Comparative Study on Different Word Embedding Techniques

S.Lovelyn Rose[1]
{slr.cse@psgtech.ac.in[1]}

Professor (Department of CSE), PSG College of Technology, Coimbatore, India [1]

Abstract. Recent advancements in distributional semantics allow a word to be represented in a multi-dimensional semantic space as vectors. Words that are semantically and syntactically similar are closer in value. Popular methods to generate such embeddings include neural networks, reduction of word-to-word co-occurrence matrices, contextual representation and so on. Most embedding techniques are based upon the distributional hypothesis which states that words in the same context are semantically similar and hence can be broadly classified into count based methods and predictive methods. Word embeddings are quite popular in the field of Natural Language Processing in addition to other fields such as text mining, Sentiment analysis, Information Retrieval, Polarity Detection and so on. This paper aims to make a comprehensive study on the different word embedding techniques available.

Keywords: Word Embedding; Term-Weighting; Glove; word2vec.

1 Introduction

With the large availability of information as unstructured digital data,the retrieval, analysis, representation of information has become quite complex.A word representation is a mathematical object associated with each word, usually a vector. This vector is multidimensional and the value in each dimension corresponds to a feature which might have a semantic or grammatical interpretation. These are the word features[1]. Polysemy (same word having different meaning) and synonymy (multiple words having the same meaning) are examples of phenomena that lead to ambiguity in natural language and are complex to resolve.

A simple representation technique is the one hot vector encoding scheme in which the terms are represented as binary vectors .The size of each vector will equal the size of the vocabulary. The binary vector will have a value of 1 at the index of the word and all the other values in the vector will be zero. But since the words are unordered, contextualInformation cannot be represented and the relevance of the word with respect to the frequency is also unknown.

The most common way to classify unstructured data is to convert it to a bag of words, create a feature vector with weightings for the terms and finally using a machine learning algorithm like SVM, naïve bayes, maximum entropy for the actual classification. The bag of words (BoW) model looks at the document as just a collection of words. It is called a “bag” of words because it does not consider the ordering of the words or the grammar. It only takes into account the presence of the word in the document and not the context of the word.

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314494

The bag-of-ngrams model is a minor improvement over the BoW model in which the order is looked at in a small range. In this model, unigrams (single words), bigrams(all pairs of words in a sentence), trigrams (triplets of words in a sentence) etc. can be used to embed contextual information in a minimal range.

Early work found its application in information retrieval and the focus was on improving the process of retrieving relevant documents. Nowadays it is used for text classification, opinion mining/ sentiment analysis, polarity detection, automatic summarization,Machine translation, Named entity resolution, Chat-bot , Speech recognition, Question answering etc.

Most IR techniques use Vector Space models(VSM). The weight of each term in a document's vector is the key component which signifies the relative importance of the term with respect to the document.[2] The vector space model allows unstructured textual data to be represented as data points in an n-dimensional space. The 'n' dimensions denote the 'n' features that represent the textual data. The data point reflects the value assigned to the features. The features are referred to as terms and the value assigned to the term is the term weight.

Term weighting aims to evaluate the importance of different terms in VSM.The three major components in a term weighting scheme are the local weight, global weight and the normalization factor[3].

A distributed representation is a denser kind of word representation and it typically uses different neural network models. In ANN based models, the general architecture is to feed the words as one hot vectors in the input neurons and get the words as output. So, the dimension of the input and output vectors are the same. The hidden layer has a reduced dimension to capture the contextual information of the word vector. An auto encoder is a type of neural network model which has three layers: Input layer, Encoding layer(Hidden layer)and the Decoding layer(Output layer).It tries to reconstruct its input during which process it effectively creates a reduced dimensional representation of the input in the encoding layer.

This paper makes a comprehensive survey of the various term weighting schemes for the purpose of text classification and also a brief study on the different embedding techniques available.

