

Words are not Equal: A Graded Weighting Model for improving Additive Composition of Word Vectors

Pranjal Singh

Dept. of Computer Science & Engg.
IIT Kanpur
spranjal@iitk.ac.in

Amitabha Mukerjee

Dept. of Computer Science & Engg.
IIT Kanpur
amit@cse.iitk.ac.in

Abstract

In recent years, distributional semantics and neural language models for words and documents have been proposed to perform various NLP tasks ignoring earlier models such as tf-idf. Since both word vector model and document vector models are built separately, their combined efficacy as well as using word vectors to build document vectors have been ignored. We propose a graded weighting scheme for building document vectors which can be combined with document vectors trained separately as well as tf-idf representation used earlier to achieve state-of-the-art results in sentiment classification on classical IMDB movie review dataset achieving an accuracy of 94.19% (improvement of 1.61% over previous best). We also implement an ensemble model which boosts our accuracy by about 0.5% using recursive neural network. Since these are language free models and can be obtained in an unsupervised manner, they are of interest for under-resourced languages such as Hindi as well and many more languages. We demonstrate the language free aspects by developing a huge increment on results for Hindi with an impressive accuracy gain on state of the art by 12% (from 80%) for two review datasets.

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 are language dependent models (e.g., (Socher et al., 2013)) and ones which are language independent models (e.g., LDA, BOW, SkipGram, NLM, etc.).

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 individual 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 (Matić, 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 weights to events which occur by surprise and less weight to events which are expected. The most popular weighting concept in this domain is the idea of tf-idf which is explained later.

We develop on this idea and propose a graded weighting scheme to build document vectors from word vectors. This idea is merged with obtaining document vectors separately in the form of distributed learning as well as tf-idf. This combined model is language independent and is enriched with vital properties of other models. This ensures that our model is effective as well as accurate in various languages. 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%) 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.

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 and Bengio, 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 et al., 2011) combined neural networks with n-gram language models in a unified approach. They cluster words to structure the output vocabulary. (Mikolov et al., 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 et al., 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 et al., 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 et al., 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. (Nee-lakantan et al., 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 $O(\log V)$ to $O(V)$).

2.2 Sentiment Analysis

Majority of the existing work in this field is in English (Pang and Lee, 2008). (Medagoda et al., 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 and Lee, 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 and Mikolov (2014) use word vector models and obtain 92.6% accuracy on IMDB movie review dataset. More difficult challenges involve short texts with nonstandard 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 et al., 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 et al., 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. The Skip gram was found to perform better on smaller corpora, and here we have focused on this model for building our word vectors. The model uses each current word 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 $w \in V$, $c \in C$, $i \in 1, \dots, d$ (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 sentencedocument.

In Paragraph Vector 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.

3.2 tf-idf

Let $D = d_1, d_2, d_3, \dots, d_N$ be N documents under study and $T = t_1, t_2, t_3, \dots, t_M$ be the M unique terms in these documents, then each document can be represented as a M -dimensional vector:

$$t_d = \{tf_1, tf_2, tf_3, \dots, tf_M\}$$

$tf - idf$ weights discounts the frequency of terms based on their importance to a particular document in the entire document set collection under consideration. This is done as follows:

$$tfidf(d, t) = tf(d, t) \times \log\left(\frac{|D|}{df(t)}\right)$$

Here $df(t)$ is the number of documents in which term t appears and the other term is the *inverse document frequency* which we later use to give weights to each word.

3.3 Vector Averaging for phrases

As an output of the word vector learning, we now have a n -dimensional vector representation for each word in the Hindi corpus. Now we need to assign features for sentences and paragraphs taken from the sentiment dataset (training and test). Mikolov et al. (2013b) and Levy et al. (2014) show that many relational similarities can be recovered by means of vector arithmetic in the embedded space. Thus, additive models are useful, though others have claimed that multiplicative models correlate better with human judgments (Mitchell and Lapata, 2008; Socher et al., 2013). In this work, we have retained the simplicity of vector averaging to model larger chunks of discourse. This models the sentence/document in the same high dimensional space.

3.4 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

In the above sentence, consider the word vectors are represented by $w(x)$ and the sentence vector as $S(x)$.

Composition	Accuracy
Average	88.42
Weighted Average	88.41
Multiplication	50.30

Table 1: Results of Vector Composition with different Operations

Analyzing the results from Table 1, 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 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.

(Socher et al., 2013) also present yet another model for semantic composition but that uses a sentiment treebank which is a very costly structure to build and it is task dependent. For under-resourced languages such as Hindi, it would take

years to build (for English, task done through Amazon Mechanical Turk). This leads to appreciation of models such as Additive Composition. When an n-gram is given to the compositional models, it is parsed into a binary tree and each leaf node, corresponding to a word, is represented as a vector. Recursive neural models will then compute parent vectors in a bottom up fashion using different types of compositionality functions g . The parent vectors are again given as features to a classifier.

3.5 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.

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

Table 2: Accuracies on 700-Movie Review Dataset

Table 2 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 $d > \text{size of learning set}$.

This sharp decrease in accuracy in both cases: Document vector and Weighted Average word vector 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.

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.

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

Table 3: Statistics of Movie Reviews from Jagran and Navbharat

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.

