CS 410: Tech Review – Word2Vec

Submitted By:
Shubhendu Bhaskar

Introduction:

The goal of this paper is to perform a technical review on Word2Vec which is one of the techniques for natural language processing. This is an important natural language processing (NLP) technique which learns various word associations using a neural network model which after training can help one identify the synonyms or similar words for a sentence.

Important Concepts:

Word Embedding:

Word embedding is an important concept of the word2vec model. This means the process of converting a non-vectorized data into vectorized data by embedding the words in a vector space. For example, "This is an apple." A machine learning model does not understand this sentence. So, through word embedding, each word is converted to or mapped to a vector. Then word2vec learns the vector similar to a neural network. Then a detailed understanding of the word is done with regard to how it is used in the text or sentence. The characteristics of the word can be semantics, context, or definitions, etc. Using this. It becomes easier to identify the word similarity and dissimilarity with other words. The word embedding allows it to be converted into vectors that can be directly fed into the machine learning models.

Architecture:

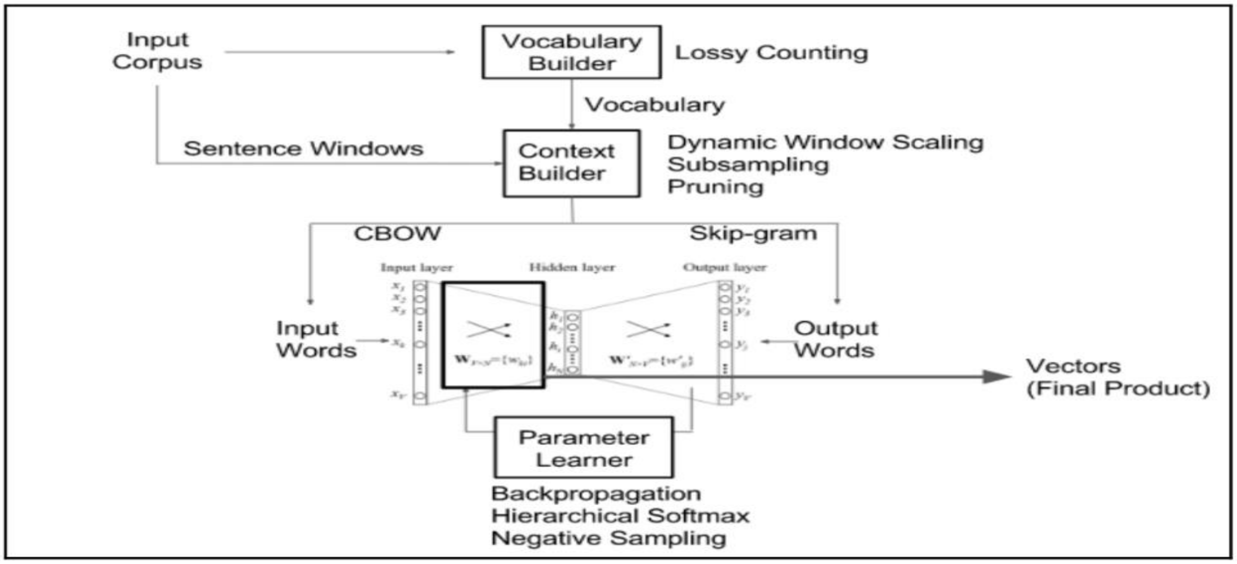
In the architecture for Word2Vec, there are three main blocks.

First is the vocabulary builder
Second is the context builder
Third is the Neural network with two layers

For creating word2Vec, the python library has genism. Below is the explanation and the pictorial representation of how the three blocks work to generate word2vec model:

1. Sentences act as the input data for the vocab builder which extracts unique words from the sentences to build the system vocabulary.

2. In the context builder, words are converted to vector form. The vocab builder that generates the index and count is fed into the context builder. From a set of words known as the context window, word pairings are formed which are later fed into the neural network.
3. There are two layers in this neural network – An input layer with a number of neurons same as number of words in the vocab builder in the training set. A second layer which is the hidden layer which contains the dimensions of the word vectors. The third output layer with number of neurons equal to the input layer.



One-Hot Coding:

This is an important concept where words are converted into their vector form. The size of the vector is equal to the number of words in the vocabulary and every word is assigned a different vector value.

Word2Vec Applications:

The word2vec NLP model is a simple model yet it is revolutionary. There are many applications of the word2Vec model, but it can be divided into two broad categories- Discovering knowledge and providing recommendations. In the case of knowledge discovery, word2vec has helped us to uncover vital information and figure out relationships between a disease and its related disease-genes that cause disease. So even if there are millions of abstracts spread across thousands of journals, the word2vec model learns associations between them .

Name Entity Recognition (NER) – word2vec finds similarity between the words using NER where all words can be fed together to get better results.

Sentiment Analysis and classification of documents – Sentiment analysis helps in understanding the context or words being used for example in the case of social media. Word2Vec also helps in accurately classifying the documents using simple statistics.

Word Clustering – Words with similar meaning are grouped together. One of the search engines that uses word2vec to improve its machine learning product is Google.

Conclusion:

Word2Vec overall is an essential part of NLP which helps machines understand language as understood by humans. Because there is a vast range of words, word embeddings help to find the similarity and dissimilarity between the words. To understand the word2vec architecture in detail – two other concepts should be learnt. One is the CBOW (Continuous bag of words) and the other is the skip-gram model. Skip-gram helps to determine the context and CBOW takes a group of words and predicts the missing one.

References:

1. https://medium.com/@vishwasbhanawat/the-architecture-of-word2vec-78659ceb6638
2. https://towardsdatascience.com/word2vec-explained-49c52b4ccb71
3. https://towardsdatascience.com/machine-learning-word-embedding-sentiment-classification-using-keras-b83c28087456
4. https://jalammar.github.io/illustrated-word2vec/