2 Statistical Term Weighting

The main objective is to increase the value of the terms that have more significance in tasks like information retrieval, text classification, opinion mining, sentiment analysis etc. Zipfs word frequency law in natural language states that given a corpus the frequency of any word is inversely proportional to its rank in the frequency table[4]. Some observations are:

i) The high frequency terms are less in number when compared to low frequency terms
ii) The low frequency terms have more power to discriminate documents
iii) Longer the document larger the number of terms
iv) Longer the document larger will be frequency of occurrence of the terms.

These observations result in the three main components that constitute statistical term weighting using statistical linguistics namely, term frequency, document frequency and document length normalization. The final term weight will be a product of all these components.

Notations

Consider a collection of 'm' entities. The term document is not used because the entity might be document, tweet, facebook post, comment etc. Each entity is represented using 'n' features as in eqn. (1)

$$E= \{\phi_1, \phi_2, \phi_3, \quad \phi_v\} \tag{1}$$

$|f_i|$ denotes the number of occurrences of feature f_i.

df_i – number of documents in which feature f_i occurs

w_{ij}– weight of the i^{th} term in the j^{th} collection c_j

k_1, b - parameters of BM25 are set to their default values of 1.2 and 0.95 respectively[5][6]

dl – total number of terms in a document. i.e. the document length

avg_dl – average document length in the collection

df_n– number of documents not in class 'C' containing feature 'f'

df_p – number of documents in class 'C' containing feature 'f'

D_n – Number of documents not in class 'C'

D_p- Number of documents in class 'C'

3 Term Frequency Component

This component weighs a term based on its presence and the number of times it is present. To reduce the effect of larger numbers the term frequency is normalized with logarithm. i.e. a term frequency twice another term frequency should not imply twice the importance. High term frequency is due to the length of the document and does not necessarily reflect the importance of the term.

Boolean scheme simply denotes the presence of absence of a term in a vector.It however imposes difficulty in assessment of relevancy between terms.

Natural term frequency denotes the number of times a term has occurred in a document. However, the use of this factor alone in calculating the term weights in a collection of documents does not guarantee adequate retrieval performance. For example,stop words, which appear in the text but carry little meaning, serving only a syntactic function but not indicating subject matter have a very high frequency and tend to diminish the impact of frequency differences among less common words, affecting the weighting process.

Variations to the natural term frequency like the Augmented double normalized term frequency and INQUERY have tried to alleviate the effect of long documents over short ones by dividing by the maximum term frequency possible. In information retrieval, the number of terms matching the query term is of importance in retrieving a document. To reduce the effect of documents with more terms,

Okapi-BM25 divides the document length by the average number of terms in the collection. This weighting function is based on the traditional Probabilistic Retrieval Model. The basic principle is that a specific document could be judged relevant to a specific query, based on the assumption that the terms are distributed differently and independently in relevant and non relevant documents. The weight of a given term is calculated on the basis of the presence or absence of query terms in each document in the collection. Terms that have appeared in previously retrieved relevant documents for a given query should be given a higher weight than if they had not appeared in those relevant documents.

Name of the Scheme	SMART Notation	Formula
Boolean	b	1 if $f_i > 0$ 0 otherwise
Natural term frequency	n	$\|f_i\|$
Log normalized term frequency	l	$1+\log_2 \|f_i\|$
Augmented double normalized term frequency	a	$0.5 + 0.5.\frac{\|f_i\|}{\max_i(\|f_i\|)}$
INQUERY	i	$0.4 + 0.6.\frac{\|f_i\|}{\max_i(\|f_i\|)}$
Okapi-BM25	o	$\frac{(k_1+1).f_i}{k_1\left((1-b)+b.\frac{dl}{avg\_dl}\right)+f_i}$

4 Global Component

This component is used to give more weight to the terms with higher discriminating power for the task at hand. It is based on the intuition that terms that occur frequently in the document of interest and very less in other documents has high discriminating power. For text classification it is further classified as supervised and unsupervised based on whether the class information is used by this component.

