Word Vector Models for Hindi: A Sentiment Analysis Evaluation

PRANJAL SINGH
spranjal@iitk.ac.in

AMITABHA MUKERJEE
amit@cse.iitk.ac.in

Index

Motivation

- Distributional semantics or word vector models have been proposed to capture both the syntactic and semantic similarity between words

- Unsupervised approach brings an interest to utilize this method for under-resourced languages such as Hindi

- Most of the works till now have been based on SentiWordNet and this area is unexplored

Earlier Work

- Earlier methods 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.

- They 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

Earlier Work

- 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) focused on adjectives/adverbs to achieve **79.0%** accuracy

- Mittal et al. (2013) incorporate hand-coded rules dealing with negation and discourse relations and extend the HSWN lexicon with more opinion words to achieve **80.2%** accuracy on movie reviews

Mikolov et al.(2013b)

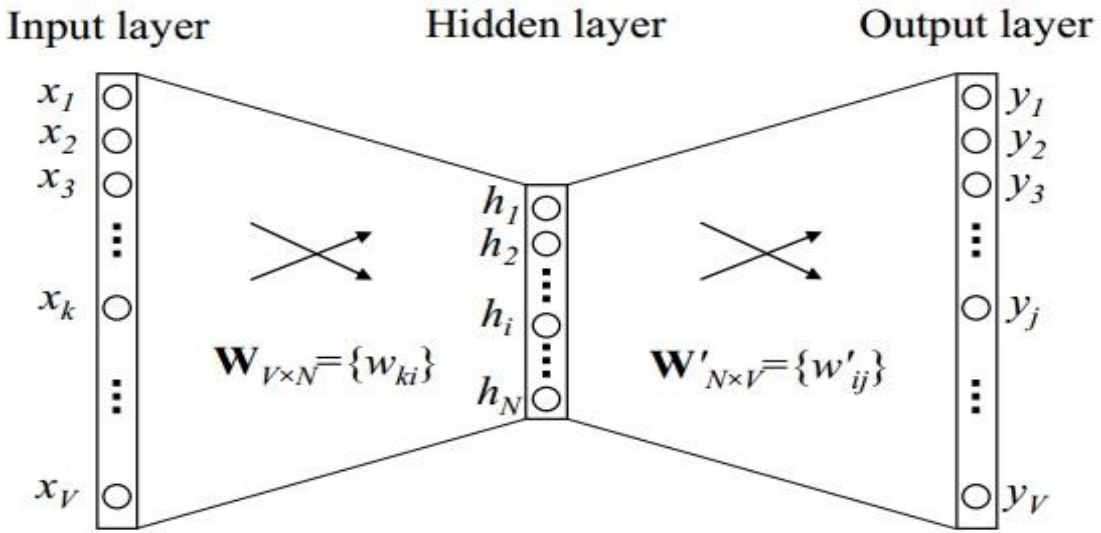
- Two architectures
  1. Continuous Bag-of-Word (CBOW)
  2. Skip Gram

CBOW: Predict the word given the context

Skip-Gram: Predict the context given the word

One Word Context (CBOW)

- Vocabulary size is **V** and hidden layer size is **N**
- Input vector is one-hot encoded vector, i.e., only one node of $\{x_1,.....,x_v\}$ is 1 and others 0
- Weights between the input layer and the output layer can be represented by a $V \times N$ matrix **W**



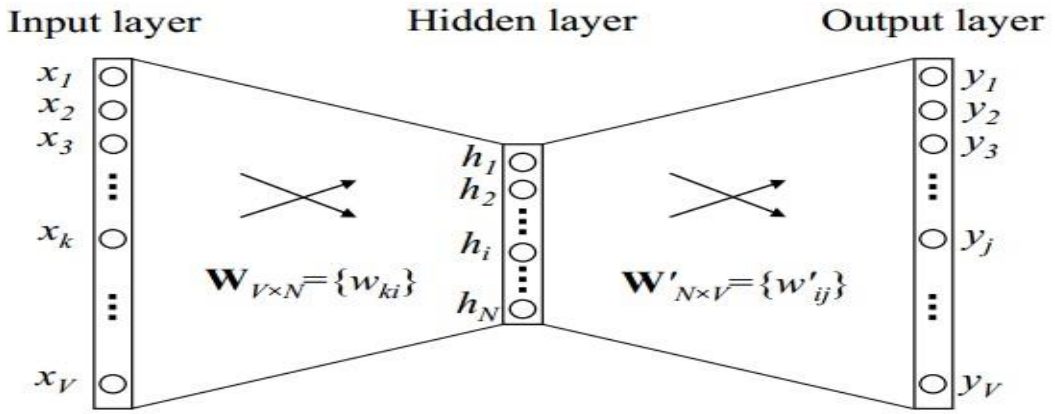
One Word Context (CBOW)

