CS 410: Technological Review – Word2Vec

Chandan Goel (chandan3@illinois.edu)

Introduction:

The objective of this paper is to perform a technological review on one of the Natural Language Processing techniques called Word2Vec. As part of CS410 course, we were introduced to two key NLP techniques called Bag of Words (BOW) and TF-IDF. These techniques had one major flaw i.e., semantic information is not captured by these two approaches. Word2Vec is a modern, state of art NLP technique that solves this problem i.e., semantic information and relation between different words are captured. For example, with this technique we can determine that King – Man + Woman = Queen. In this paper, we will discuss how this algorithm is able to determine the similarity.

Word2Vec Concept:

Word2Vec technique was introduced in 2013 by a team of researchers led by Tomas Mikolov. This algorithm uses a neural network model to learn word associations from a large corpus of text. It is a shallow, two-layer neural network trained to reconstruct linguistic contexts of words. It takes in a large corpus of words as input and generates a vector space with each unique word in the corpus. Each word in this space can be represented with several hundred dimensions (instead of single number) and are positioned in the vector space such that words that share comment contexts in the corpus are near one another. This model can detect synonymous words or can suggest additional words for a partial sentence.

Word2Vec Architecture:

From a high level, a Word2Vec model consists of the following three building blocks,

- Vocabulary Builder
- Context Builder
- 2 Layer Neural Network

Vocabulary Builder:

- Basic building block of Word2Vec model.
- **Input**: Raw data in form of sentences.
- **Output**: Unique words to build vocabulary of the system.

Context Builder:

- Converts words to vector representations.
- Input: Vocabulary object (output of Vocabulary Builder)
- Output: Word pairings based on the context window (# of words: generally, 5 to 10 words)

2 Layer Neural Network

- Word2vec uses the neural network for training.
- Input Layer: # of neurons = # of words in vocabulary builder in training set.
- Hidden Layer: Dimensions of word vectors
- Output Layer: # of neurons as input layer

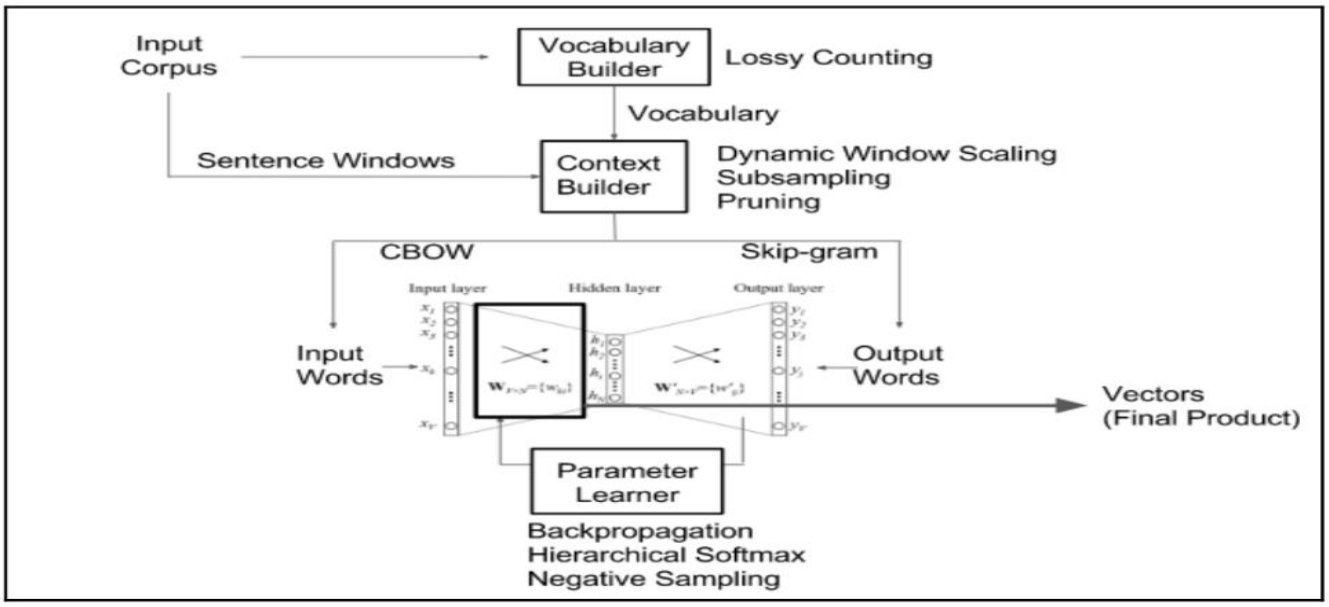

Figure 1: Word2Vec Architecture

Word2Vec Architecture: Types

Word2Vec trains words against other words that neighbors them in input corpus using the following two approaches

- **Continuous Bag of Words (CBOW):** using context of words to predict target word
- **Skip Gram:** using a word to predict a target context.

