Words are not Equal: Graded Weighting Model for building Composite Document Vectors

Pranjal Singh
B.Tech.-M.Tech. Dual Degree
Computer Science & Engineering
Indian Institute of Technology Kanpur
pranjals16@gmail.com

Amitabha Mukerjee
Professor
Computer Science & Engineering
Indian Institute of Technology Kanpur
amit@cse.iitk.ac.in

Abstract

Despite the success of distributional semantics, composing phrases from word vectors remains an important challenge. Several methods have been tried for benchmark tasks such as sentiment classification, including word vector averaging, matrix-vector approaches based on parsing, and on-the-fly learning of paragraph vectors. Most models usually omit stop words from the composition. Instead of such an yes-no decision, we consider several graded schemes where words are weighted according to their discriminatory relevance with respect to its use in the document (e.g., idf). Some of these methods (particularly tf-idf) are seen to result in a significant improvement in performance over prior state of the art. Further, combining such approaches into an ensemble based on alternate classifiers such as the RNN model, results in an 1.6% performance improvement on the standard IMDB movie review dataset, and a 7.01% improvement on Amazon product reviews. Since these are language free models and can be obtained in an unsupervised manner, they are of interest also for under-resourced languages such as Hindi as well and many more languages. We demonstrate the language free aspects by showing a gain of 12% for two review datasets over earlier results, and also release a new larger dataset for future testing (Singh, 2015).

1 Introduction

Language representation is a very crucial aspect to perform various NLP tasks and has been looked into great detail in recent times. Language representation models have fallen into broadly two categories: ones which require hand-trained language databases such as treebanks (e.g., (Socher et al., 2013)) and ones which are language agnostic and work on raw corpora (e.g., LDA, BOW, Skip-Gram, NLM, etc.). Liu (2015) compare various language agnostic models for topic modeling.

Language independent models such as LDA and BOW have been quite effective since long time. Variants of BOW such as tf-idf had changed the perception of researchers towards these models when they were proved effective in various NLP tasks. LDA was able to model inter and intra documental statistical and relational structure quite well overcoming the drawbacks of BOW. But the semantic and syntactical dependencies were still ignored. After the introduction of neural language vector models, NLP saw a huge diversion in representation of words and documents. For indi-

Method	IMDB	Amazon	Hindi
RNNLM (Baseline)	86.45	90.03	78.84
Paragraph Vector (Le and Mikolov, 2014)	92.58	91.30	74.57
Averaged Vector	88.42	88.52	79.62
Weighted Average Vector	89.56	88.63	85.90
Composite Document Vector	93.91	92.17	90.30

Table 1: Comparison of accuracies on 3 Datasets (IMDB, Amazon Electronics Review and Hindi Movie Reviews (IITB)) for various types of document composition models. The state of the art for these tasks are: IMDB: 92.58% (Le and Mikolov, 2014); Amazon:85.90% (Dredze et al., 2008), Hindi:79.0% (Bakliwal et al., 2012).

D S Sharma, R Sangal and E Sherly. Proc. of the 12th Intl. Conference on Natural Language Processing, pages 11–19,
Trivandrum, India. December 2015.

vidual words, vectors are obtained via distributional learning, the mechanisms for which varies from document-term matrix factorization (Landauer and Dumais, 1997), various forms of deep learning (Collobert et al., 2011; Turian et al., 2010; Socher et al., 2013), optimizing models to explain co-occurrence constraints (Mikolov et al., 2013b; Pennington et al., 2014), etc. Once the word vectors have been assigned, similarity between words can be captured via cosine distances. The same models have been extended ((Le and Mikolov, 2014)) with new variables to build vector models for sentences and documents. These models include the essence of individual words as well as their relative order in terms of sentence vector which was earlier absent in word vectors. The advantage of these approaches is that they can capture both the syntactic and the semantic similarity between words/documents in terms of their projections onto a high-dimensional vector space; further, it seems that one can tune the privileging of syntax over semantics by using local as opposed to large contexts (Huang et al., 2012).

Some grammarians have been trying to find whether sentence meaning accrues by combining word meanings, or whether words gain their meanings based on the context they appear in (Matilal, 2001). (Turney et al., 2010) give a detailed overview of various vector space models and their composition. A surprising event in Information Theory has higher information content than an expected event (Shanon, 1948). The same happens when we give weights to word vectors. We give more weight to events which evoke surprise and less weight to events which are expected. The most popular weighting concept in this domain is the idea of tf-idf which we have utilized in this work (Refer Table 1).

