Word Vector Averaging: Parserless Approach to Sentiment Analysis

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 compositional semantics or word vector models have been proposed to capture both the syntactic and semantic similarity between words. Since these can be obtained in an unsupervised manner, they are of interest for under-resourced languages such as Hindi. We test the efficacy of such an approach for Hindi, first by a subjective overview which shows that a reasonable measure of word similarity seems to be captured in the model. We then apply it to the standard problem of sentiment analysis, for which several small database exist in Hindi. This requires some mechanism for dealing modeling larger strings given the vectors for its words, and several methods - additive, multiplicative, or tensor neural models, have been proposed. Here we find that the simplest - an additive average, results in an accuracy of 91.1% for an existing product review dataset and and 89.9% for a Hindi movie review dataset; both numbers are nearly 10% higher than earlier state of the art. The results suggest that at the very least, it would be important to explore further with such methods for other applications.

1 Introduction

Over a period of nearly a millenium, there was an extensive debate between Indian grammarians on whether sentence meaning accrues by combining word meanings, or whether words gain their meanings based on the context they appear in (Matilal , 1990). The former position, that meaning is *compositional*, has been associated with the fregean enterprise of semantics, whereas recent models, building on large corpora of text (and associated multimedia) a large degree of success has accrued to models that attempt to model word meaning based on their linguistic context (e.g. (Landauer et al., 1997)). The latter line has resulted in strong improvements in several NLP tasks using word vectors (Collobert et al., 2008; Turian et al., 2010; Mikolov et al., 2013; Socher et al., 2013). The advantage of these approaches is that they can capture both the syntactic and the semantic similarity between words 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).

For resource-poor languages, these approaches have the added lure that many of these methods can work directly with large raw text corpora, and avoid contentious issues such as decicing on a POS-tagset, or expensive human annotated resources such as treebanks. For Indian languages therefore, it would be natural to seek to apply such methods. At the same time, it must be noted that many approaches combine POS-tags and even parse tree structures into the models for higher accuracies in specific tasks.

Vector models for individual words are obtained via distributional learning, the mechanisms for which varies from document-term matrix factorization (Landauer et al., 1997), various forms of deep learning (Collobert et al., 2008; Turian et al., 2010; Socher et al., 2013), optimizing models to explain co-occurrence constraints (Mikolov et al., 2013; Pennington et al., 2014),etc. Once the word vectors have been assigned, similarity between words can be captured via cosine distances.

One difficulty with this approach is that how words should be combined into larger phrases is not clear. In past work, inverse-similarity weighted averaging appears to work to some extent even for complex tasks such as essay grading (Landauer et al., 2003), but multiplicative mod-

els (based on a reducing the tensor products of the vectors) appears to correlate better with human judgements (Mitchell et al., 2008; Socher et al., 2013). Another complexity in composition is that composing words across phrasal boundaries are less meaningful than composing them within a phrase - this has led to models that evaluate the nodes of a parse tree, so that only coherent phrases are evaluated (Socher et al., 2013).

1.1 Sentiment Analysis

In order to evaluate the efficacy of the model, we apply it to the task of sentiment analysis. Here the problem is that of identifying the polarity of sentences such as

- Positive: रामू ने कहानी की रफ़्तार कहीं थमने नहीं दी [Ramu didn't allow the pace of the story to subside]
- Negative: पर्दे पर दिखाया जा रहा खौफ़ सिनेमाघर मे नहीं पसर पाता [The horror shown on the screen didn't flip out in the theater]

See (Liu et al. 2012) for a recent survey.

This is a problem that has attracted reasonable attention in Hindi (see section 2), since most sentiment analysis is oriented towards semantics, and one may bypass the syntactic processing which for Hindi is still poor. Methods that have been used are largely based on assigning a sentiment effect for individual words, and then combining these in some manner to come up with an overall sentiment for the document. Such methods, that ignore the order of the words have been criticized since the import of a sentence can change completely simply by re-arranging the words, which would leave the results unaffected. It is to improve such situations that several groups have attempted to model the composition of words into larger contexts (Mikolov et al., 2013; Socher et al., 2013; Johnson et al., 2014). However, most of the work on sentiment analysis in Hindi has not attempted to form richer compositional analyses. For the two corpora used here, the best results, obtained by combining a sentiment lexicon with hand-crafted analysis (e.g. modeling negation and "but" phrases), reach an accuracy of about 80%.

