Word vector models for Hindi: A Sentiment Analysis evaluation

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 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 quite easily. We then apply it to the sentiment analysis for two small Hindi databases from earlier work. In order to handle larger strings from the word vectors, several methods - additive, multiplicative, or tensor neural models, have been proposed. Here we find that even the simplest - an additive average, results in an impressive accuracy gain on state of the art by 10% (from 80%) for two review datasets. The results suggest that it may be worthwhile to explore such methods further for Indian languages.

1 Introduction

Over a period of nearly a millenium, Indian 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 , 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 are completely unsupervised and work directly with large raw text corpora, thus avoiding contentious issues such as deciding on a POS-tagset, or expensive human annotated resources such as treebanks. For Indian languages which are highly inflected, stemming or identifying the lemma is another problem which such models can overcome, provided the corpus is large enough. Nonetheless, this approach remains under-explored for Indian languages. At the same time, it must be noted that 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).

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 problem in this approach is that of combining the word vectors into larger phrases. 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 models (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). The results reported here, are based on applying the Skip-Gram model (Mikolov et al., 2013) to Hindi.

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 (Liu et al. 2012); for example:

- Positive: रामू ने कहानी की रफ़तार 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 reach the theater]

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 remains poor for Hindi. 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 ignore word order and have been criticized since the import of a sentence can change completely simply by re-arranging the words, though the sentiment evaluation remains the same. Several groups have attempted to improve the situation by modeling the composition of words into larger contexts (Le et al., 2014; 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 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).

In this work, we first learn a distributional word vector model based on the wikipedia Hindi corpus as well as the sentiment corpus, and then we use this to discern the polarities on the 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 (a gain of nearly 10%). When used for the much better-researched, larger datasets of English the system does respectably, but well behind the very best models that attempt more complex composition models. 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, e.g. owing to 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 a well-known research area in NLP today (see reviews in (Liu , 1990) and Pang et al. (2008), and also challenge in SemEval-2014). Early work on movie review sentiments 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. They used distributed bag-of-words model, which they call as *paragraph vector*. 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).

There has been limited work on sentiment analysis in Hindi – see review in (Medagoda et al., 2013), who surveys sentiment analysis in non-English languages). 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 (45 + 75 seed words given to the system); their final accuracy was 79.0% for the product review dataset. Mittal et al. (2013) incorporate hand-coded rules dealing with negation and discourse relations and extend the HSWN lexicon with more opinion words. Their algorithm achieves 80.2% accuracy on classification of movie reviews on a separate dataset.

3 Word-Vectors by using 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. 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 $w \in V$, $c \in C$, $i \in 1, ..., d$ (a total of $|C| \times |V| \times d$ parameters).

3.1 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 et al., 2008; Socher et al., 2013). In this work, we have retained teh simplicity of vector averaging to model larger chunks of discourse. This models the sentence/document in the same high dimensional space.

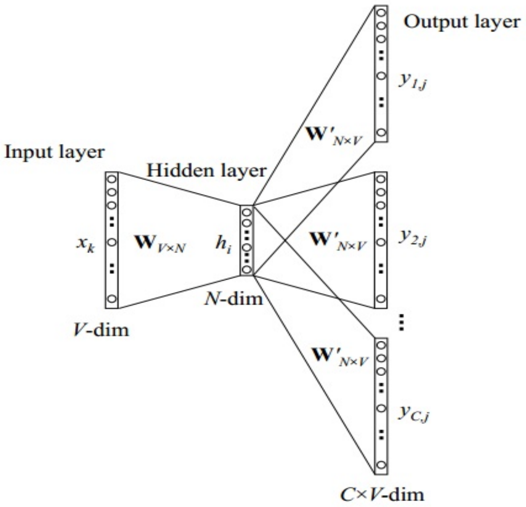


Figure 1: Skip Gram Model(Figure from Rong (2014))

A preprocessing step involved removing some words that appear at very high or very low frequencies in the corpus. Our model was trained 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.

Algorithm

1. Input the Hindi text corpus
2. Train skip-gram model to obtain word vector representation
3. Given a sentiment training set, obtain average vector data for each sentence/document
4. Obtain tf-idf vector for each sentence/document in the corpus
5. Concatenate vectors of step 3 and step 4 to obtain a feature set for a training instance
6. Train linear SVM with m-fold cross validation to create a classifier (here m=20)

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.

The quality of word vectors can be evaluated by comparing them with words which are closer to them semantically and syntactically. This is usually done via cosine similarity. Another evaluation can be done through tSNE (Maaten et al., 2008) which helps in visualization which maps each high-dimensional data point to a two or three-dimensional map. In our experiment, we took 5K words and plotted them with tSNE (fig. 3).