The term frequency indicates the importance of the term in a given document, but knowing the term importance in a collection of documents is also significant. Term frequency was criticized as a method of determining term significance because in its simplest form, it treats all terms equally based on raw count, which does not take into account the term's discriminating power. To resolve this problem Sparck Jones[6] suggested the use of the relative collection frequency or inverse document frequency (IDF), making the frequency of the term in the collection as a whole a variable in retrieval. IDF places greater emphasis on the value of a term as a means of distinguishing one document from another than on its value as an indication of the content of the document itself.

Information Gain scheme proposed by Yang and Pedersen (1997) tries to find out how well each single feature discriminates the given dataset. Information entropy is used to measure the degree of uncertainty. Information Gain is the overall entropy of the training set minus the entropy of the feature. Thus, it measures how much the feature reduces the dataset's uncertainty when it is observed.

Chi-square(Yang and Pedersen, 1997) measures the lack of independence between the feature and the category, the higher value of the χ^2, the closer relationship the feature and the class have.

Relevance frequency boosts the terms which have high frequency in the positive category, that, helps in selecting the positive samples from the negative ones (Lan et al.(2009)).

Name of the scheme	Notation	Formula
Inverse Document Frequency	t	$\log_2 \frac{\lvert f_i \rvert}{df_i}$
Delta Inverse Document Frequency	Δidf	$\log_2 \frac{(D_n . df_p(f_i) + 1)}{(D_p . df_n(f_i) + 1)}$
Inverse category Frequency	icf	$\log_2 \frac{\lvert C \rvert}{cf(f_i)}$
Information Gain	IG(f_i)	$-\Sigma_{c \in C} p(c).\log(p(c)) + \left(\frac{df}{N}.-\Sigma_{c \in C} p(c\lvert f).\log(p(c\lvert f))\right) + \left(\frac{df}{N}.-\Sigma_{c \in C} p(c\lvert \bar{f}).\log\left(p(c\lvert \bar{f})\right)\right)$
Chi-square	χ^2	$\frac{N.((df_c.d\bar{f_{\bar{c}}}) - (df_{\bar{c}}d\bar{f_c}))^2}{df.d\bar{f}.N_c.N_{\bar{c}}}$
Relevance Frequency	rf	$\log(2 + \frac{df_p}{\max(1, df_n)})$

5 Normalization Component

The importance of the terms should not be affected by the length of the documents. To eliminate the length effect, we use the cosinenormalization to limit the term weighting range within (0-1). Specially, the binary feature representation does not use any normalization since the original value is 0 or 1.Ferguson et.al showed that the normalization component reduced the performance when used in short documents like tweets.

The higher the value of the normalization factor for a document is, the lower are the chances of retrieval for that document. In effect, the probability of retrieval of a document is inversely related to the normalization factor used in the term weight estimation for that document. This relationship suggests that to boost the chances of retrieval for documents of a certain length, we should lower the value of the normalization factor for those documents, and vice-versa. The pivoted normalization scheme is based on this principle.

Name of the scheme	Notation	Formula
Cosine normalized	c	$\sqrt{\sum_{\forall i in d j} w_{ij}^2}$
Pivoted unique normalization	u	$(1 - slope) * pivot + slope * c$
Byte size normalization	b	$\frac{1}{charlength^{\propto}}$

6 Types of term weighting schemes

Traditionally, the words are weighted based on their occurrence statistics. The later stages in term weighting has seen a shift towards considering the contextual word co-occurrence information for document representation. Instead of assigning weights in the term vector space, we are now assigning weights in the semantic vector space of the document. They are termed distributed representation because the representation of a word is not confined to a particular value in the vector. But rather multiple values represent a word and a value represents multiple words.

Term weighting models can be broadly classified based on the type of information to be weighted upon: Statistic or Semantic.

Statistical models rely on the local ,global weighting schemes and a normalization factor.In Graph of Words we represent documents as a graph comprising of words, reduce it to retain the core points of the graph as keywords. It is based on node centrality and local components like the number of neighbors, indegree and outdegree of a node and the global component which is the inverse of the average shortest path distance from the node to any other node in the graph are used to calculate the term weight.

