The Word Space Model: An Introduction

Asst. Prof. Arwinder Singh
Dept. of Computer Science, Mata Sundri University Girls College, Mansa, India
arwindersingh13@gmail.com

Abstract - The Word Space Model is a computational model to represent word meanings. This paper presents an introduction to the Word Space Model and its applications in various NLP tasks. This model is used to define similarity between words in the form of special proximity. WSM is a representation of large text in the form of vectors and then similarity is calculated using cosine angle between vectors. The earlier approaches were based on co-occurrences (Schutze, 1993 and Sahlgren, 2006) but modern models are the predictive models (Mikolov, 2013). WSM can be applied in various NLP approaches such as Information Retrieval, Question Answer, Word Sense Disambiguation, Query expansion and Summarization.

Keywords: Word Space Model, Random Indexing, Latent Semantic Analysis, Word2Vec.

I. INTRODUCTION

It becomes necessary for every language to define linguistically correct sentences, so we demand for a model of meanings. NLP’s various applications need for meanings such as lexicon acquisition, word sense disambiguation, machine translation, information access and dialogue systems. To overcome these problems and to improve accuracy of NLP applications, various models have been designed.

The representation of large amount of text on a space and to find the similarity between lexical items is a difficult task. The Word Space Model is a solution to this problem. WSM is a technique to represent large text in the form of vectors (Schutze, 1993). These vectors are high dimensional and sparse in nature. Then these vectors are used to find the similarity between lexical items. The similarity is cosine angle between vectors which represents less the angle means more similar the items.

WSM perform the similarity measure on words as well as phrases and documents.

The Word Space Model can be applied in various NLP applications. This is very useful for information retrieval so that most of the search engines are using word space models. It’s also helpful for word similarity as well as document similarity. The queries can be expanding with WSM. The word sense disambiguation and summarization can also be performed using WSM’s.

II. THE WORD SPACE MODEL

Word space model is to represent the meanings of lexical items using vectors. The advantage of WSM is that it can represent large amount of text in semantic space in the form of vectors. Then the similarity of lexical items is calculated from the angles between vectors.

It is a Computational Model of meaning which is designed to define meanings. The model is also known as model of meanings which is totally based upon language data to compute semantic similarity. As its name implies, this model is a representation of meanings of words in current context. It works with current available data, so the change in meanings of different words, will change the model.

The model shows semantic similarity between words by using n-dimensional space where n is an integer value from 1 to a very big number. So the point is that how this high dimensional model can be handled? The solution of this problem is to use 1-dimensional or 2-dimensional representation. As shown in the following figure, words close in space means they are similar in meaning and words far from each other represent their dissimilarity.

Figure: 1 (a) 1 dimensional word space and (b) is 2-dimensional word space
Co-occurrence matrix Word by Word

The working of word space model includes various steps to generate semantic similarity between words in source text. The first step is to generate co-occurrence matrix for the given text using window size one as shown in following example:

this is Punjabi University Patiala

<table>
<thead>
<tr>
<th>Word</th>
<th>Co-Occurrences</th>
<th>This University</th>
<th>is Patiala</th>
</tr>
</thead>
<tbody>
<tr>
<td>this</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>is</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Punjabi</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>University</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Patiala</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

A list for each word can be defined by row or column wise from co-occurrence count matrix, for example, list for Punjabi is (0, 1, 0, 1, 0) and this list is also considered as context vectors. These context vectors are the solution of distributional statistics to a geometric representation.

Co-occurrence matrix Word by Document

Let's say that each line in this example is a document, so we have three documents of two sentences each. We can represent this example with a token-document matrix or a type-document matrix. The token-document matrix has twelve rows, one for each token, and three columns, one for each line (Figure 1). The type-document matrix has nine rows, one for each type, and three columns. A row vector for a token has binary values: an element is 1 if the given token appears in the given document and 0 otherwise. A row vector for a type has integer values: an element is the frequency of the given type in the given document. These vectors are related, in that a type vector is the sum of the corresponding token vectors. For example, the row vector for the type Ever is the sum of the two token vectors for the two tokens of Ever.

Ever tried. Ever failed.
No matter. Try again.

 Fail again. Fail better.