In this work, we focus on a graded approach to assessing the importance of each word in a compositional models. Graded models such as tf-idf have long been used in NLP, but they do not seem to have been used in word vector composition tasks yet. The intuition is that in discrete cutoff functions, while simple, raise questions regarding threshold (what constitutes a stop word), and do not degrade performance gradually (Fig. 1).

We claim that the document vectors in hand is a much better representation of each document than doing it separately. This can be justified by the fact that we now incorporate contribution of each word as per its importance as well as well as that of document without ignoring tf-idf representation which performs considerably well in tasks such as retrieval. Hence, we call this as composite document vector representation. We then go a step ahead to build an ensemble of our model and recurrent neural network, which essentially has the properties of a generative model, to achieve state-of-the-art result on IMDB movie review dataset (94.19%) and on Amazon electronics reviews dataset with a significant improvement over previous best. The world class results in English clearly indicate the efficacy of our approach and improvements in Hindi depict the deficiency in other models which were used earlier.

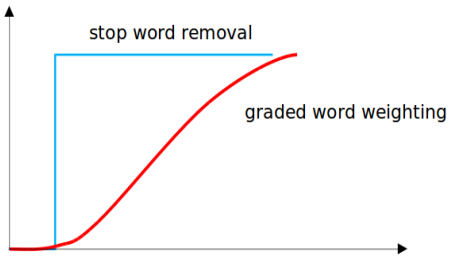

Figure 1: Intuition behind the proposed approach is that graded mechanisms for word combination may do better than discrete models that simply reject stop words and treat all other words equally(e.g. averaging methods).

2 Related Work

2.1 Neural Language Model

Language modeling problem involved using frequency counts of n-grams for so many years but it ignores a large number of n-grams which are not seen while training leading to data sparsity and over-fitting of training data. Also these n-gram models along with BOW suffers from the curse of dimensionality. Neural Networks tend to overcome the drawbacks of n-gram models because they can model continuous variables or distributed representation, which is a necessity if we would like to find better generalizations over the highly discrete word sequences (Bengio et al., 2003). Neural language models were introduced by Bengio et al., 2001 (revised in 2003(Bengio et al., 2003)). They build a mapping C from each word i of the vocabulary V to a feature vector $C(i) \in \mathbb{R}^m$, m is the number of features; a probability function g over words expressed with

C; and finally learn the word vector and parameters of probability function. Morin(2005) proposed a hierarchical model to speed up the training cost by clustering similar words before computing their probability in order to only have to do one computation per word cluster at the output layer of the NN. Le(2011) combined neural networks with n-gram language models in a unified approach. They cluster words to structure the output vocabulary. Mikolov(2010) achieved the best reduction in perplexity by using recurrent neural network which uses the the current input as well as the output of the previous iteration. Mikolov(2011) present several modifications of the original recurrent neural network language model (RNN LM). The present approaches that lead to more than 15 times speedup for both training and testing phases. Collobert(2011) show the use of semi-supervised learning using deep neural networks to perform at the state-of-the-art of various NLP tasks. Wang(2014) propose a word vector neural-network model, which takes both sentiment and semantic information into account. This word vector expression model learns word semantics and sentiment at the same time as well as fuses unsupervised contextual information and sentence level supervised labels. Neelakantan(2014) took word vector models to next level where they proposed multiple embeddings per word. The problem that still remains in hand is that either these models are computationally expensive or they have failed to generalize properly. We, therefore, adopt skipgram model (Mikolov et al., 2013b), details about which have been discussed in next section, because deep network model of Collobert et al.(2008) takes too much time for training (skipgram reduces computational complexity from O(V) to O(log V) (Morin and Bengio, 2005)).

2.2 Sentiment Analysis

Majority of the existing work in this field is in English (Pang and Lee, 2008). Medagoda(2013) surveys sentiment analysis in non-English languages while (Sharma et al., 2014) give a summary of work done in Hindi in the field of opinion mining. There have been heuristic based and machine learning based models used in this domain. Heuristic based methods, in general, classify text sentiments on the basis of total number of derived positive or negative sentiment oriented features. But these models rely heavily on human engineered features which, in general, is a domain and language dependent task. Several groups have attempted to improve the situation by modeling the composition of words into larger contexts (Le and Mikolov, 2014; Socher et al., 2013; Johnson and Zhang, 2014; Baroni et al., 2014).