In Semantical term weighting the terms are weighted based on its association with other words.This may include count based methods like LSA[7],LDA[8] and neural predictive models such as word2vec[9],GloVe[10]. The count of the number of times terms occur together are used to form the co-occurrence matrix[11].

Latent Semantic Analysis/Indexing was one of the earlier methods developed to incorporate contextual information in the representation. It is a technique in distributional semantics and mainly based on the assumption that the words that are close in meaning will appear in similar context.LSA was the precursor to the modern day 'Topic Modeling'. A term-document matrix is constructed with the rows being unique terms and the columns corresponding to the documents.[7]The values in the matrix is the frequency with which a word appears with respect to the document. A slight variation to this would be utilizing a tf-idf transformation, which roughly corresponds to the frequency of a term within a document divided by its frequency in the entire corpus.[Link]

LDA is an an example of probablistic topic modeling.[8]It uses two matrices , a topic per document matrix and a word per topic matrix .The values of these matrices are probabilities of choosing a particular word/document on a particular topic.HDP(HierarchialDirichlet Process)-LDA is an extension that is used to address the case when the number of topics are not known apriori.So a base distribution is chosen that accounts for countably infinite set of possible topics and the finite set of topics distribution is sampled from this base distribution[12].

Probabilistic Latent Semantic Indexing is an approach to automated document indexing which is based on a statistical latent class model for factor analysis of count data.[13] Probabilistic Latent Semantic Analysis (PLSA) was originally proposed to provide a probabilistic method for the analysis of co-occurrence data.

PMI is a common measure for the strength of association between two words. It is defined as the log ratio between the joint probability of two words w and c and the product of their marginal probabilities.[14]

