**Technical Blog for research paper -‘ Efficient Estimation of Word Representations in Vector Space’**

**PURPOSE**: To cover and convert all technical aspects of the WORD2VEC research paper, into blog post which is simple and understandable for all readers.

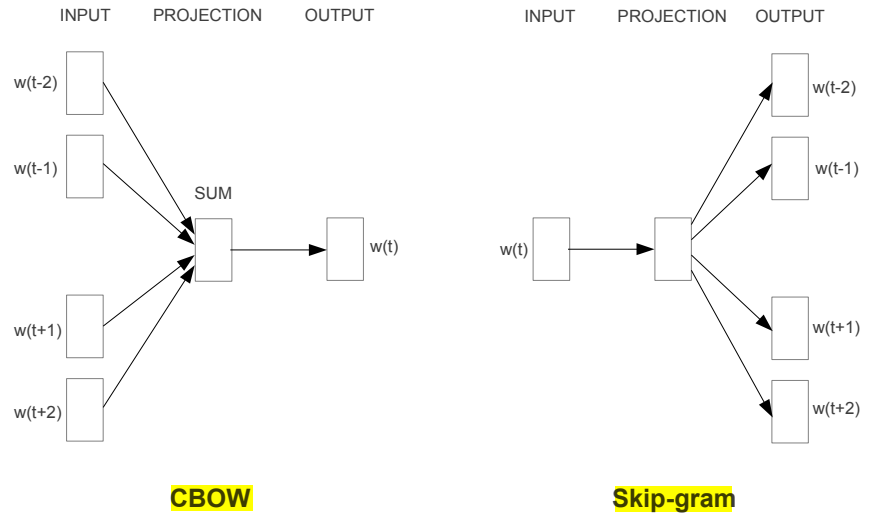
**Original Paper :**  <https://arxiv.org/pdf/1301.3781.pdf>

**INTRODUCTION**:

* For every word, a vector is associated. This means, for a sentence, a matrix will be associated. So for every word there will be some kind of mapping and this mapping is nothing but **word2vec**
* Instead of random assignment of the mapping (Many current NLP systems and techniques treat words as atomic units), if the words with similarity are mapped to the vectors such that the distance between the vectors is minimum, then it will make more sense.
* Above can be understood in form of analogy example - if**, man : woman :: then, king : ???** The obvious answer would be **queen**. Here four different vectors are linked to each other based on their meaning.
* There are few limitations with the existing simple techniques such as less amount of data, less accuracy, higher computation cost, etc.
* In the subsequent summary we will be understanding, how can **word2vec** maps word (**w**) to a vector (**v**) such that the vector satisfies these two properties –
  + Vectors should not be sparse,
  + Vector should understand meaning or context.

**WORD2VEC SUMMARY:**

* Parameters used for measuring the quality of the resulting vector representations are
  + Closeness of similar words.
  + Multiple degrees of similarity between the words.
* Issues caused due to previous models ():
  + Most of the complexity is caused by the non-linear hidden layer in the models.
  + Data representation is more precisely but efficiency is lost.
* Hence the aim is to explore simpler models (to train on much more data efficiently ) rather than Neural networks which represent the data more precisely but efficiency is lost. And to minimize computational complexity.
* Proposed models:
  + Continuous Bag-of-Words Model (CBOW): It predicts the current word based on the context. The order of words in the history does not influence the projection.
  + Continuous Skip-gram Model: It predicts surrounding words given the current word.
  + Due to the overhead of the distributed framework, the CPU usage of the CBOW model and the Skip-gram model are much closer to each other than their single-machine implementations



* Q. if**, man : woman :: then, king : ???** Simple algebraic operations with the vector representation of words
  + Step1: To compute vector X = vector(”woman”)−vector(”man”) +vector(”king”)
  + Step2: To search in the vector space for the word closest to X measured by cosine distance
  + Step3: When the word vectors are well trained, it is possible to find the correct answer (word queen)

**FINAL THOUGHTS AND CONCLUSION:**

* Word vectors can be used to significantly improve and simplify many NLP applications.
* All models are trained using stochastic gradient descent and backpropagation
* The prime objective of word2vec is to maximize the accuracy, while minimizing the computational complexity
* It is possible to train high quality word vectors using very simple model architectures using word2vec
* Applications of word2vec:
  + In automatic extension of facts in Knowledge Bases
  + For verification of correctness of existing facts
  + sentiment analysis
  + paraphrase detection
* Because of the much lower computational complexity, it is possible to compute very accurate high dimensional word vectors from a much larger data set. Hence word vector will be very important building block for future NLP applications.