Pang(2004) achieved an accuracy of 87.2% (Pang et al. 2004) on a dataset that discarded objective sentences and used text categorization techniques on the subjective sentences. Le(2014) use paragraph vector model and obtain 92.6% accuracy on IMDB movie review dataset. More difficult challenges involve short texts with non-standard vocabularies,as in twitter. Here, some authors focus on building extensive feature sets (e.g. Mohammad et al.(2013); F-score 89.14).

However, most of the work on sentiment analysis in Hindi has not attempted to form richer compositional analyses. For the type of corpora used here, the best results, obtained by combining a sentiment lexicon with hand-crafted rules (e.g. modeling negation and "but" phrases), reach an accuracy of 80% (Mittal et al., 2013). Joshi(2010) compared three approaches: In-language sentiment analysis, Machine Translation and Resource Based Sentiment Analysis. By using WordNet linking, words in English SentiWordNet were replaced by equivalent Hindi words to get H-SWN. The final accuracy achieved by them is 78.1%. Bakliwal(2012) traversed the WordNet ontology to antonyms and synonyms to identify polarity shifts in the word space. Further improvements were achieved by using a partial stemmer (there is no good stemmer / morphological analyzer for Hindi), and focusing on adjective/adverbs (seed words given to the system); their final accuracy was 79.0% for the product review dataset. Mukherjee et al. (2012) presented the inclusion of discourse markers in a bag-of-words model and how it improved the sentiment classification accuracy by 2-4%.

Many approaches seek to improve their performance by combining POS-tags and even parse tree structures into the models for higher accuracies in specific tasks (Socher et al., 2013). One problem in this approach is that of combining the word vectors to build document vectors because of issues in merging parse trees. Also these models are language dependent and computationally very expensive.

3 Method

The algorithms and data structures used in this thesis have been introduced and discussed below.

3.1 Distributed Representation

Mikolov et al. (2013b) proposed two neural network models for building word vectors from large unlabeled corpora; Continuous Bag of Words(CBOW) and Skip-Gram. In the CBOW model, the context is the input, and one tries to learn a vector for the central word; in Skip grams, the input is the target word and one tries to guess the set of contexts. We have adopted skipgram model to build vector representations for words as it performs better with larger vocabulary.

Each current word acts as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word. The objective is to maximize the probability of the context given a word within a language model:

$$p(c|w;\theta) = \frac{\exp^{v_c \cdot v_w}}{\sum_{c' \in C} \exp^{v_c \cdot v_w}}$$

where v_c and $v_w \in R^d$ are vector representations for context c and word w respectively. C is the set of all available contexts. The parameters θ are v_{c_i}, v_{w_i} for $w \in V$, $c \in C$, $i \in 1, ..., d$ (a total of $|C| \times |V| \times d$ parameters).

This distributed representation of sentences and documents (Le and Mikolov, 2014) modifies word2vec (Skip-Gram) algorithm to unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents. The algorithm represents each document by a dense vector which is later trained and tuned to predict words in the document. In this framework, every paragraph is mapped to a unique vector and id, represented by a matrix D, which is a column matrix. Every word is mapped to a unique vector and word vectors are concatenated or averaged to predict the context, i.e., the next word.

The paragraph vector is shared across all contexts generated from the same paragraph but not across paragraphs. The word vector matrix W, however, is shared across paragraphs. i.e., the vector for "good" is the same for all paragraphs. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph. The advantage of using paragraph vectors is that they inherit the property of word vectors, i.e., the semantics of the words. In addition, they also take into consideration a small context around each word which is in close resemblance to the n-gram model with a large n. This property is crucial because the n-gram model preserves a lot of information of the sentence/paragraph, which includes the word order also. This model also performs better than the Bag-of-Words model which would create a very high-dimensional representation that has very poor generalization.

Our model incorporates property of document vector as well as property of word vectors to build an enhanced representation of documents without ignoring the properties of tf-idf representation.

3.2 Semantic Composition

The Principle of Compositionality is that meaning of a complex expression is determined by the meaning of its parts or constituents and the rules which guide this combination. It is also known as *Frege's Principle*. In our case, the constituents are word vectors and the expression in hand is the sentence/document vector. For example,

The movie is funny and the screenplay is good

Composition	Accuracy
Multiplication	50.30
Average	88.42
Idf Graded Weighted Average	**89.56**

Table 2: Results of Vector Composition with different Operations

Analyzing the results from Table 2, we observed that when we deal with large number of features, there is a presence of large number of *zeros* and presence of a single zero in a feature will make that features contribution zero in the final vector, which happens in our case and thus multiplicative composition fails.