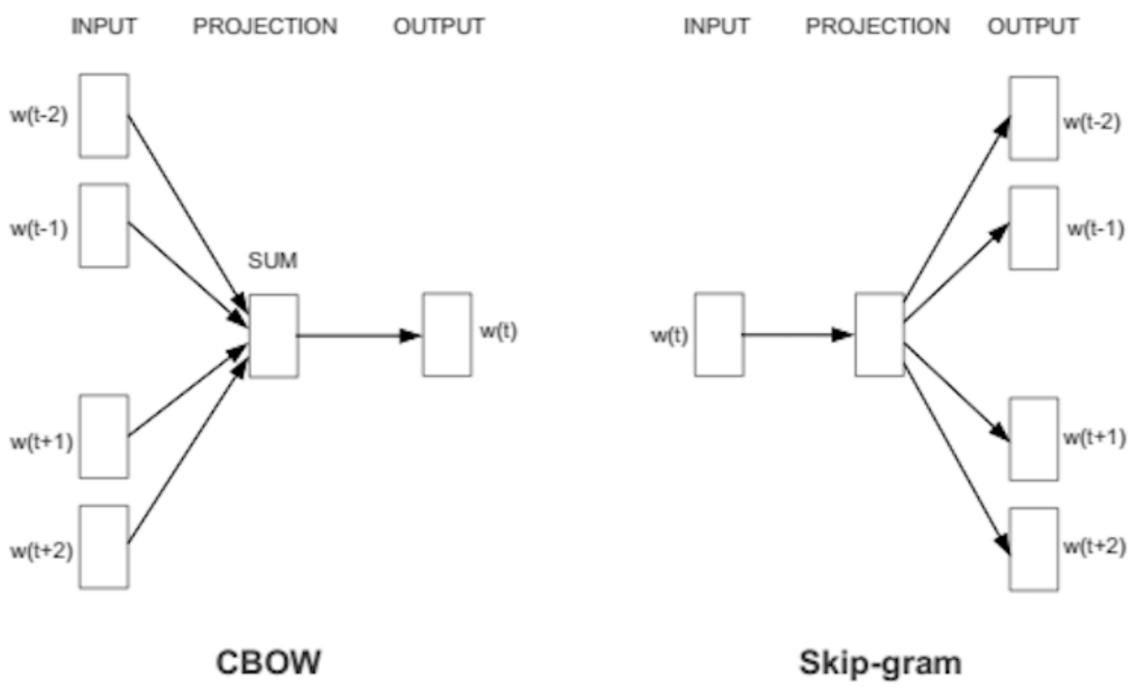

Figure 2: Word2Vec Architecture – CBOW vs Spik-gram

Word2Vec Applications:

Word2Vec is a modern, state of art NLP technique that can capture semantic information and relationship between different words. It is a simple idea that has revolutionized the field of NLP by allowing information systems capture the semantic context of sentences that has made them start to think and react like humans. Few of many applications of Word2Vec model include,

- Discovering knowledge:
- Name Entity Recognition (NER)
- Sentiment Analysis
- Word Clustering
- Recommendation

Conclusion:

Word2Vec is a state of art NLP technique that has revolutionize the way information systems can interact and communicate like humans. This technique has enabled the systems to capture the semantic information of a sentence which has been a problem with older techniques like BOWs and TF-IDF. Given a large corpus of text, Word2Vec produces an embedding vector associated with each word in the corpus such that words with similar meaning (characteristics) are near each other. Two main architectures associated with this technique include – CBOW and Skip-gram model.

Overall, Word2Vec is an essential part of Natural Language Processing which enables machines think and act like humans. It provides future researchers a foundational knowledge to build on top of. We are hopeful with advancement of technology, researchers and scientists will further improve the effectiveness of this technique.

References:

1. https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
2. https://israelg99.github.io/2017-03-23-Word2Vec-Explained/
3. https://wiki.pathmind.com/word2vec
4. https://towardsdatascience.com/word2vec-explained-49c52b4ccb71
5. https://medium.datadriveninvestor.com/how-does-word2vec-work-57e461f23f00
6. https://towardsdatascience.com/word2vec-models-are-simple-yet-revolutionary-de1fef544b87
7. https://medium.com/@vishwasbhanawat/the-architecture-of-word2vec-78659ceb6638
8. https://jalammar.github.io/illustrated-word2vec/
9. https://www.youtube.com/watch?v=hQwFeIupNP0
10. https://www.youtube.com/watch?v=Otde6VGvhWM&t=1055s
11. https://www.fer.unizg.hr/_download/repository/TAR-2020-reading-05.pdf