Distributional Hypothesis

Because context vectors are built by looking at what contexts their words occur in, it can be said that words that have occurred in similar contexts will have similar context vectors, and words with similar context vectors have occurred in similar contexts.

Words occur in similar contexts \( \iff \) Words have similar context vectors

Mathematical forms of similarity

Context vectors which are generated in previous step are further used to determine vector similarity. This similarity can be computed with various methods which can be separated into two categories i.e. similarity measure and distance measure. Distance measure means small distance between words and the inverse of this distance is equals to similarity measure.

\[
\text{sim} (x, y) = \frac{1}{\text{dis} (x, y)}
\]

Scalar product of two vectors is also a measurement of similarity:

\[
\text{sim}_S(x, y) = x_1y_1 + x_2y_2 + \ldots + x_ny_n
\]

Euclidian distance is:
International Journal of Scientific Research in Computer Science Applications and Management Studies

\[ dist_k(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]

Minkowski’s method is:

\[ dist_m(x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^N \right)^{\frac{1}{N}} \]

Euclidian and Minkowski’s methods are not sufficient for similarity measurement. The suitable method for similarity measurement is used in Word Space Model which computes cosine of the angles with in vectors:

\[ sim_{\cos}(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \]

III. APPLICATIONS OF WORD SPACE MODEL

Word Space Model approach is useful for various NLP applications. The following are the semantic applications of WSM:

A. Information Retrieval

There is lot of information stored and search engines work to find the data from stored text according search query. WSM works fine to search similar data from stored documents. This approach was used for the first time in document retrieval by (Salton et al., 1975) and till now lot of work has been done and various approaches were discovered for improvements. Random Indexing (Sahlgren, 2005) was played an important role for smoothing complexity and then incremental SVD was recommended for scaling problems.

B. Question answering system

To generate the answers for asked query is a challenging task in NLP. The working of question answering system is to find the similar text from stored documents. The fundamental task of WSM is to find the similarity between two documents. So WSM can be applied to question answering system (Tellex et al., 2003) and produces good results.

C. Word similarity

Word similarity means how much two words are similar to each other. The vector model is useful to find the closeness between words by finding the angle between vectors. Each of the word is represented as vectors of the same dimension. Deerwester et al. (1990) researched that word similarity can be generated from vectors. After that various research has been done to find similar words such as synonyms.

D. Word and Document clustering

There are lot of documents which have similar data as well as dissimilar. The clustering of documents means to store group of data in smaller groups. The first step for clustering is to generated vectors and each value of a vector tells the importance of a word in the document. Then appropriate algorithms take these vectors as input and produce groups of data.

E. Word Sense Disambiguation

To generate the senses for words in Natural Language Processing environment is known as WSD. A word-context frequency matrix used by Towell and Voorhees (1993) for Word Sense Disambiguation where each of the vector corresponding to a type annotated with a sense tag. The word senses have been also generated by using Random Indexing by (Niladri Chatterjee and Shiwal Mohan, 2008).

F. Query Expansion

All the search engines take a query as an input find the results. In some cases, search engines unable to find results related to submitted. So need to expand the query for more and accurate results. WSM’s are doing well in query expansion and for more results (Cao, Jiang, Pei, He, Liao, Chen, & Li, 2008).

G. Text Summarization

To generate the summary of various documents means to find the important sentences from various documents. There is a big literature in text summarization system and developed using different approaches. Extractive text summarization is defined by (Mani, 2001) which means to collect important sentences from original documents. Niladri Chatterjee and Pramod Kumar Sahoo used WSM to generate extractive text summarization.

IV. PREPROCESSING OF DATA

The input for the model will be the plain text which is used to find similarity. Some systems use this whole text as an input but some models require preprocessing of data. To modify the input text is known as preprocessing which includes stemming, lemmatization, removing stop words and frequency threshold.

Stop words are common words in every data, so need to remove them to avoid larger dimensionality. Stop words such as: of, the, a, an, is, are etc. Some of the words can be repeating words which generate a higher frequency. This higher frequency doesn’t represent specific meanings. So the researchers have
Stemming and lemmatization is a process to cut off some characters from beginning of a word or from end. This is done to reduce the number of words from vocabulary, for ex. Study can be represented as studies or studying. So by removing last three characters, it becomes stud and study. Stemming has some disadvantages because it does not produces accurate words all the times. The solution is to use lemmatization.