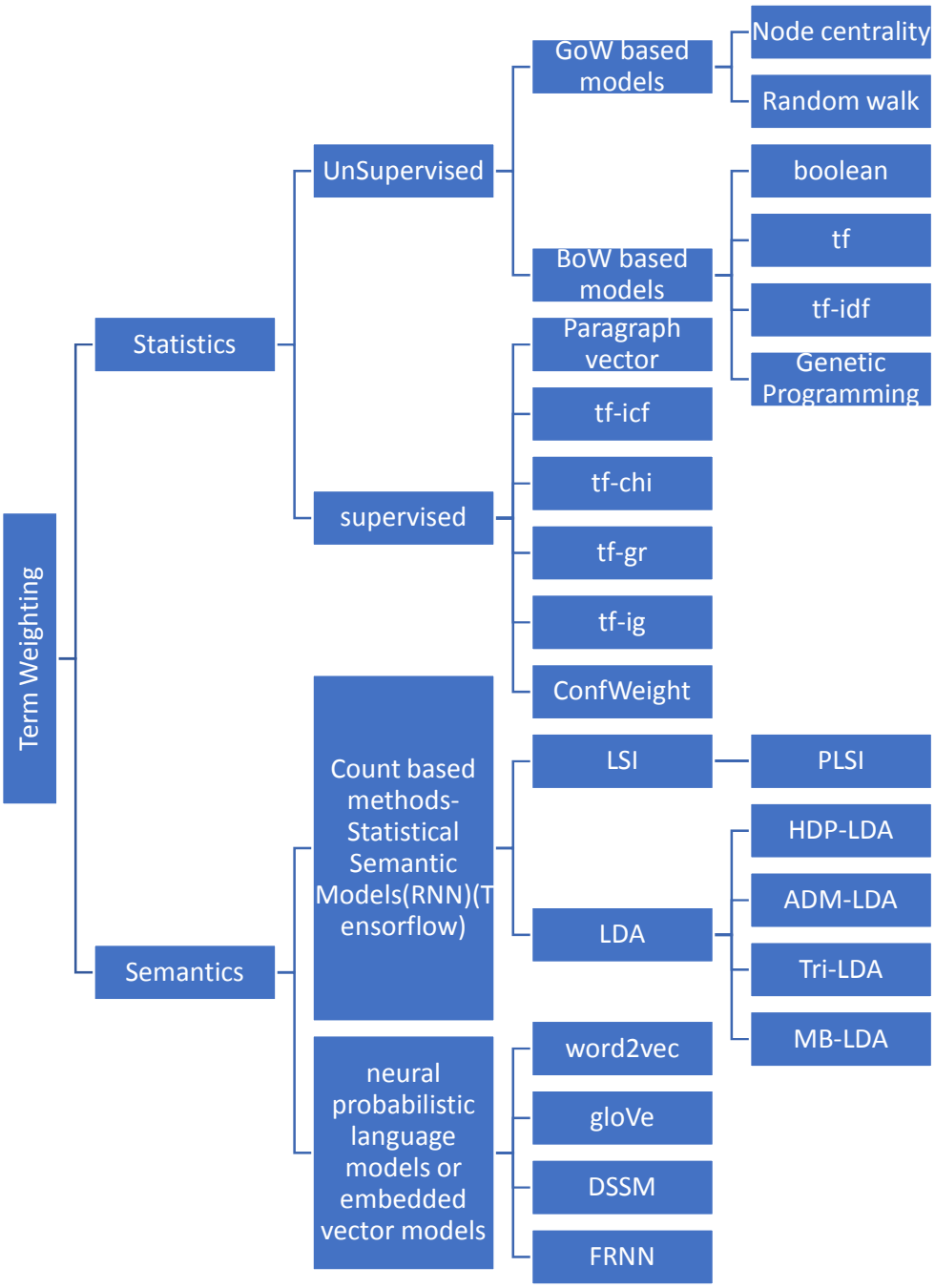

Fig. 1: Classification of term weighting schemes

7 Neural Probablistic Language Models

Neural networks based word embeddings was first introduced by Bengioet. Al. It has become popular after Mikolov et al's Word2Vec because it reduced the complexity of finding the word embedding by not involving external annotations.

Mikolov et.al. have proposed two model architectures for learning distributed representation of words by means of minimizing computational complexity. The main observation in other earlier neural network based embedding techniques is that most of the complexity is caused by the hidden layer in the model which is non-linear[9].This proposed model termed as word2vec is a predictive model because it can either predict a word's context or given the context words it can predict the target word from raw text. The former functionality is described through the skip gram model and the latter functionality through the continuous bag of words model architecture.

The Continuous bag-of-words(CBOW) architecture is called so because like the traditional bag of words architecture it does not take into account the history of the words and also uses continuous distributed representation of the context words to predict the target word. Since a log linear classifier is used ,the model complexity is much lesser than the feed forward neural network model.

The Skip Gram model uses current word as the input and uses log linear classifier to predict words of a specified range before and after the current word(context words).By increasing the range the quality of the resulting word vectors can be increased but by compromising the complexity of the model. One advantage that neural networks provide for word vectors is that it could encode many linguistic patterns and irregularities explicitly.

An extension to the skip gram model with an aim to improve the quality of the vectors and its training speed was proposed by Mikolov et. al [15].The training objective of skipgram model would be to extract word representations that aid in determining the surrounding or context words in a sentence or document. The training advantage that skip gram model has over all other neural based word embedding techniques is that it does not involve complex matrix multiplications. The authors have showed that by sub-sampling frequent words during the training phase results in significant speedup and also this improves the accuracy of representing less frequent words.

The skipgram model aims to increase the average log probability,

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}\,|w_t) \quad (2)$$

where c is the size of the context window, w_t being the target or center word.This is achieved by using a softmax function,

$$p(w_o|w_i) = \frac{\exp\,(v'^{T}_{w_O} v_{wI})}{\sum_{w=1}^{W}\exp\,(v'^{T}_{w} v_{wI})} \quad (3)$$

where W is the total number of words in the vocabulary,$v_w$and $v'_{w\ are}$ the input and output vector representations of any word w.The cost of computing using this formulation amounts proportional to W. A computationally efficient approximation to this formulation was given by Morin and Bengio .The advantage of this method is that only $\log_2(W)$ output nodes need to be evaluated[15].This method termed the hierarchical softmax uses a binary tree like representation with W nodes as leaves and each non-leaf node represents the relative probabilities of its child nodes. The probabilities are assigned to words through random walk. The standard softmax function uses two representations $v_w$and v'_w to each word w, the

hierarchical softmax method assigns one representation $v_w$for each word and one representation v'_n for every non-leaf node.