In this work, we first learn a distributional word vector model based on the wikipedia Hindi corpus, and then we use this to discern the polarities in on two existing corpora of movie and product reviews. To our own surprise, we find that even a simple additive composition model improves the state of the art in this task significantly, to just shy of 90% (a gain of nearly 10%). At the same time, when applied to English the system does respectably, but well behind the very best models that attempt more complex composition marginally. So the question arises as to whether the very significant gains in Hindi are due to some quirk in the dataset, or could it be that Hindi word vectors are particularly informative, maybe because of the much more highly inflected nature of its surface forms. Also, if the results are not corpus-specific, it also raises the possibility that word vector methods may result in significant gains in other similar problems for Hindi.

2 Related Work

Sentiment analysis is an important task in the field of NLP. SemEval-2014 had tasks from the domain of sentiment analysis which included aspect based sentiment analysis. It has emerged as one of the most important NLP tasks these days. Liu et al. (2012) and Pang et al. (2008) have provided a comprehensive literature of work done in this area. These days the focus has shifted mainly on microblogs(mainly Twitter) and reviews. Mohammad et al.(2013) used SVM as classifier and built a feature set consisting of punctuations, emoticons, n-grams, etc. from Twitter(2013) data and achieved F-score value of 89.14. Pang et al. (2004) did sentiment analysis on movie reviews and achieved accuracy of 87.2% with SVM classifier. They discarded objective sentences from the document and used text categorization techniques on the subjective ones. Le and Mikolov (2014) obtained 7.42% error rate on IMDB movie review dataset. They used distributed bag-of-words model, which they call as *paragraph vector*.

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.

In Hindi, due to lack of corpus, there has been a very limited amount of work done till now. Joshi et al. (2010) proposed a fallback strategy in their paper. They used a SVM classifier in order

to determine the polarity of the opinion, in their first approach which is named as in-language sentiment analysis. This strategy is based on 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 78.14.

Bakliwal et al.(2012) used graph based method to create lexicon. By using simple graph traversal, they determined how synonym and antonym relations can be used to generate the subjectivity lexicon. Their proposed algorithm achieved approximately 79% accuracy on classification of reviews. 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%. Mittal et al. (2013) developed an efficient approach based on negation and discourse relation to identifying the sentiments from Hindi content .They developed an annotated corpus for Hindi language and improve the existing Hindi SentiWordNet(HSWN) by incorporating more opinion words into it. Then they devised the rules for handling negation and discourse that affect the sentiments expressed in the review. Their proposed algorithm achieved 80.21% accuracy on classification of reviews.

3 Our Approach

3.1 Skip-Gram

Mikolov et al. (2013b) proposed two neural network models for building word vectors from large unlabelled corpora; Continuous Bag of Words(CBOW) and Skip-Gram. We have used skip-gram model for building word vectors in this case as it performs better than CBOW on small corpus. It is opposite of the CBOW model; here the input is the target word and output layer is the set of context words. It 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 equation below:

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

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).

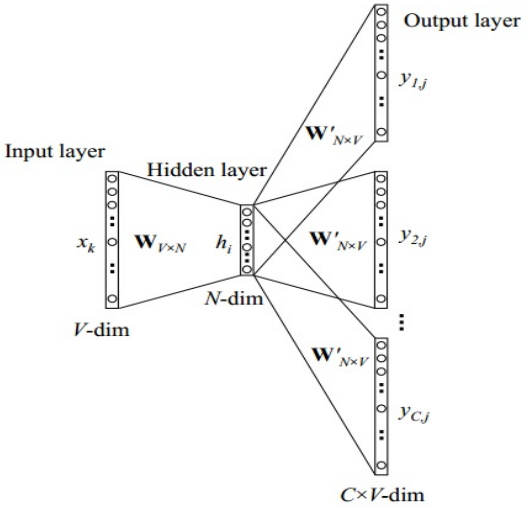


Figure 1: Skip Gram Model(Figure from Mikolov et al. (2013b))

3.2 Method

The heart of the methodology lies in generating word vectors from the raw text. For this purpose we first trained skip-gram model on the Hindi dataset. As an output, we now have a n-dimensional vector representation for each word in the corpus. So given a sentence/document, we have averaged out the vectors to create a single vector for that particular sentence/document. This vector averaging is inspired by Mikolov et al. (2013b) and Levy et al. (2014) in which they have shown that many relational similarities can be recovered by means of vector arithmetic in the embedded space. This models the sentence/document in a high dimensional space. We also trained our model on the Hindi Wikipedia dump to create vector representations for words. The previous two vectors were concatenated to create another feature set for training purpose.

4 Experiment Setup

This section describes the corpus used in our experiment along with different experiment models and parameters. In all the experiments, we did 20-fold cross validation to calculate classification accuracy using linear SVM.