Lemmatization removes inflections very carefully. It uses dictionaries to retain the original meaning of words. So studies and studying will be modified as study all the time.

V. PARAMETERS OF WSM

Parameters setting for implementation of WSM is a fundamental step to gain the good results from word space model. There are some parameters which need to be defined to get good results. Following are the basic parameters of WSM:

A. Vector size

The vector size is equal to the number of unique words in the corpus. As discussed in section 2, each word is represented as a vector as shown in Table 1. Here, the vector size is 5 because unique words are 5 in the input data. The vector size represents the dimensionality of our word space. To design a good WSM, use small vector size as compare to vocabulary. Nilardri Chatterji and PK Sahoo (2014) used formula $dRI=2\sqrt{m}$ to set the context vector size in implementation of RI where m is the number of distinct words and $\|$ is ceiling function. So various researchers have set the vector size manually instead of using this formula. K. Douwe and C. Stephen (2014) used 50,000 as vector size.

B. Window size

The window size means the number of words to left and right from current word. As discussed in the example in Table 1 of section 2, window size one means one preceded word and one succeeded word. If we consider window size of two, then two words to the left and two words to the right will be considered. The window size can vary from one to five, ten or fifty. To choose the window size depends upon your work. A smaller window size work better to find similarity of concrete nouns as defined by Hill et al. (2013) whereas larger window size good for abstract nouns. K. Douwe and C. Stephen (2014) uses 3, 7, 5, 9 and a full sentence as window size.

C. Context

The context is the list of surrounding words of the target word. To design co-occurrence matrix, context is defined by given window size. So changing the size of window will affect the context.

D. Stop words removal

This parameter is very important for defining the size of the vector. The size of the vector depends upon the number of unique words in the corpus. Stop words are those which are occurring again and again such as the, and, a, is, are. To remove these words from the corpus will decrease the size of the vector.

E. High frequency cut off

There can be various repeated words in the list which gain the maximum frequency as compare to other words. High frequency words are not useful in WSM’s. So need to remove these words from the corpus.

VI. THE PROBLEMS WITH WSM

A. The Trouble of High Sparseness

There is a problem with context vectors which are generated by using co-occurrence matrix. The problem is the presence of more zeros in co-occurrence matrix. These zeros representing that most of the words are not co-occur. So, now need to reduce the size of matrix which is known as dimensionality reduction. This reduction process can be implemented by using dimensionality reduction approaches.

B. Dimensionality

Word Space Model takes plain text as input and represent the data on large space. This space means thousands of dimensions. The input text is represented in n dimensional vectors where n is equal to the vocabulary in the document. To increase the vocabulary means the increment in dimensions.

C. Dimensionality Reduction

Implementing data in word space produces very high dimensions and data sparseness also. So need to reduce these dimensions into low dimensional data without losing the meanings of data. Some approaches reduce dimensionality by performing two steps i.e. first to create co-occurrence matrix and the performing dimensionality reduction technique (LSA). Some of the approaches reduced dimensionality in a single step such as Random Indexing(RI).

VII. IMPLEMENTATION OF WSM

The WSM as discussed in section 1 is a mathematical approach to represent large text in the
form of vectors. From the implementation point of view, plain text can be represented as vectors and then these vectors can be used for different NLP tasks. There are two major approaches to generate large data in the form of vectors i.e. traditional Distributed Semantic Models and novel Neural Network Models. DSM are also known as count models which means these models work with co-occurrence matrices. On the other hand, Neural Network Models work with predictions. The aim of this paper is just to introduce the different approaches of the implementation of WSM.

A. Latent Semantic Analysis

LSA is a technique to reduce the high dimensional data representation into low dimensions. LSA was developed (Dumais et al., 1988; Deerwester et al., 1990) in 1980’s as an extension to VSM in information retrieval system. The traditional VSM was unable to handle synonyms, then LSA is designed to do this. Because traditional word space model can’t distinguish between boat and ship at the time of information retrieval.

The implementation of LSA is a two step process, one is to generate count matrix and second is to perform SVD (Singular Value Decomposition) (Berry et al., 1995). SVD is a matrix factorization approach which break down the original matrix into various other matrices (approximately three) and these smaller matrices can be multiplied to produce the original one.

