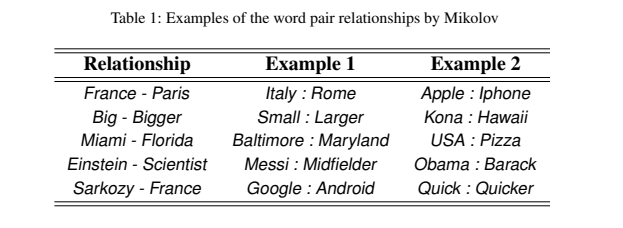
Semantic similarity has an important role in the field of linguistics, especially those related to the similarity of words meanings. Semantic similarity between words is the search for similarities between two words or more. In terms of the similarity of words meaning, two words may differ syntactically 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.

**Related Work**

In a previous study related to the Word2Vec model, I tried 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 the 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 be used as comparative material.

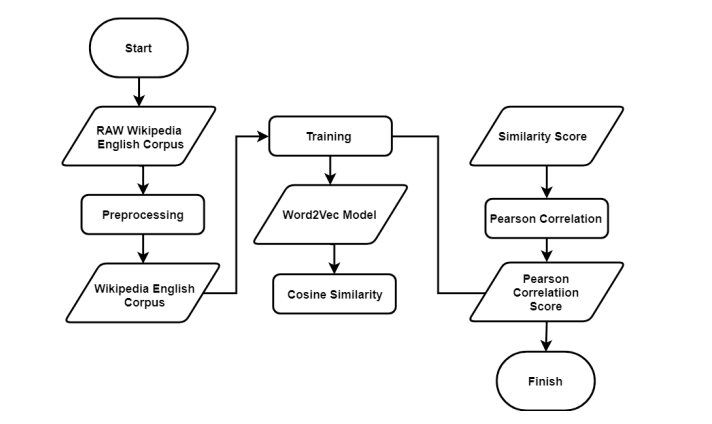
**Semantic Similarity**

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 of 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.



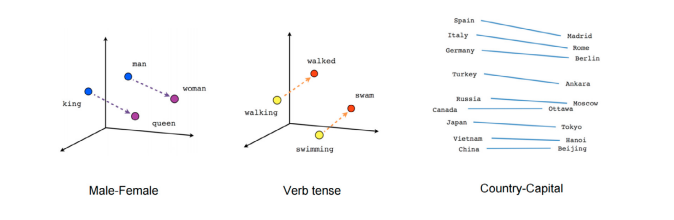
**System Overview**

Following Diagram will show the whole working



**Word Embeddings**

Word Embeddings is the collective name for a set of language modeling 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 vectors based on several features they have such as window 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 then 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 SkipGram 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.



**Data Set**

Data Set was wikipedia scraped data on which the embedding was built. Data first preprocessed which includes tokenization and removing some junk data.

**Summary:**

In the above mentioned literature review data is taken from wikipedia. However, in our project data is taken from various sources like BBC, CNN and from various books written by English authors.

**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;.

**10** Wikipedia