We, therefore, adopt both simple and idf weighted average methods in our work. The advantage with addition is that, it doesnot increase the dimension of the vector and captures high level semantics with ease. In fact, (Zou et al., 2013) have used simple average to construct phrase vectors which they have later used to find phrase level similarity using cosine distance.

(Mikolov et al., 2013c) showed that relations between words are reflected to a large extent in the

offsets between their vector embeddings. They also use additive composition to reflect semantic dependencies.

queen - king $\approx$ *woman - man*

(Blacoe and Lapata, 2012) clearly show that vectors of Neural Language Model and Distributed Model when used with additive composition outperform those with multiplicative composition in Paraphrase Classification task. DM vectors outperform by nearly giving accuracy difference of 6%. They also perform very well on Phrase similarity tasks.

We, therefore, propose graded weighting schema for better composition of vectors which is described below.

3.2.1 Graded Weighting

We describe two approaches to incorporate graded weighting into word vectors for building document vectors. Let v_{w_i} be the vector representation of the i^{th} word. Then document vector v_{d_i} for i^{th} document is:

$$v_{d_i} = \begin{cases} 0 & w_k \in stopwords \\ \sum\limits_{w_k \in d_i} v_{w_k} & w_k \notin stopwords \end{cases}$$

The above equation is 0-1 step-function which ignores contribution of all stop words. Now we propose another schema which weighs the contribution of each word while building document vector with a graded approach. We define $idf(t,d) = \log(\frac{|D|}{df(t)})$ where t is the term, d is the document and other notations are same as in previous subsection. The new document vector representation considering this graded schema is:

$$v_{d_i} = \begin{cases} 0 & idf(w_k, d_i) \leq \delta \\ \sum\limits_{w_k \in d_i} idf(w_k, d_i).v_{w_k} & otherwise \end{cases}$$

where δ is a pre-defined threshold below which the word has no importance and above which the *idf* terms gives importance to that particular word.

Till date, everyone has ignored how to effectively use vector composition techniques and as a result, this area has seen very less attention. But we have successfully used *idf* values to give weights to word vectors and hence obtain much better sentence/document vectors. The advantage of this model is that once we obtain *idf* values from training corpus, we can directly use it with test corpus without any additional computation. The results (see 4.3) obtained by using this technique clearly demonstrate how effective it is for tasks such as sentiment analysis.

3.3 Composite Representation

This experiment redefined document representation in NLP used for sentiment classification. It has the property of including both syntactic and semantic properties of a piece of text. The limitations of skip-gram word vectors have been fulfilled by document vectors and hence we achieve state-of-the-art results on IMDB movie review dataset as well as amazon electronics review dataset.

We first generated n-dimensional word vectors by training skip-gram model on the datasets. We then assigned weights to word vectors for each document to create document vectors. This now acts as a feature set for that particular document. We then created *tf-idf* vectors for each document. This can be seen as a vector representation of that particular document. We then concatenated these document vectors with document vectors obtained after training the desired dataset separately with the model proposed in (Le and Mikolov, 2014). Discrimination weighted vectors give a great boost to classification accuracies on various datasets and hence justifies our claim.

3.4 Dimensionality Reduction

Dimensionality Reduction is the process of reducing the number of random variables in such a way that the remaining variables effectively reproduce most of the variability of the dataset. The reason for using such techniques is because of the *curse of dimensionality* which is a phenomena that occurs in high-dimension but doesn't occur in low-dimension.

Table 3 summarizes how feature selection has improved classification accuracy on the 700 Movie review dataset. With ANOVA-F, we selected around 4k features but with PCA, this number was just 50. So, the low accuracy with PCA can be attributed to the fact that we may have lost some important features in low dimension. Also, PCA cannot work with size of dimension $d >$*size of learning set*. This sharp decrease in accuracy in both cases happens because ANOVA-F selects features with larger variance across group and thus reduces noise to a larger extent whereas PCA reduces angular variance which is not effective in this case due to the distribution of data points in high-dimensional space.