$$h = \mathbf{x}^T \mathbf{W} = \mathbf{v}_{Wi}$$

$\mathbf{v}_{wI}$ is the vector representation of the input word w_I

$$u_j = \mathbf{v}'^{T}_{w_j} \cdot \mathbf{h}$$

u_j is score of each word in vocabulary and v'_{w_j} is the *j-th* column of matrix **W'**



One Word Context (CBOW)

We then use soft-max, a log-linear classification model, to obtain the posterior distribution of words, which is a multinomial distribution

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

$$p(w_j|w_I) = \frac{\exp\left({\mathbf{v}'_{w_O}}^T \mathbf{v}_{w_I}\right)}{\sum_{j'=1}^{V} \exp\left({\mathbf{v}'_{w'_j}}^T \mathbf{v}_{w_I}\right)}$$

One Word Context (CBOW)

Update Equation for hidden->output weights

The training objective is to maximize the conditional probability of observing the actual output word w_O (denote its index in the output layer as j^*) given the input context word w_I with regard to the weights

$$\begin{aligned} \max p(w_O|w_I) &= \max y_{j^*} \\ &= \max \log y_{j^*} \\ &= u_{j^*} - \log \sum_{j'=1}^{V} \exp(u_{j'}) := -E \end{aligned}$$

Using Stochastic Gradient Descent, update equation becomes,

$$\mathbf{v}'^{(\text{new})}_{w_j} = \mathbf{v}'^{(\text{old})}_{w_j} - \eta \cdot e_j \cdot \mathbf{h} \qquad \text{for } j = 1, 2, \cdots, V.$$

One Word Context (CBOW)

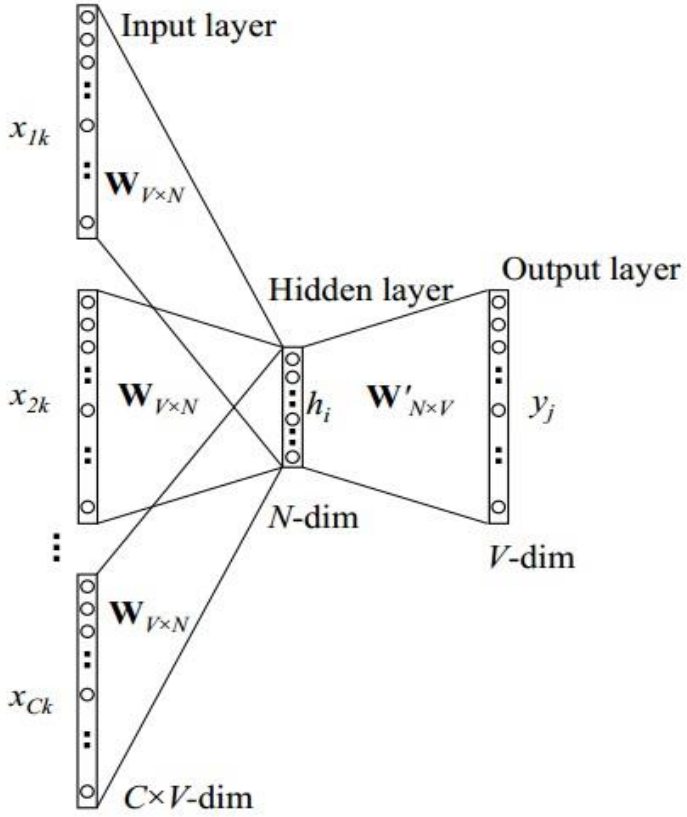
Update Equation for input->hidden weights

$$\mathbf{v}_{w_I}^{(\text{new})} = \mathbf{v}_{w_I}^{(\text{old})} - \eta \cdot \text{EH}$$

