The paper discusses the current neural network architecture for a **word2vec** model called the **Skip-gram** model, its limitations and the main enhancements for optimization of the model in terms of quality of results as well as training speed. Word2vec is a class of models that maps words or phrases from any given vocabulary to vectors of real numbers called **word embeddings.** Word2vec takes a large corpus of text as its input and produces an n-dimensional vector space (feature space) where each unique word in the corpus is assigned a corresponding vector in the space and vectors are positioned such that words that share common contexts in the corpus are located close to one another. Given a set of sentences or corpus, the Skip-gram model loops on the words of each sentence and uses the current word to predict the surrounding window of context words.

The objective of the skip-gram model is to '**learn**' word vector representations that would capture semantic and syntactic word relationships and prove useful in predicting nearby words.The skip-gram neural network 'learns' this by **training** on **word pairs** found in text documents. When these word pairs are fed into the model, the neural net outputs probabilities that relate to how likely it is to find each vocabulary word near the input word. The model maximizes the average log probability of the context word predicted, given the input word i.e. p(w0|wi) and in this process learns the **weights of the hidden layer**. These weights are the word embeddings and consequently the model learns the word embeddings that would maximise above mentioned probability. But, the current skip-gram model has certain limitations :

1. It currently does not deal with frequently occurring words i.e. stop-words like 'a', 'the' that rarely explain the context and provide much less information than infrequently occurring words
2. It relies on word vector representations rather than phrase representations, which leads to its inability to represent idiomatic phrases, for eg. 'Air Canada' that is not a natural combination of Air and Canada
3. Its output probability p(w0|wi) is defined using the **softmax function** and so, the ultimate training objective i.e. Δ log p(w0|wi) becomes proportional to the number of words in the vocabulary. Hence the training procedure is quite computationally expensive given a large vocabulary set.

The paper suggests extensions to the original model for alleviating the above problems.

1. **Subsampling -** For each word encountered in the training text, there is a chance that it will be effectively deleted from the text where the probability of deletion is related to the word's frequency. Thus, infrequent words are sampled more often than the frequent words to create new training data. It accelerates learning and even significantly improves the accuracy of the learned vectors of the rare words.
2. **Phrase Representations -** Extension from word to phrase vector representations by treating each phrase as an individual token in the training of the model
3. **Modifying the optimization objective :**
    a. **Hierarchical Softmax -** It uses a binary tree representation of the output layer such that every word 'w' can be reached by an appropriate path from the root of the tree; length of the path being L(w). Then the computation of the objective function Δ log p(w0|wi) becomes proportional to the length L(w0) of the path. This is less than or equal to the logarithm of the number of words in the vocabulary, hence, less computationally intensive
    b. **Negative Sampling -** Traditionally, when training a neural network, **all** of the neuron weights for a training example are tweaked slightly so that it predicts that training sample more accurately. In comparison, in negative sampling, for each training example only a small percentage of the weights are modified rather than all. This implies that with negative sampling, a small number of **'negative'** (for which the network outputs a zero in terms of one-hot encoding) samples are randomly selected for which the weights are updated.