Method	Feature Selection	Accuracy
Document Vector + tfidf	None	74.57
	PCA(n=50)	76.33
	ANOVA-F	88.07
Weighted Word Vector + tfidf	None	76.43
	ANOVA-F	90.37
	PCA(n=50)	78.61

Table 3: Accuracies on our newly released 700-Movie Review Dataset

This newly released dataset is much larger than previous standard dataset and very less focused towards sentiment of the review.

4 Experiment

In this section, we describe the experiments and analyze the results.

4.1 Datasets

We have used 3 datasets for experiments in Hindi and 5 for English. All the datasets including our self created Hindi dataset are described below.
We experimented on two Hindi review datasets. One is the Product Review dataset (LTG, IIIT Hyderabad) containing 350 Positive reviews and 350 Negative reviews. The other is a Movie Review dataset (CFILT, IIT Bombay) containing 127 Positive reviews and 125 Negative reviews. Each review is around 1-2 sentences long and the sentences are mainly focused on sentiment, either positive or negative.
Our 700-Movie Review Corpus in Hindi contains movie reviews from websites such as *Dainik Jagran* and *Navbharat Times*. The movie reviews are longer than the previous corpus and contains subjects other than sentiment. There are in total 697 movie reviews from both the websites. The statistics compiled is described below.

For experiments in English, we trained on IMDB movie review dataset (Maas et al.(2013)) which consists of 25,000 positive and 25,000 negative reviews. It also contains an additional 50,000 unlabeled documents for unsupervised learning.

Positive Reviews	356
Negative Reviews	341
Total Reviews	697
29.7 sentences per document	
494.6 words per document	

Table 4: Statistics of Movie Reviews of the 700-Movie Reviews Dataset

The Trip Advisor Review dataset contains around 240K reviews (206MB) from hotel domain. Reviews with overall rating >=3 were annotated as positive and those with overall rating <3 were annotated as negative. The dataset was split into 80-20 ratio for training and testing purpose.

We also took amazon reviews for our experiments. Reviews with overall rating >=3 were annotated as positive and those with overall rating <3 were annotated as negative. The dataset was split into 80-20 ratio for training and testing purpose. There were 3 review datasets: Electronics dataset consists of 1,241,778 reviews, Watches Dataset consists of 68,356 reviews and MP3 Dataset consists of 31,000 reviews.

4.2 SkipGram or CBOW

We present an interesting experiment to demonstrate that skipgram indeed performs better than CBOW. SkipGram model tends to predict a context given a word whereas CBOW model predicts a word given a context. It seems intuitive and also from observation (Mikolov et al., 2013a) that SkipGram will perform better on semantic tasks and CBOW on syntactic tasks. We now try to evaluate how they differ on classification accuracies on the two datasets: *Watches* and *MP3*. Figure 2 show that skipgram outperforms CBOW on sentiment classification task. It can be justified by the fact that sentiment inclination of a document is more oriented towards semantics of that document rather than just syntax and our results clearly demonstrate this fact.

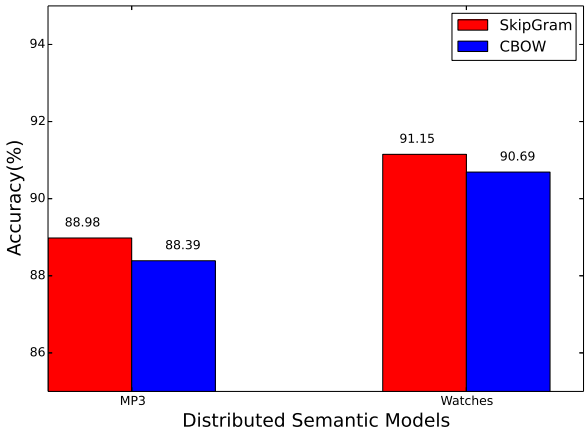

Figure 2: Variation of Accuracy with skipgram and cbow on Watches and MP3 Datasets.

4.3 Results

Method	Accuracy
Maas et al.(2011)	88.89
NBSVM-bi (Wang & Manning, 2012)	91.22
NBSVM-uni (Wang & Manning, 2012)	88.29
SVM-uni (Wang & Manning, 2012)	89.16
Paragraph Vector (Le and Mikolov(2014))	92.58
Weighted WordVector+Wiki(Our Method)	88.60
Weighted WordVector+TfIdf(Our Method)	90.67
Composite Document Vector	**93.91**

Table 5: Results on IMDB Movie Review Dataset

Table 5 summarizes the results obtained by others and by us on the IMDB movie review dataset. We have gone above the previous best (Le and Mikolov, 2014) by a margin of 1.33% using discrimination weighting. The main contributor for improvement in results is our new document vector which overcomes the weaknesses of BOW and document vectors taken separately.