Multi Word Context (CBOW)

$$\begin{aligned}\mathbf{h} &= \frac{1}{C}\mathbf{W}\cdot(\mathbf{x}_1+\mathbf{x}_2+\cdots+\mathbf{x}_C)\\ &= \frac{1}{C}\cdot(\mathbf{v}_{w_1}+\mathbf{v}_{w_2}+\cdots+\mathbf{v}_{w_C})\end{aligned}$$

$$\begin{aligned}E &= -\log p(w_O|w_{I,1},\cdots,w_{I,C})\\ &= -u_{j^*}+\log\sum_{j'=1}^{V}\exp(u_{j'})\\ &= -\mathbf{v}'^{T}_{w_O}\cdot\mathbf{h}+\log\sum_{j'=1}^{V}\exp(\mathbf{v}'^{T}_{w_j}\cdot\mathbf{h})\end{aligned}$$



Multi Word Context (CBOW)

Update Equations

1) $$\mathbf{v}'^{(\text{new})}_{w_j} = \mathbf{v}'^{(\text{old})}_{w_j} - \eta \cdot e_j \cdot \mathbf{h} \qquad \text{for } j = 1, 2, \cdots, V.$$

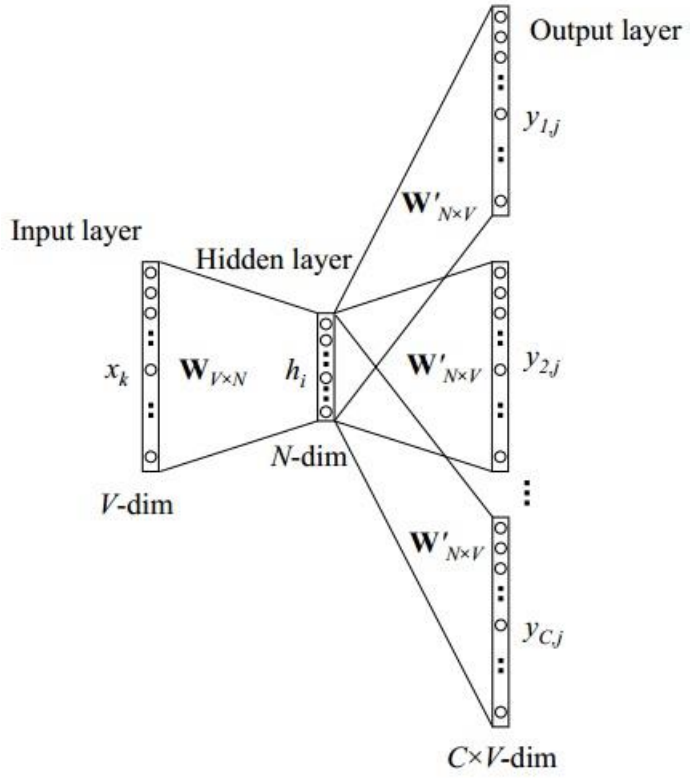
2) $$\mathbf{v}^{(\text{new})}_{w_{I,c}} = \mathbf{v}^{(\text{old})}_{w_{I,c}} - \frac{1}{C} \cdot \eta \cdot \text{EH} \qquad \text{for } c = 1, 2, \cdots, C.$$

Skip-Gram Model

- It is the opposite of CBOW Model

- On the output layer, instead of outputing one multinomial distribution, we are outputing *C* multinomial distributions.

