# Efficient Estimation of Word Representations in Vector Space

**Authors:** Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean
**Publisher:** Computing Research Repository (Arxiv preprint)
**Year:** 2013

1. **Abstract:**

"We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities."

2. **Introduction:**
   - The purpose of the paper is to produce novel architectures for the learning of continuous word vector representations from huge data sets with millions and billions of words.
   - The approaches so far had not considered the relation of similarity in the words, due to the fact that it is very simple and robust.
   - Despite of being simple, they do have their own limitations in computing the word vectors from the complex and large datasets.
   - To consider an example, the simplest structure N-gram underperforms with neural network based models.
   - To avoid those limitations, the authors introduced two models for continuous word representations to learn from huge data and dimensionality. Moreover, they have created a test data for grading the syntactic and semantic accuracies.

3. **Related work:**
   - Many works had been done in creating neural network language model and they have been extended to adjustments for the better accuracy and precision.
   - All the works had been trained on the limited datasets with a simple dimensions of 50-100.
   - So, when the large and complex datasets had been trained on the similar architectures, they would be computationally expensive in training the data. For this reason new architectures are to be needed.

4. **Approach:**
    - The authors had used the feed forward language model with hierarchical softmax function, in which the vocabulary is described by a binary tree called as Huffman binary tree, to reduce the complexity of the learning.
    - Followed, they have also adopted the Recurrent Neural Network language model, which can create a short term memory. The past information can be represented by a hidden layer.
    - For training purposes, they have used the former methods and in addition the architectures they have introduced on a distributed framework called as DistBelief.
    - This distributed framework helps in training the data with multiple models (from tens to hundreds replicas) at a time and the algorithm adaptive gradient is needed for this type of frameworks, which can be called as Adagrad.
    - It is obvious that the complexity occurs due to the non-linear models of networks, which made authors to concentrate on simpler models to train huge data efficiently. This lead to introduction of two new architectures.
    - **Continuous Bag-of-words model:**
        - In this model, the hidden nonlinear network is removed and a new projection layer is used to result in the probable word. All the previous and future words are summed up to estimate the current word by summing the vectors.
        - The complexity of the networks is :
            - $Q = N \times D + D \times \log 2 (V )$, where Q = Complexity of the sentence
                - $N \times D$ = Dimensionality of the vector
                - V = Output layer dimensionality
        - The weight matrix for all the word positions is shared as in NNLM from input to the projection matrix.
    - **Continuous Skip gram model:**
        - In contrast to the CBOW model, in this model the previous word is used to create a range of words that can be occured with certain probabilities.
        - As the range increases, the computational expense also increases. But as we know that the most distant words will be less related, we will limit the range to a certain quantity.
        - The complexity of the model is:
            - $Q = C \times (D + D \times \log 2 (V ))$,where C = max.distance of the words.

- The test set had been defined with nine different types of syntactic questions and five types of semantic questions.
- The training epochs were used with stochastic gradient descent and the backpropagation.

## 5. Results:

- The results had been examined on both semantic and syntactic basis. Instead of showing the relations between the words, the authors tried a new way of testing the results.
- They have created a type of analogies and checked the solution by giving a new word. Surprisingly, it worked when the word vectors are well trained.
- The answers were evaluated correct for the given question when only the prediction is equal to the required answer. This eliminated a chance of counting synonyms into the correct answers.
- It is proved that the NNLM(Neural net Language Model) had performed well than the RNN and CBOW better that NNLM in the case of syntactic questions. Skip gram had underperformed to CBOW but better than NNLM in the syntactic case.
- In the instance of the semantic questions, Skip-gram model outperformed the CBOW and NNLM.
- In the Microsoft Sentence Completion challenge where the blanks of the sentences had to be filled, the Skip-gram had not performed better which is outperformed by RNNLMs. But the combination of both Skip-gram and RNNLMs had outperformed all the other methods with an accuracy of 58.9%.

## 6. Conclusion and Future Work:

- It proved to be a better approach for the large and complex datasets in NLP by the results provided.
- The new approaches had outperformed the syntactic and semantic orientations with respect to all other available approaches.
- The authors also claimed that the time and data limit due to their constraints had limited the accuracy, which would be better if the training is done with higher dimensionality.
- For the future, they had left the work to compare with other approaches like Latent Relational Analysis and several others.
- As a followed up work, the authors also created the models using C++ code, which resulted in the higher training speed than the former reports, which they stated, that is going to be published in NIPS 2013 paper.