Method	Weight	Accuracy(1)	Accuracy(2)
0-1 Weighting	0	93.84	93.06
	1	**93.91**	**93.18**
Graded idf Weighting	2	**93.89**	93.17
	2.5	93.87	93.16
	2.8	93.86	93.16
	3	93.86	**93.22**
	4	93.83	93.12
	5	93.75	93.03

Table 6: Results on IMDB Movie Reviews using Various Weighting Techniques(Composite Document Vector);Accuracy(2) is when we exclude tf-idf features

Tables 6 demonstrate the effectiveness of our proposed graded weighting technique. Without *tf-idf* features, our proposed method performs better in the graded idf weighting case and when we include *tf-idf* features, 0-1 weighting perform better than idf graded technique and both perform better than the previous state-of-the-art. We see that with larger weights there is a decrease in accuracy and that is because we are now filtering out more words which are important while building document vector. Table 7 is a further improvement in results once we incorporate predictions of RNNLM and Composite document vector model together(voting ensemble). Here, we first trained a *RNNLM* and then obtained predictions on test reviews in terms of probability. We trained Linear SVM classifier using new Document Vectors and then obtained predictions on test reviews. We then merged these two predictions using a voting based approach to obtain final classification.

Method	Accuracy
Composite Document Vector	93.91
Composite Document Vector + RNNLM (Our Method)	**94.19**

Table 7: Results on IMDB Movie Review Dataset

Table 8 presents result of experiment conducted on famous Amazon electronics review dataset (Leskovec and Krevl, 2014). Our vector averaging method alone has beaten previous best by 3.3%.

Features	Accuracy
(Dredze et al., 2008)	85.90
Max Entropy (Dredze et al., 2008)	83.79
WordVector Averaging(Our Method)	88.63
Composite Document Vector (Our Method)	92.17
Composite Document Vector + RNNLM	**92.91**

Table 8: Results on Amazon Electronics Review Dataset

Features	Accuracy(1)	Accuracy(2)
WordVector Averaging	78.0	79.62
WordVector+tf-idf	90.73	89.52
WordVector+tf-idf without stop words	91.14	89.97
Weighted WordVector	89.71	85.90
Weighted WordVector+tfidf	**92.89**	**90.30**

Table 9: Accuracies for Product Review and Movie Review Datasets.

Table 9 represents the results using five different techniques for feature set construction. We see that there is a slight improvement in accuracy on both datasets once we remove stop-words but the major breakthrough occurs once we used weighted averaging technique for construction of document vectors from word vectors.

Experiment	Features	Accuracy
In-language with SVM (Joshi et al., 2010)	tfidf	78.14
MT Based with SVM (Joshi et al., 2010)	tfidf	65.96
Improved HindiSWN (Bakliwal et al., 2012)	Adj. & Adv. presence	79.0
WordVector Averaging	word vector	78.0
Word Vector Averaging	word vector+tfidf	89.97
Weighted Word Vector with SVM (Our method)	tfidf+weighted word vector	**90.30**

Table 10: Comparison of Approaches: Movie Review Dataset

Experiment	Features	Accuracy
Subjective Lexicon (Bakliwal et al., 2012)	Simple Scoring	79.03
Hindi-SWN Baseline (Arora et al., 2013)	Adj. & Adv. presence	69.30
Word Vector with SVM	word vector+tfidf	91.14
Weighted Word Vector with SVM (Our method)	tfidf+weighted word vector	**92.89**

Table 11: Comparison of Approaches: Product Review Dataset

Table 10 and 11 compares our best method with other methods which have performed well using techniques such as tf-idf, subjective lexicon, etc.

5 Conclusion

In this work we present an early experiment on the possibilities of distributional semantic models (word vectors) for low-resource, highly inflected languages such as Hindi. What is interesting is that our word vector averaging method along with tf-idf results in improvements of accuracy compared to existing state-of-the art methods for sentiment analysis in Hindi (from 80.2% to 90.3% on IITB Movie Review Dataset). Also from Table 1, we can see that paragraph vector proposed by (Le and Mikolov, 2014) doesn't perform well owing to the fact that the Hindi dataset just contains single sentences highlighting the weakness of this model. The size of the corpus is also small to learn paragraph vectors. Thus, our model overcomes these weaknesses with a better document representation. We observe that pruning high-frequency stop words improves the accuracy by around 0.45%. This is most likely because such words tend to occur in most of the documents and don't contribute to sentiment. For example, the word फिल्म(Film) occurs in 139/252 documents in Movie Reviews(55.16%) and has little effect on sentiment. Similarly words such as सिद्धार्थ(Siddharth) occur in 2/252 documents in Movie Reviews(0.79%). These words don't provide much information.