An extension to the hierarchical softmax is the Noise Contrastive Estimation(NCE).NCE suggests that a good model must be able to distinguish data(target words) from noise through logistic regression. A simplified version of NCE is the negative sampling technique which aims to give high probabilities to real words and lower probabilities to noise. This is computationally efficient because the loss function can be computed only for the number of noise words selected and not for the entire vocabulary.

Another technique suggested was the sub sampling of frequent words,where each word w_i in the vocabulary can be discarded with a probability,

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \qquad (4)$$

where $f(w_i)$ is the frequency of the word and t is the threshold. This subsamples words with frequency greater than t while preserving the rankings of the frequencies, mainly done to counter the imbalance between rare and frequent words.

Pennington et.al. have suggested an idea that combines the ideologies presented by two architectures: global matrix factorization and local context window, resulting in a new global log bilinear regression model called GloVe[10]. The local context window methods like skip gram and CBOW, fail to incorporate global statistical information as a result of which repetition and patterns aren't learned well by these models.The authors discovered that instead of the model using raw co-occurrence probabilities,better discriminative capabilities can be achieved using ratios of these co-occurrence probabilities. The co-occurrence probability of any word "i" with an arbitrary word "j" that appears in the context of "i" is given by,

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}} \qquad (5)$$

where X_{ij} is the number of times word j appears in the context of word i, X_i is the number of times any word appears in the context of word i and it can be represented as the sum of all words "k" that appear in the context of "i".Words that don't aid in better discrimination of words "i" and "j" are referred to as noise and co-occurrence probabilities helps to filter out the noise points. Let the function that the model is trying to learn be given by F, which represents the mapping from a space to compare two words with a context word to a space of co-occurrence probability ratios.

$$F\left(w_i, w_j, \widetilde{w_k}\right) = \frac{P_{ik}}{P_{jk}} \qquad (6)$$

w's are real valued vectors, and to incorporate the ratio between two words the authors propose using the vector differences as input resulting in the following equation:

$$F\left(w_i - w_j, \widetilde{w_k}\right) = \frac{P_{ik}}{P_{jk}} \qquad (7)$$

To preserve the linear structure ,the vector input function is converted into a scalar by taking the dot product of the difference vectors and the context vector.

$$F\left((w_i - w_j)^T . \widetilde{w_k}\right) = \frac{P_{ik}}{P_{jk}} (8)$$

The authors propose to discriminate ordinary words and context words by inducing the function F to be homomorphic from the additive group of real numbers to the multiplicative group of positive real numbers resulting in the below equation:

$$F\left((w_i - w_j)^T . \widetilde{w_k}\right) = \frac{F(w_i{}^T . \widetilde{w_k})}{F\left(w_j{}^T . \widetilde{w_k}\right)} \quad (9)$$

Equating this with the above equation results in,

$$\frac{P_{ik}}{P_{jk}} = \frac{F(w_i{}^T . \widetilde{w_k})}{F\left(w_j{}^T . \widetilde{w_k}\right)} \quad (10)$$

$$F(w_i{}^T . \widetilde{w_k}) = P_{ik} = \frac{X_{ik}}{X_i} \quad (11)$$

The word-word co-occurrence matrix makes arbitrary distinction between a word and a context word and to encode the same in the model ,exchange symmetry ought to be maintained let F be an exponential function .

$$(w_i{}^T . \widetilde{w_k}) = \log P_{ik}$$
$$= \log\left(\frac{X_{ik}}{X_i}\right) = \log X_{ik} - \log X_i \quad (12)$$

Since the [2] term is independent of context words choice, this is replaced with bias b_i and introducing a bias for the context word k, restores symmetry as,

$$w_i{}^T . \widetilde{w_k} + b_i + \widetilde{b_k} = \log X_{ik} \quad (13)$$

The authors identify a major drawback to this model as it weighs all the co-occurrences equally and propose a new weighted least squares regression model, with f being the weighting function.