- Each output is computed using the same hidden →output matrix

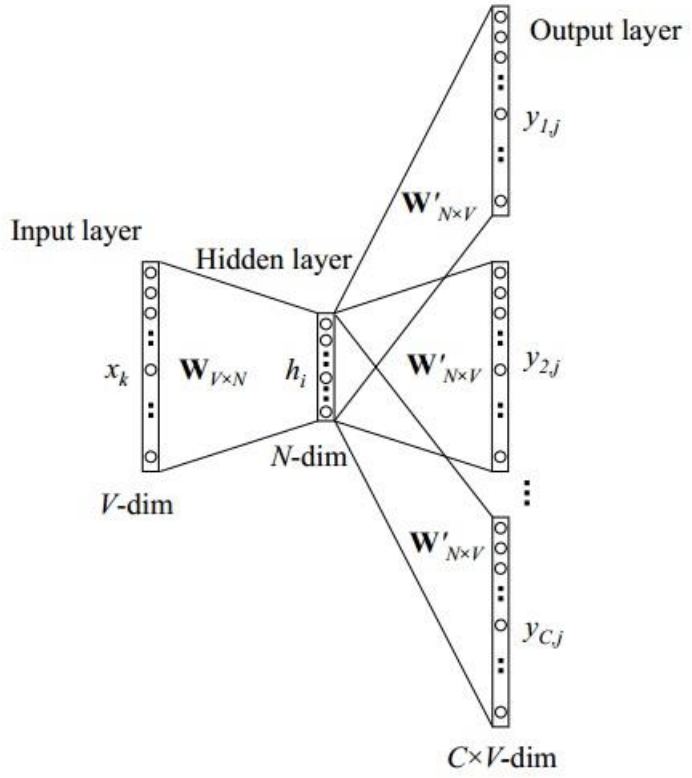


$$p(w_{c,j} = w_{O,c}|w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

Skip-Gram Model

Loss Function

$$
\begin{aligned}
E &= -\log p(w_{O,1}, w_{O,2}, \cdots, w_{O,C}|w_I) \\
&= -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})} \\
&= -\sum_{c=1}^{C} u_{j_c^*} + C \cdot \log \sum_{j'=1}^{V} \exp(u_{j'})
\end{aligned}
$$



Optimization

1. Hierarchical Softmax (Morin & Bengio, 2005)

2. Negative Sampling (Gutmann & Hyvarinen, 2012)

tf-idf(term frequency–inverse document frequency)

- It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus

- The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus

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

Here $df(t)$ is the number of documents in which term t appears.

D is the total number of documents

tf(d,t) is the term-frequency

Sentiment Analysis

- Sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document.

- A basic task in sentiment analysis is classifying the *polarity* of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

(Source: Wikipedia)

Sentiment Analysis

- 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 खौफ सिनेमाघर में नहीं पसर पाटा

  [The horror shown on the screen didn't reach the theater]

Here the problem is that of identifying the polarity of sentences (Liu et al., 2012)

- 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
- Most of the work on sentiment analysis in Hindi has not attempted to form richer compositional analyses.

Additive Composition

- 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.

- Additive Composition is addition of vectors in high dimensional space and it represents composite meaning of these vectors (e.g. addition of word vectors for a sentence)

- There have been debates for over a period of a millennium that whether sentence meaning accrues by combining word meanings, or whether words gain their meanings based on the context they appear (Matilal, 1990)

- Here, we find that even the simplest - an additive average, results in an impressive accuracy gain on state of the art