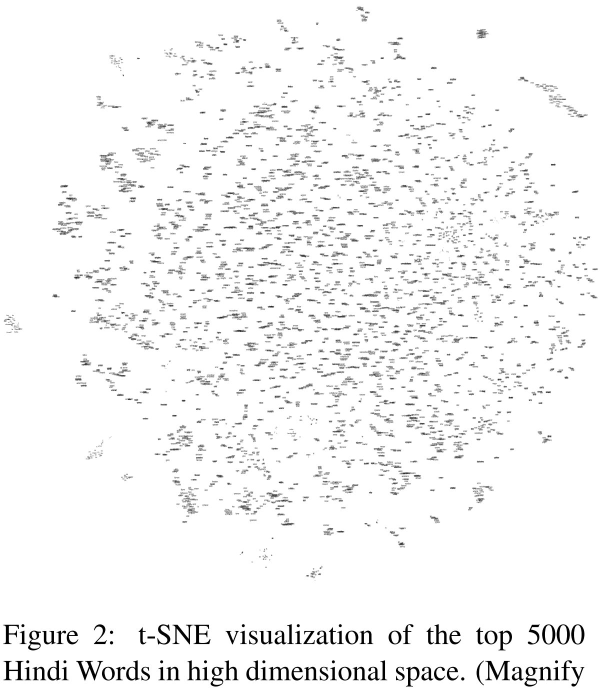
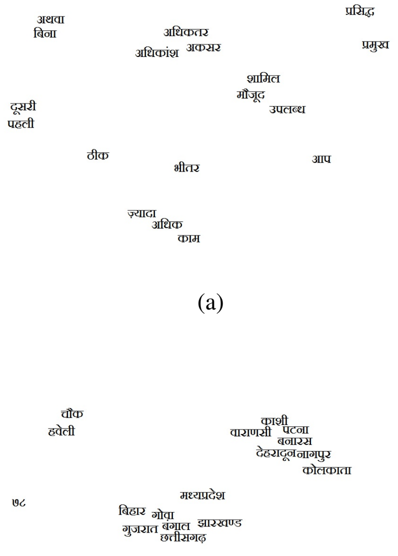


Figure 2: t-SNE visualization of the top 5000 Hindi Words in high dimensional space. (Magnify to see details).

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 ज़्यादा and अधिक.



(a)

(b)

Figure 3: A closer look at two clusters in the visualization showing a) quantity relations and b) locations.

4.2 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.

5 Results

Features	Accuracy
Dredze et al.(2008)	85.90
Max Entropy	83.79
WordVector Averaging (Our Method)	**89.23**

Table 1: Results on Amazon Electronics Review 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 2: Results on IMDB Movie Review Dataset

Table 2 represents the results using three different techniques for feature set construction.

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 3: Accuracies for Product Review and Movie Review Datasets.

Table 3 and 4 compares our best method 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 4: Comparison of Approaches: Product Review Dataset

Experiment	**Features**	**Accuracy**
WordVector Averaging	word vector	78.0
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
Improved Hindi-SWN (Bakliwal et al.(2012))	Adj. and Adv. presence	79.0

Table 5: 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 6: Some sentiment words and their neighbors

6 Conclusion and Future Work

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 89.9%).

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 focussed 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.

In our future work, we seek to explore various compositional models - a) weighted average - where weights are determined based on cosine distances in vector space; b) multiplicational models. Another aspect we are considering is to incorporate multiple word vectors for the same surface token in cases of polysemy - this would directly be useful for word sense disambiguation. 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.

We also 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.

Before concludiong, 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.

References

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

Akshat Bakliwal, Piyush Arora, and Vasudeva Varma. 2012. Hindi Subjective Lexicon: A Lexical Resource For Hindi Polarity Classification. *Proceedings of the Eight Intl Conference on Language Resources and Evaluation (LREC)*,

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

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Morgan and Claypool Publishers*

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-2014*.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. *ACL-2012*.

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

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. *ACL-10*,

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

Namita Mittal, Basant Agarwal, Garvit Chouhan, Nitin Bania, and Prateek Pareek. 2013. Sentiment Analysis of Hindi Reviews based on Negation and Discourse Relation. *Proc. 11th Workshop Asian Language Resources, (IJCNLP -2013)*,

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

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*,

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

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-13*,

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. *EMNLP-12*,

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

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*, Routledge.

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*,

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *NIPS-13*,

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

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