We also see that when number of features accumulate to a large number than there are few redundant features creating noise in the representation of the text. We tried to reduce this noise by using feature variance techniques. The large increase in accuracy(around 11%) justifies our claim.

Before concluding, we return to the unexpectedly high improvement in accuracy achieved. One possibility we considered is that when the skip-grams are learned from the entire review corpus, it incorporates some knowledge of the test data. But this seems unlikely since the difference in including this vs not including it, is not too significant. The best explanation may be that the earlier methods, which were all in some sense based on a sentiWordnet, and at that one that was initially translated from English, were essentially very weak. This is also clear in an analysis from (Bakliwal et al., 2012), which shows intern-annotator agreement on sentiment words are very poor (70%) - i.e. about 30% of these words have poor human agreement. Compared to this, the word vector model provides considerable power underlining the claim that distributional semantics is a topic worth exploring for Indian languages.

Our experiments on new dataset and existing datasets show that our method is competitive with existing methods including state-of-the-art. This new concept of document vectors can overcome the weaknesses of existing models which were either deficient in capturing syntactic or semantic properties of text. These models failed to incorporate contribution of each word while we have tapped this area and hence achieved state-of-the art results. The ensemble of RNNLM and Composite Document Vector has beaten state-of-the-art by a significant margin and has opened this area for future research. These models have the advantage that they don't require parsing at any step neither do they require a lot of heavy pre-processing. These tasks require a lot of extra effort and they slow the progress a lot.

6 Future Work

Distributional semantics approaches remain relatively under-explored for Indian languages, and our results suggest that there may be substantial benefits to exploring these approaches for Indian languages. While this work has focused on sentiment classification, it may also improve a range of tasks from verbal analogy tests to ontology learning, as has been reported for other languages. For future work, we can explore various compositional models - a) weighted average - where weights are determined based on cosine distances in vector space; b) weighted multiplicative models. Identifying morphological variants would be another direction to explore for better accuracy. With regard to sentiment analysis, the idea of aspect-based models (or part-based sentiment analysis), which looks into constituents in a document and classify

their sentiment polarity separately, remains to be explored in Hindi.

In English, our Composite document vectors has led open a new area to look at where there can be many possible ensembles which may improve our work. Also, we could incorporate multiple word vectors here as well to distinguish between polysemous words. Another interesting and open area is to look at *Region of Importance* in NLP where we filter out sentiment oriented sentences and phrases from a unfocused corpus which contains text from various domains. The code and parameters are available at *github* for future research.

References

Akshat Bakliwal, Piyush Arora, and Vasudeva Varma. 2012. Hindi subjective lexicon: A lexical resource for hindi adjective polarity classification. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA).

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, 9.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. 2010. A fall-back strategy for sentiment analysis in hindi: a case study.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, J Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5524–5527. IEEE.

Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings.

B. K. Matilal. 2001. *The Word and The World India's contribution to the study of language*. Oxford Paperback, Delhi.

Nishantha Medagoda, Subana Shanmuganathan, and Jacqueline Whalley. 2013. A comparative analysis of opinion mining and sentiment classification in non-english languages. In *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*, pages 144–148. IEEE.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan H Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Namita Mittal, Basant Agarwal, Garvit Chouhan, Nitin Bania, and Prateek Pareek. 2013. Sentiment analysis of hindi review based on negation and discourse relation. In *proceedings of International Joint Conference on Natural Language Processing*, pages 45–50.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252. Citeseer.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of EMNLP*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Richa Sharma, Shweta Nigam, and Rekha Jain. 2014. Opinion mining in hindi language: A survey. *arXiv preprint arXiv:1404.4935*.

Pranjal Singh. 2015. Decompositional semantics for document embedding. `http://www.cse.iitk.ac.in/users/grounded-lang/spranjal/`.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu. 2014. Sentiment classification: The contribution of ensemble learning. *Decision support systems*, 57:77–93.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.