Finally LSA is a:

- words-by-documents matrix;
- according to the formula;

\[
fi = (\log (TFij) + 1)^{\alpha} \left( 1 - \frac{\sum_j p_{ij} \log \frac{p_{ij}}{\log D}}{\log D} \right)
\]

- truncated SVD for dimensionality reduction;
- cosine similarity is used to find similarity between vectors.

B. Random Indexing

The Random Indexing technique can be described as a two-step operation:

First, each context (e.g. each document or each word) in the data is assigned a unique and randomly generated representation called an index vector. These index vectors are sparse, high-dimensional, and ternary, which means that their dimensionality (d) is on the order of thousands.

Then, context vectors are produced by scanning through the text, and each time a word occurs in a context (e.g. in a document, or within a sliding context window), that context's d-dimensional index vector is added to the context vector for the word in question. Words are thus represented by d-dimensional context vectors that are effectively the sum of the words' contexts.

Example

\[
\begin{align*}
\text{w1} & \rightarrow 0 0 0 0 \\
\text{w2} & \rightarrow 0 0 0 0 \\
\text{w3} & \rightarrow 0 0 0 0 \\
\text{w4} & \rightarrow 0 0 0 0 \\
\end{align*}
\]

Instead assign a vector to each neighbour word where each vector consists of just zeros and a single 1. No two vectors can have a 1 in the same position. So these could be the vectors assigned to our 4 neighbour words:

\[
\begin{align*}
\text{n1} & \rightarrow (1 0 0 0) \\
\text{n2} & \rightarrow (0 1 0 0) \\
\text{n3} & \rightarrow (0 0 1 0) \\
\text{n4} & \rightarrow (0 0 0 1) \\
\end{align*}
\]

Now when you encounter neighbour word n4 next to word in context w2, you just take w2's row, which is a vector, and add it to n4's vector:

\[
\begin{align*}
\text{w2} & \rightarrow (0 0 0 0) + \\
\text{n4} & \rightarrow (0 0 0 1) \\
\end{align*}
\]

Do this for every neighbour next to every word in the context and you will have your co-occurrence matrix.

C. Word2Vec

Word2Vec is another technique to represent words with vectors to describe similarity between words. This is a famous word embedding model and is also known as starter for Deep Learning. It’s a predictive model to learn the word embedding from large text. The basic purpose of the model is to group the similar items and also to preserve the semantic relations between words. The famous example of word2vec is \( v(king) - v(man) + v(woman) = v(queen) \). Now, we can cluster the words, can define the relationship between words and can find similar words. The model can be defined in two approaches i.e. Continuous Bag of Words Model (CBow) and Skip Gram (SG). This
model represents the relationship between different words as shown in the Figure 2.

\[ \text{WOMAN} \quad \text{AUNT} \]
\[ \text{MAN} \quad \text{UNCLE} \]
\[ \text{QUEEN} \quad \text{KING} \]

Figure 2

**Continuous Bag of Words Model (CBoW)**

This approach is to predict the word from its context (Mikolov, 2013a, Mikolov, 2013b). The context may a single word or group of words. The input for the model will be \( w(i-2), w(i-1), w(i+1), w(i+2) \) and predict the \( w(i) \) word as explained in below diagram. The prediction of target word depends upon its context. This approach produces results with high accuracy but requires large amount of data. CBOW works better for frequent words. The main advantage of the model is that is low on memory. It doesn’t require large RAM as like co-occurrence matrix.

**Skip Gram**

The skip gram model is opposite to CBOW. This model predicts the context from given word (Mikolov, 2013a, Mikolov, 2013b). For example, \( w_1 \) is the given word and the output will be \( w(i-2), w(i-1), w(i+1), w(i+2) \). This approach works better for infrequent words. The context can be one word or multiple words.

VIII. CONCLUSION

As discussed in previous sections, traditional word space models are different from modern word space model. Traditional models were known as count models and present word space models are predictive models. The word space models a computation model which is used to define large text in the form stores the semantic relations between words. So this vector model can be applied in various NLP approaches as discussed in section 3. Deep Learning techniques are famous now days and are producing good results.

REFERENCES