4.1 Corpus

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. Similarly, for English, we trained on IMDB movie review dataset (Maas et al.(2013)) which consists of 25,000 positive and 25,000 negative reviews.
We also trained our skip-gram model on Hindi Wikipedia text dump (approx. 290MB) containing around 24M words with 724K words in the vocabulary. This provided us with good embeddings due to larger size and contents from almost all domains.

4.2 Word Embeddings

The quality of word vectors can be evaluated when they are compared with words which are closer to them semantically and syntactically. This can be accomplished by taking cosine similarity of each word with every other word in the vocabulary and outputting top few words which have greater cosine similarity. The other is to use tSNE (Maaten et al., 2008) which helps in visualizing high-dimensional data by giving each data point a location in a two or three-dimensional map. In our experiment, we took 5K words and plotted them with tSNE to highlight how these word vectors represent semantic and syntactic relations among words.

Figure 3 gives a closer look into few clusters which depicts the relation between words in high dimensional space. Figure 3(a) shows that words such as मौजूद and उपलब्ध are closer to each other but farther from words such as ज्यादा and अधिक which are closer to each other.

4.3 Skip-Gram and *tf-idf* based Word Vectors

In this experiment, we first generated 300-dimensional word vectors by training skip-gram model on both review corpus. The context size was taken as 5. 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. In this case, the dimension of each word vector obtained from skip-gram training was 500.

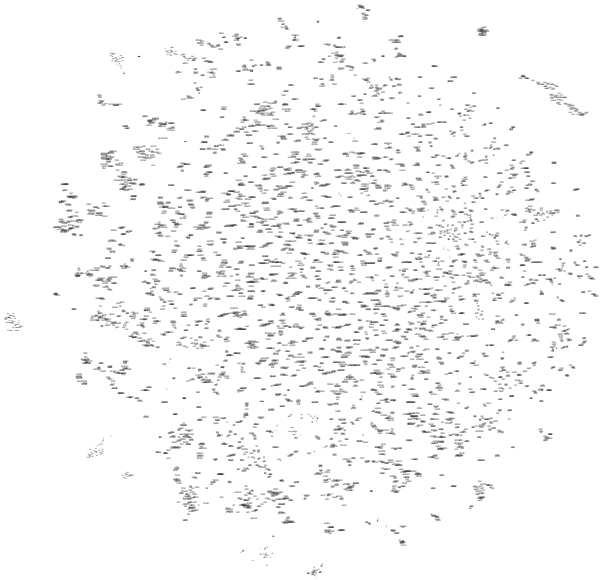
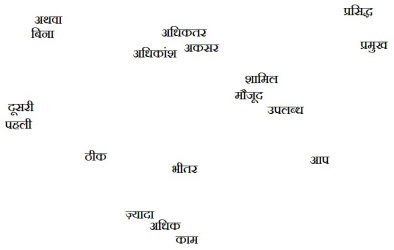
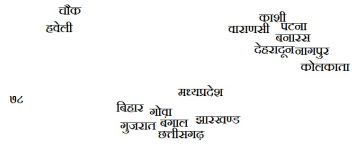


Figure 2: t-SNE visualization of Hindi Words in high dimensional space



(a)



(b)

Figure 3: A closer look at few clusters in the tSNE visualization

5 Results

Table 1 is a summary of results which we and others have obtained on the IMDB dataset.

Features	Accuracy
Maas et al.(2011)	88.89
Paragraph Vector (Le and Mikolov(2014))	92.58
WordVector Averaging+Wiki (Our Method)	87.56

Table 1: IMDB Movie Review Dataset

Table 2 represents the results which we have obtained on the Product Review Dataset and Movie Review Dataset using three different techniques for feature set construction. There has been a large improvement in accuracy when we include *tf-idf* features along with word vectors.

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**

Table 2: Accuracy(1):Product Review Dataset and Accuracy(2): Movie Review Dataset

Table 3 and 4 compares our best method(on Product and Movie Review Datasets respectively) with various other methods which have performed well using techniques such as *tf-idf*, subjective lexicon, etc.

Experiment	Features	Accuracy
Word Vector with SVM (Our method)	tf-idf with word vector	**91.14**
Subjective Lexicon (Bakliwal et al.(2012))	Simple Scoring	79.03
Hindi-SWN Baseline (Arora et al.(2013))	Adjective and Adverb presence	69.30

Table 3: Comparison of Approaches: Product Review Dataset

Experiment	Features	Accuracy
Word Vector with SVM (Our method)	tf-idf; word vector	**89.97**
In language using SVM (Joshi et al.(2010))	tf-idf	78.14
MT Based using SVM (Joshi et al.(2010))	tf-idf	65.96
Hindi-SWN Baseline (Arora et al.(2013))	Adj. and Adv. presence	68.40

Table 4: Comparison of Approaches: Movie Review Dataset

Table 5 shows the top few similar words for certain words from the corpus with cosine similarity as a distance metric.