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i{}^T . \widetilde{w_k} + b_i + \widetilde{b_k} - \log X_{ik}\right)^2 \quad (14)$$

The authors have concluded that the two classes of methods, predictive and count based methods are not fundamentally different as both the models use the underlying co-occurrence statistics but the count based methods more efficiently capture the global statistics.

William et. al. have tried to quantify the semantic changes over time using embeddings by comparing popular approaches like PPMI,SVD,word2vec on novel benchmarks and based on the analysis proposed two statistical laws relating frequency and polysemy to semantic change.Word Embeddingshave shown possibilities to be used as a diachronic tool. [16] William et al. proposed two quantitative laws of semantic change: (1)Law of Conformity which states that the rate of semantic change scales with an inverse power-law of word frequency, (2) Law of innovation states that, independent of frequency,words that are polysemous have higher rates of semantic change.

In Positive Pointwise Mutual Information(PPMI) the word embedding contains the PPMI values between a word vector (wi)and its context words. [14]The word vectors correspond to the rows of the PPMI matrix.SVDembeddings are typically a lower dimensional approximation of the PPMI embeddings.

SVD is one of the most popular methods for dimensionality reduction and it was initially used in latent semantic analysis (LSA). SVD factories the word-context co-occurrence matrix into the product of three matrices: U,Σ , VT where U and V are orthonormal matrices and Σ is a diagonal matrix of eigen values in decreasing order[17].SVD is often used to factorize the matrix produced by PPMI. Generally, only the top d elements of Σ are kept and are typically used as the word and context vectors respectively.

Yang et al. have proposed Topical word embedding model to enhance the discriminativeness by using topic models to assign topic for each word in the corpus and learn embedding based on both the words and the topics. Multi prototype vector models cluster context of a into groups and then generate a distinct vector for each cluster. [18]. The basic

intuition is to allow a word to have different embeddings under different topics.Three topical models have been proposed by the authors.TWE-1, the first model considers each topic as a pseudo word and learns topic embeddings and word embeddings separately and simultaneously and a topical word embedding is built according to the embeddings of the words and the topics.In TWE-2 model, each topic-word pair is considered a pseudo word and the embeddings are then learnt directly.TWE-3 reserves distinct embeddings for each word and each topic.The embedding for each word-topic is built by concatenating the corresponding word and topic embeddings.

8 Experimental Setup

The efficiency of the word embeddings obtained from word2vec ,gloveand LSA models were analyzed by performing a sentiment analysis task on the same. Sentiment analysis is the process of determining the emotion/sentiment over a series of words to aid in better understanding of the polarity of the context.Traditionally,one hot ending is used to represent words which results in sparse vectors and also the semantics of the words aren't considered resulting in different representation of similar words.

Word embeddings overcome the aforementioned issues as they are denser word representations and the semantics of the words are also preserved thus aiding in better insights. The dataset used for evaluation is the 'Sentiment140', created by Stanford University, which comprises tweets collected from various sources and annotated with a polarity.The dataset comprises of the following fields, polarity of the tweet,id,date,query,username and the text. The first task however is to have a document-level(tweets) representation from each of the individual word representations. Since we are dealing with a binary classification task(positive/negative polarity), the evaluation metric to be used for comparing the model efficiency is the accuracy. Accuracy can be defined as the number of times the model predicted a positive class over the dataset.

The first technique would be to average the individual word vectors associated with the tweet, summing it up and dividing by the frequency of the individual words. This preserves the dimension of the word-vectors when mapped to document-vectors. The second method is to sum the word vectors without averaging them.