Methodology

- 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

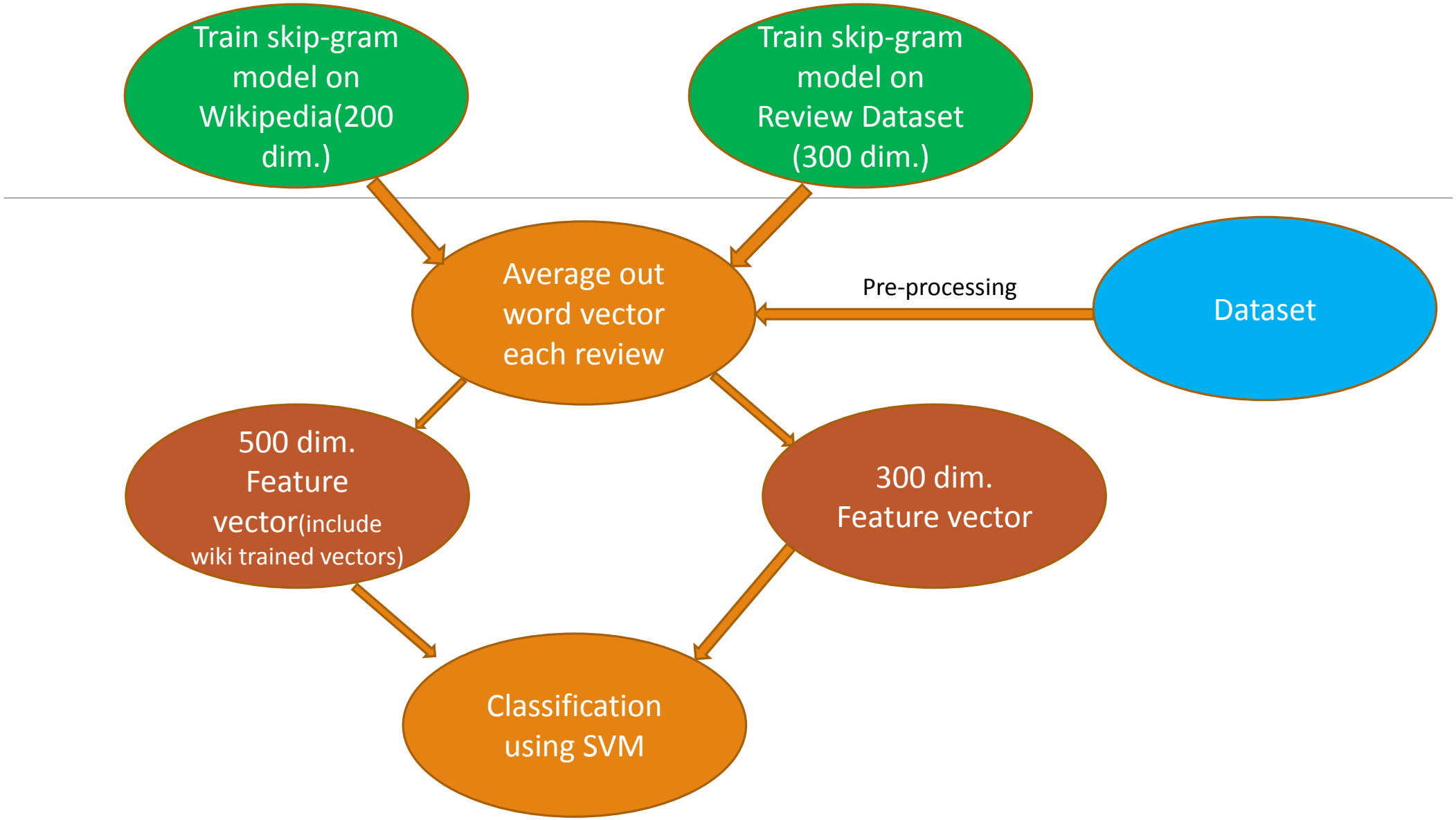
- We then use *additive composition* to form vectors for each document/sentence

- We also use *tf-idf* to form a richer feature set

Methodology

Preprocessing step involved removing some words that appear at very high or very low frequencies in the corpus

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)


Train skip-gram model on Wikipedia(200 dim.)
Train skip-gram model on Review Dataset (300 dim.)
Average out word vector each review
Pre-processing
Dataset
500 dim. Feature vector(include wiki trained vectors)
300 dim. Feature vector
Classification using SVM

Dataset

1. IMDB 50,000 Movie Review Dataset(English)
   I. Contains 25,000 Positive and 25,000 Negative reviews
   II. 25,000 training examples and 25,000 testing examples

2. Hindi Product Review Dataset(IIIT)
   I. Contains 350 Positive and 350 Negative reviews

3. Hindi Movie Review Dataset(IITB)
   I. Contains 127 Positive and 125 Negative reviews

Dataset

- Hindi :Wikipedia text dump (290 MB) ~4 mins of training
  - 23848940 words
  - 723737 words in vocabulary
  - 106876 words after removing words with *freq<5*
- English: Wikipedia text dump (9.3 GB) ~3.5 hrs of training
  - 1,703,849,452 words
  - 13,027,758 words in vocabulary
  - 1,778,685 words after removing words with *freq<5*

Result (Hindi)