अच्छा	खराब	भयानक
बहुत	निरासाजनक	भयन्कर
सुपर	कम्ज़ोर	भीषण
केवल	नाज़ुक	भयावह
इतना	बदतर	अवसाद

Table 5: Sentiment Words and their Neighbors

6 Conclusion and Future Work

Our word vector averaging method along with tf-idf outperforms all the existing methods. We also see that pruning stop words on the basis of frequency in the corpus also improves the accuracy slightly. The explanation for this improvement could be that words which have very high frequency tend to occur in most of the documents and they don't contribute to the sentiment. Words which have very low frequency can be considered as noise and can be safely pruned. For example, words such as फिल्म occur in 139/252 documents in Movie Review Dataset(55.16%) and have no effect on sentiment information of a document. Similarly words such as सिद्धार्थ occur in 2/252 documents in Movie Review Dataset(0.79%). These words don't provide much information.
Our experiment results clearly indicate that proposed methods are helpful for much accurate sentiment classification. We expect that we can further improve our model by giving higher weights to adjectives and adverbs. Also, skip-gram model assumes that raw text for training is of large size. But in our case the training corpus was very small. So we can expect much better performance with our proposed approach if we have a much larger sentiment corpus. We could provide a possible contribution in the construction of HSWN with the help of word vectors and their similarity. We could also incorporate word sense disambiguation and morphological variants for better accuracy while classification. The area of aspect-based sentiment analysis, also known as part-based sentiment analysis in Hindi is also unexplored which looks into several constituents of a document and classify their sentiment polarity individually.

Acknowledgements

We thank LTG, IIIT Hyderabad and CFLT, IIT Bombay for providing us with the datasets. We also thank Computer Science and Engineering Department, IIT Kanpur for providing us with all the resources for the successful execution of this work.

References

Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. 2010. A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study. *International Conference on Natural language Processing (ICON)*

Bimal Krishna Matilal. 1990. The word and the world: India's contribution to the study of language. *Oxford University Press, USA*

Bruno Ohana, and Brendan Tierney. 2009. Sentiment Classification of Reviews Using SentiWordNet. *9th. IT & T Conference*

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. *COLING*,

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

Jeff Mitchell, and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. *JMLR*, 08/236.244

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *EMNLP*, 12/abs

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 12/384.394

L.J.P. van der Maaten, and G.E. Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. *JMLR*, 08/2579.2605

Namita Mittal, Basant Agarwal, Garvit Chouhan, Nitin Bania, and Prateek Pareek. 2013. Sentiment Analysis of Hindi Reviews based on Negation and Discourse Relation. *In the proceedings of 11th Workshop on Asian Language Resources, (in conjunction with IJCNLP -2013)*, 13/45.50

Nishantha Medagoda, Subana Shanmuganathan, and Jacqueline Whalley. 2013. A Comparative Analysis of Opinion Mining and Sentiment Classification in non-English Languages. *International Conference on Advances in ICT for Emerging Regions (ICTer)*, 13/144.148

Omer Levy, and Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. *In Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 14/171.180

Piyush Arora, Akshat Bakliwal, and Vasudeva Varma. 2012. Hindi Subjective Lexicon Generation using WordNet Graph Traversal. *International Journal of Computational Linguistics and Applications*, 3.1/25.39

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2008. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12/2493.2537

Richa Sharma, Shweta Nigam, and Rekha Jain. 2014. Opinion Mining In Hindi Language: A Survey. *CoRR*, abs/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. *EMNLP*, abs/1631.1642.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, abs/1201.1211.

Rie Johnson, and Tong Zhang. 2014. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. *arXiv preprint*, abs/1412.1058.

Thomas K. Landauer, Darrell Laham, and Peter W. Foltz. 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor. *Automated essay scoring: A cross-disciplinary perspective*, abs/87.112.

Thomas K. Landauer, and Susan T Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104.2/211.240.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 26/3111.3119.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint* , abs/1309.4168.

Will Y. Zou, Richard Socher Weston, Daniel Cer, and Christopher D. Manning. 2013. Bilingual Word Embeddings for Phrase-Based Machine Translation. *EMNLP*, 13/1393.1398