A logistic regression model was used to evaluate the accuracy with which the tweets were correctly predicted. GloVe pre-trained vectors for tweets was used. The training and validation sets for the two techniques were created separately and logistic regression model was applied and the following accuracies were observed on the two techniques mentioned above, 74.30% and 74.70%. Word2Vec model was trained with the dataset and resulting embeddings were used with a logistic regression model for the two techniques (average and sum).The following accuracies were observed,74.44% and 74.78%.

In an aim to improve the accuracies a neural network model was used for classifying the tweets that yielded an accuracy score of 80.56% with the summing technique. Therefore by using sophisticated neural network architectures the accuracy of the sentiment analysis model can further be enhanced. The results favor word2vec as the better predictive model for the sentiment analysis task.

LSA based word vectors for the same dataset using unigram, bigram and trigram over a maximum of 30000 features and applying the save over a logistic regression model for sentiment analysis yielded an accuracy of 85.44%

LSA yields better results compared to the predictive models GloVe and word2vec, however these results are specific to the given dataset and the training parameters used.

9 Conclusion

In this survey,the various techniques available for word representations were discussed briefly and also the various components and metrics used were also discussed. The trend in word embeddings is moving towards predictive neural based models which give denser word representation as opposed to the count-based models providing sparser representations. Also a sample experiment was performed to compare the word2vec and glove models through a sentiment analysis task. The lack of standard evaluation metrics for the word embeddings is a definite problem in this field. However several new models are being developed with an aim of incorporating multi-lingual and multi-sense word embeddings for use in NLP tasks.

References

[1] L. Ratinov and J. Turian, "Word representations : A simple and general method for semi-supervised learning,", pp. 384–394, 2010.

[2] M. Abdel, "Neurocomputing New term weighting schemes with combination of multiple classi fi ers for sentiment analysis," Neurocomputing, vol. 167, pp. 434–442, 2015.

[3] H. Wu, "Reducing Over-Weighting in Supervised Term Weighting for Sentiment Analysis," pp. 1322–1330, 2014.

[4] [4] S. T. Piantadosi, "Zipf ' s word frequency law in natural language : a critical review and future directions," pp. 1–24, 2015.

[5] G. Paltoglou and M. Thelwall, "A study of Information Retrieval weighting schemes for sentiment analysis," pp. 1386–1395, 2010.

[6] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," vol. 60, 2004.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," J. Am. Soc. Inf. Sci., vol. 41, pp. 391–407, 1990.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," vol. 3, pp. 993–1022, 2003.

[9] Mikolov, T., Sutskever, S., Chen, K., Corrado, G., and Dean, J.,"Efficient Estimation of Word Representations in Vector Space",ICLR workshop,2013.

[10] J. Pennington, R. Socher, and C. D. Manning, "GloVe : Global Vectors for Word Representation."

[11] Baroni M., Dinu G., & Kruszewski, G.," Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors", ACL, 238–247,2014.

[12] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," pp. 1–30, 2005.

[13] T. Hofmann, "Probabilistic Latent Semantic Analysis".

[14] O. Levy and Y. Goldberg, "Linguistic Regularities in Sparse and Explicit Word Representations," Proc. Eighteenth Conf. Comput. Nat. Lang. Learn., pp. 171–180, 2014.

[15] Mikolov, T., Sutskever, S., Chen, K., Corrado, G., and Dean, J., "Distributed representations of words and phrases and their compositionality", NIPS, pages 3111–3119,2013.

[16] William L.Hamilton,Jure Leskovec,Dan Jurafsky,"Diachronic word embeddings reveal statistical laws of semantic change",Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,pages 1489-1501,August 7-12,2016.

[17] H. Series and L. Algebra, "Singular Value Decomposition and Least Squares Solutions ," vol. 420, pp. 403–420.

[18] Yang Liu, Zhiyan Liu, Tat-Seng Chua,Maosong Sun,"Topical Word Embeddings", Association for the Advancement of Artificial Intelligence,2015.