Assignment - 3

Submitted by : Preetam Keshari Nahak
116CS0205

Introduction :

INPUT : Existing material master records with long descriptions I excel format

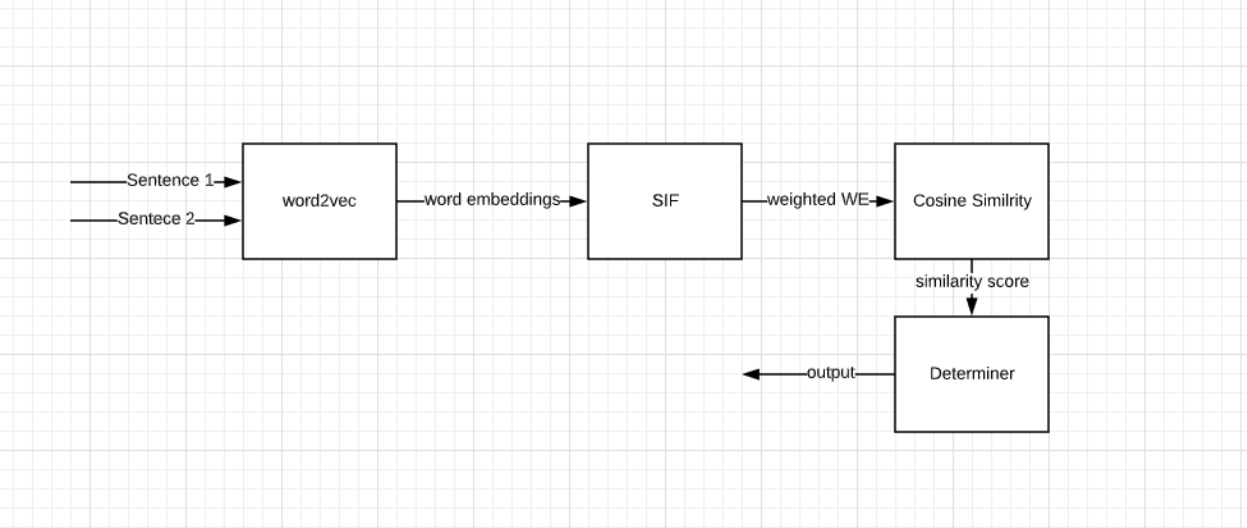
TASKS :

- Identify duplicates in the existing material records
- Prevent future duplication of material records

The problem statement boils down to implementing an efficient document matching / document similarity finding software using deep learning techniques which should be able to determine the duplicate records with reference to a master records dataset. With this understanding about the problem statement, below we propose an approach to address this problem.

Proposed approach :

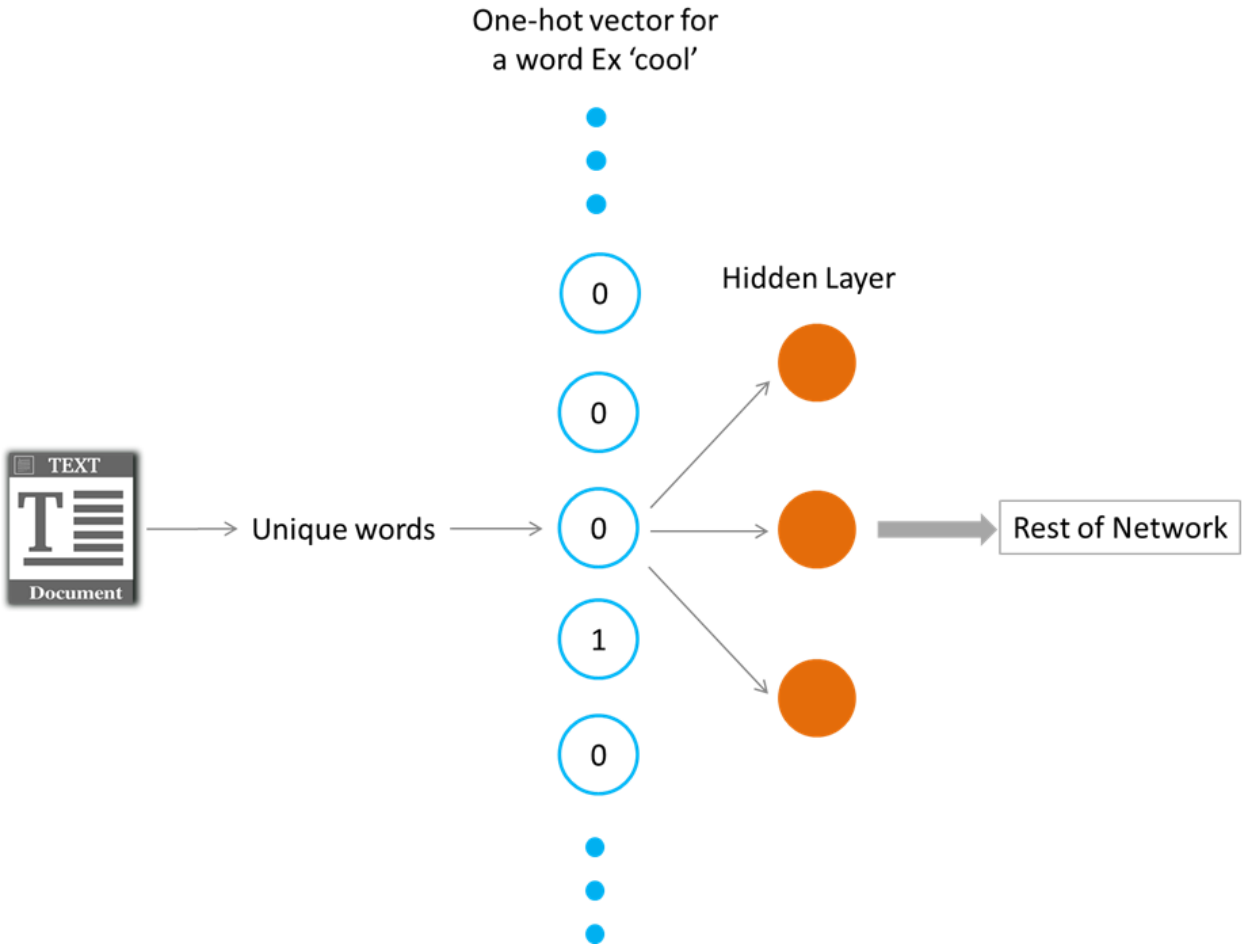
Different types of approaches can be adopted to solve the underlying problem. Here we will try to solve the problem based upon an efficient word embedding technique word2vec. The basic skeleton of this approach can be demonstrated as:



[Fig 1. Proposed architecture]

Basically the whole application consists of three main layers/modules. Word2vec, SIF and cosine-similarity module. The functionalities of the determiner module is described in further sections. Let's discuss the functionalities and usage of these one by one.

Objective of word embedding is to represent words as vector in a desired dimension say D, which will preserve the similarities among words with similar context. There are two types of word2vec techniques COBW (Common bag of words model) and skip-gram model. We will follow the skip-gram model for word embedding purpose. The basic structure of skip-gram model is shown below:



[Fig 2. Skip-gram word2vec method. src : Google]

From a set of text or documents, unique vocabulary set is created and each word is one-hot encoded. Then using a definite fixed window size say W, we find the leading W neighbor-hood words and following W neighbor-hood words of the word under consideration, say focused word. Then we prepare pairs of all possible

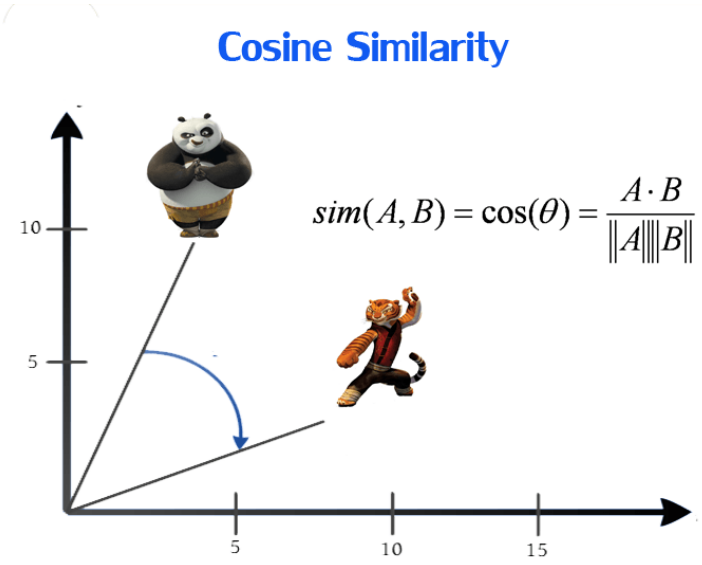
neighbor-hood combination with the focused word. We use these neighboring words as the possible output data for a given input focused word. The we prepare our dataset accordingly. We then feed this processed data to a neural network model with one hidden layer and train the model. After training, the weights available for the hidden layer values will be our word embedding matrix of dimension vocabulary_size*D. When we multiply the one-hot encoded form of a word with the word embedding matrix, we get the vector representation of the word of dimension D.

Next, we will discuss about our second module : SIF. SIF is also known as smooth inverse frequency. In order to represent a sentence as a vector in the D-dimensional space, we can follow two ways. First one is, taking average of all word embeddings of the words present in the sentence. But this method is not efficient as it gives equal weights to all words even which carry less information about the sentence. So instead of taking the average of all the word embeddings of all the words present in the sentence, we take weighted average of the word embeddings and the weights of the embeddings are determined by

$$a/(a + p(w))$$

where a is a parameter that is typically set to 0.001 and p(w) is the estimated frequency of the word in a reference corpus. So the most frequent word in the corpus will have more weight compared to the less frequent one. In this way we find vectorial representation of a sentence.

In the next step, we use cosine similarity to find similarity between two sentences. A figurative description is given below of the same. Cosine similarity is defined as the finding the angle between two vectors. Formula for finding this value is given below.



[Fig 3. Cosine similarity]

Then in the determiner module, we compare the cosine similarity value with a threshold in the determiner and decide whether we should discard or accept the document.

Methodology:

First we will create our vocabulary set of unique words present in the master material records. Then we feed the on-hot encoded form of each word to our word embedding model to find the word embedding matrix. We can use word2vec skip-gram model for this purpose. This word embedding matrix will be the backbone of the whole application. Using this matrix and SIF we prepare vector representation of all the long descriptions of the master material records.

Next, when a new material record is received, we find out the word embedding of the words present in the sentence and then use SIF to find its vector form.

Then in the determiner step, we will compare this vector with vector of all the master material records and calculate cosine similarities. As soon as the cosine similarity score exceeds a particular pre-defined threshold value, we term the new material record as duplicate one and discard it.

Conclusion :

Here we demonstrated an efficient approach to solve the desired problem using deep neural networks. The model can be improved with more data and better model architecture and training.