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: Watches,

Electronics and MP3 each of size 30.8MB, 728.4MB and 27.7MB respectively. 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

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 1 show that

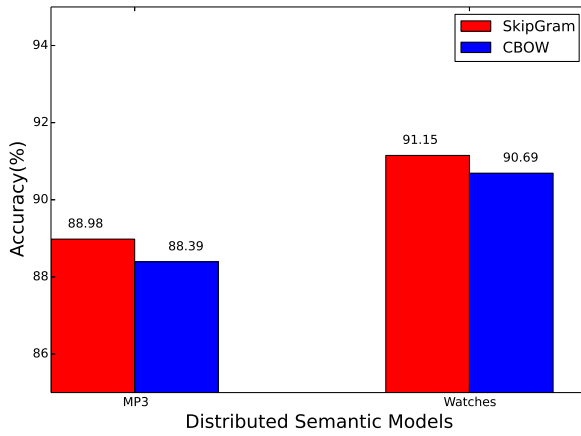


Figure 1: Variation of Accuracy with skipgram and cbow on Watches and MP3 Datasets.

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.

4.3 Graded Weighting

In this experiment, we first generated n -dimensional word vectors by training skip-gram model on product and movie review corpus in Hindi and IMDB corpus in English. We then averaged-out word vectors for each document to create document vectors. This now acts as a feature set for that particular document. We also 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 averaging-out word vector of each document.

Algorithm ?? presents a new approach to build document vectors using distributional semantics. 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. Refer to ?? to see how this new technique has improved sentiment classification on Hindi reviews. 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.

4.4 Enhanced Document Vector

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.

This experiment was done on IMDB movie review dataset. Here, we first trained a *recursive neural network* and then obtained predictions on test reviews in terms of probability. We trained another classifier (Linear SVM) using Enhanced Document Vectors and then obtained predictions on test reviews. We then merged these two predictions using a simple heuristic to obtain final classification. This led to a tremendous increase in classification accuracy (Refer section ??).

(van der Maaten and Hinton, 2008) represents embeddings

4.5 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
WordVector+Wiki(Our Method)	88.60
WordVector+TfIdf(Our Method)	89.03
Enhanced Document Vector	93.91

Table 4: Results on IMDB Movie Review Dataset

Table 4 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 enhanced document vector.

Method	Accuracy	Experiment	Features	Accuracy
WordVector Averaging+Tfidf+Document Vector	93.91	Subjective Lexicon (Bakliwal et al.(2012))	Simple Scoring	79.03
Enhanced Document Vector+RNNLM(Our Method)	94.19	Hindi-SWN Baseline (Arora et al.(2013))	Adjective and Adverb presence	69.30
		Word Vector with SVM (Our method)	tfidf with word vector	91.14
		Weighted Word Vector with SVM (Our method)	tfidf+weighted word vector	92.89

Table 5: Results on IMDB Movie Review Dataset

The main contributor for improvement in results is our enhanced document vector which overcomes the weaknesses of BOW and document vectors taken separately.

Table 5 is a further improvement in results once we incorporate predictions of RNNLM and enhanced document vector model together(voting ensemble).

Table 6 presents result of another experiment conducted on famous Amazon electronics review dataset(refer ??). Our vector averaging method alone has beaten previous best by 3.3% which was based on maximum entropy method.

Features	Accuracy
Dredze et al.(2008)	85.90
Max Entropy	83.79
WordVector Averaging(Our Method)	89.41
WordVector Averaging + Document Vector (Our Method)	92.17
WordVector Averaging + Document Vector + RNNLM (Our Method)	92.91

Table 6: 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 7: Accuracies for Product Review and Movie Review Datasets.

Table 7 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.

Table 8 and 9 compares our best method with various other methods which have performed well using techniques such as *tf-idf*, subjective lexicon, etc.

Table 8: Comparison of Approaches: Product Review Dataset

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

Table 9: Comparison of Approaches: Movie Review Dataset

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).

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. Similarly, words with very low frequency are noisy and can be pruned. For example, the word फिल्म occurs in 139/252 documents in Movie Review Dataset(55.16%) and has little effect on sentiment. Similarly words such as सिद्धार्थ occur in 2/252 documents in Movie Review Dataset(0.79%). These words don't provide much information.

We also observe that giving weights to word vectors while adding has resulted in a significant increase of accuracy. This seems much in accordance with expectation as pure addition ignores

the importance of word which is directly related to its ratio of frequency in whole corpus and its frequency in a particular document. We have successfully tapped this area which most of the works have ignored. We also see that when number of features accumulate to a large number than there are few redundant features present also. This brings a noise in the representation of the text in high dimension. We tried to reduce this noise by using feature variance technique called ANOVA-F and PCA and have succeeded as well. There is a large increase in accuracy(around 11%).

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, especially as amplified by the tf-idf process. Thus, this also seems to underline 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 concept of Enhanced Document Vectors can overcome the weaknesses of existing models which were either deficient in capturing syntactic or semantic properties of text. The ensemble of RNNLM and Enhanced 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. Another point to note is that we are re-computing the word vectors for the two review corpora, which are extremely small. We may expect better performance with a larger sentiment corpus. In English, our enhanced 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.

Acknowledgments

I would like to express my sincere gratitude towards my supervisor, Dr. Amitabha Mukerjee, for his constant support and encouragement. I am grateful for his patient guidance and advice in giving a proper direction to my efforts. I thank the Department of Computer Science and Engineering, IIT Kanpur, for providing the necessary infrastructure and a congenial environment for research work. I thank Tomas Mikolov for making the code of word2vec and sentence2vec publicly available.

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, Raffaella 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.
- 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 Bhat-tacharyya. 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.
- 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 Cernocký, 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.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *In Proceedings of ACL-08: HLT*, pages 236–244.
- 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. Cite-seer.
- 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*.
- 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.
- L.J.P. van der Maaten and G.E. Hinton. 2008. Visualizing high-dimensional data using t-sne.
- 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.