Odd One out

1) भारत रूस मुम्बई चीन

Ans: मुम्बई

2) लड़की बेहन महिला मर्द

Ans: मर्द

3) नेता मंत्री सरकार उद्योग

Ans: उद्योग

Result (Hindi)

Top Similar Words

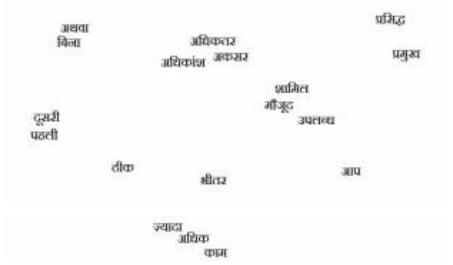
अच्छा	खराब	भयानक
बहुत	निरासाजनक	भयन्कर
सुपर	कम्ज़ोर	भीषण
केवल	नाज़ुक	भयावह
इतना	बदतर	अवसाद

Result (Hindi)

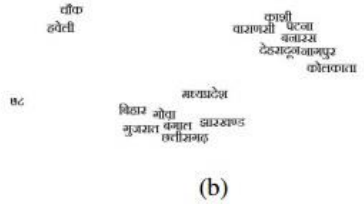
Top Similar Words

गूगल	मूवी	जनवरी	रसायन
वेब	कॉमेडी	मई	खगोल
माइक्रोसॉफ्ट	लास्ट	नवंबर	भौतिकी
जीमेल	स्टोरी	अक्टूबर	जीवविज्ञान
सर्वर	म्यूजिक	अगस्त	अभियांत्रिकी
इंटरनेट	सीरीज	नवंबर	शरीरक्रिया

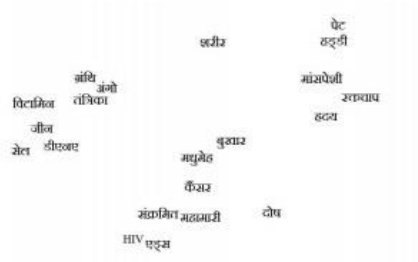
tsne Visualization of few Clusters



(a)



(b)



(c)

Result(English-IMDB)

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

*** Skip-gram model was trained with vector dimension as 300 and min word-count as 20**

Result(Hindi-Product Review: IIIT)

Feature Set	Accuracy(%)
Word Vector Averaging	78.0
Word Vector Averaging + tf-idf	90.73
Word Vector Averaging + tf-idf without stop words	**91.14**

Result(Hindi-Movie Review: IITB)

Feature Set	Accuracy(%)
Word Vector Averaging	79.62
Word Vector Averaging + tf-idf	89.52
Word Vector Averaging + tf-idf without stop words	**89.97**

Result(Hindi-Product Review: IIIT)

Comparison of Approaches

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

Result(Hindi-Movie Review: IITB)

Comparison of Approaches

Experiment	Features	Accuracy(%)
Word Vector Averaging	word vector	78.0
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))	Adjective and Adverb Presence	79.0
Word Vector with SVM (our method)	tf-idf with word vector	**89.97**

Conclusion

- 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

- 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.

Future Work

- 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.
- 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.
- 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.

References

1. 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)*
2. 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)*.
3. Bimal Krishna Matilal. 1990. The word and the world: India's contribution to the study of language. *Oxford University Press, USA*
4. Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Morgan and Claypool Publishers*
5. Bruno Ohana, and Brendan Tierney. 2009. Sentiment Classification of Reviews Using SentiWordNet. *9th. IT & T Conference*
6. 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*.
7. 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*.
8. Jeff Mitchell, and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. *JMLR*.
9. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *EMNLP*.
10. Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. *ACL-10*.
11. L.J.P. van der Maaten, and G.E. Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. *JMLR*.
12. Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014. Frege in Space: A Program of Compositional Distributional Semantics. *Linguistic Issues in Language Technology.*

References

13. 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)*.
14. 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)*.
15. 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*.
16. Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2008. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*
17. 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*.
18. Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. *EMNLP-12*.
19. Rie Johnson, and Tong Zhang. 2014. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. *arXiv preprint*.
20. 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.
21. 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*.
22. Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *NIPS-13*.
23. Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*.
24. Quoc V. Le, and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint.*

Thank You!!!