Sentiment Analysis In Hindi

Naman Bansal and Umair Z Ahmed
Advisor: Amitabha Mukherjee
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India
{namanb, umair} @cse.iitk.ac.in

Abstract

The basic task of Sentiment Analysis is to classify the polarity of the opinion expressed in a given text into positive or negative. With the rise of social networks, blogs, online ratings websites, etc, this task has become even more important in recent times to provide succinct summary of the online expression. In this project, we use Semi-Supervised approaches to train a Deep Belief Network on a small percentage of labelled data and assign polarity to unlabelled data. We report accuracies on the IIT-Bombay Hindi movie review dataset and also on online movie reviews manually collected and annotated by us. We are able to achieve an accuracy of 64% on test set, by training on as few as 150 labeled reviews.

1. Introduction

Sentiment analysis is the task of classifying the polarity of a given text at the document/sentence/feature level into positive or negative (and in some cases neutral) class. Labeling the reviews with their sentiment would provide succinct summaries to readers. Also, sentiments reflect the opinion of the user on a product which would be of high value to its company. Sentiment analysis is used in a multitude of applications ranging from recommender systems to business intelligence applications.

Most of the research in this domain has been focused on English language and work has been done at syntactic, semantic and discourse levels. Previous work on Sentiment analysis of Indian Languages has mainly focused on supervised methods though. We attempt to detect the sentiment of sentences written in Hindi language with devanagari script, using semi-supervised approaches. The main challenges compared to English language are:

- Hindi is morphologically rich and is a free order language as compared to English.

- Also, the scarcity of resources for the Hindi language brings challenges ranging from collection and generation of datasets.

We use semi-supervised approaches since

- Supervised polarity classification systems are typically domain-specific and hence systems trained on one dataset typically perform much worse on a different dataset.
- There is very little annotated data available for low resource languages such as Hindi. And, annotating a large amount of data is an expensive process.

2. Previous Works

Sentiment analysis for Indian Languages has primarily been focussing on using:

1. Machine Translation to translate the data in English to Hindi.
2. Bi-Lingual dictionary for English and Indian Languages
3. Hindi WordNet expansion to exploit synonyms and antonym polarity

Recent contribution in Hindi Subjective Lexicon was done by Bakliwal et al. [2], where they created a resource for Hindi Polarity Classification. They used Hindi WordNet to retrieve synonyms and antonyms of a given word in hindi for which they knew the polarity and then assigned the similar polarity to synonyms and opposite polarity to antonyms.

Labelling emotion in Bengali blog corpus by Das et al. [3]. They manually annotated the corpora at sentence level and then used some standard techniques like Kaapa and MASI to handle the differences between the annotators.

Joshi et al., 2010 created H-SWN (Hindi-SentiWordNet) using two lexical resources namely English SentiWordNet and English-Hindi WordNet Linking. Using WordNet linking they replaced words in English SentiWordNet with equivalent Hindi words to get H-SWN.

3. Datasets

300 sentences of movie review are available from IIT Bombay for research purposes, it includes 150 positive and 150 negative reviews. In addition to this we have manually collected around 300 sentences from hindi movie review site (j*agran.com*). We have stored the reviews in a xml format so that it is easy to parse .

We have following information in our manually collected review:

- Stars given to the movie by the rater
- Link from which the review has been taken
- Positively Annotated lines form the review
- Negatively Annotated lines form the review
- Full review for the movie and its sentiment

```
<movie id="87" star="2.5" link="
http://www.jagran.com/entertainment/reviews-film-review-heroin-M00978.html">
    <selectedLines>
        <line sentiment="pos">अभिनय क्षमता का कमाल है कि वह इस कमजोर किरदार में भी अपनी छाप छोड़ जाती हैं। हीरोइन सिर्फ करीना कपूर की
        भाव-भंगिमाओं और अदाओं के लिए देखी जा सकती है।अपने हिस्से के दृश्यों को संजीदगी से निभाया है।
        </line>
        <line sentiment="neg">साधारण और औसत फिल्म निकली। हीरोइन उनकी पिछली फिल्मों की तुलना में कमजोर और एकांगी है।फिल्म में ऐसे एक्सपोजर की
        खास जरूरत नहीं थी। उसे हमारी हमदर्दी नहीं मिल पाती।ऐसा लगता है कि संवाद पहले अंगेरजी में लिखे गए हों और फिर उनका हिंदी अनुवाद कर दिया गया हो। संवादों में हिंदी
        की रवानी नहीं है। उम्मीदों पर खरी नही उतरी हीरोइन।
        </line>
    </selectedLines>
    <review sentiment="pos">
कल तक हीरोइन की हर तरफ चर्चा थी। निर्माण के पहले हीरोइनों की अदला-बदली से विवादों में आ जाने की वजह से फिल्म के प्रति जिज्ञासा भी बढ़ गई थी। और फिर करीना कपूर जिस तरह से
जी-जान से फिल्म के प्रचार में जुटी थीं, उस से तो यही लग रहा था कि उन्होंने भी कुछ भांप लिया है। रिलीज के बाद से सारी जिज्ञासाएं काफूर हो गई हैं। मधुर भंडारकर की हीरोइन साधारण
और औसत फिल्म निकली। हीरोइन उनकी पिछली। .....
    </review> <!-->The entire review.</!-->
</movie>
```

Example of our collected reviews

Note: We are looking at the sentence level sentiment and hence, the word review in the rest of the paper implies a single sentence review.

4. Data Preprocessing

For each movie review, we first remove noise which doesn't contribute (or hurts the accuracy of classification) such as:

- Punctuations
- Numbers
- Words of length one
- Words that occur only in a single review
- Words with high document frequency, many of which are stopwords or domain specific general-purpose words

Then, each review is represented as a vector of unigrams, using binary weight equal to 1, for terms present in a vector. The dataset is represented as a Matrix X of size Dx(R+T).

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{R+T}\right] = \begin{bmatrix} x_1^1, x_1^2, \ldots, x_1^{R+T} \\ x_2^1, x_2^2, \ldots, x_2^{R+T} \\ \vdots, \vdots, \ldots, \vdots \\ x_D^1, x_D^2, \ldots, x_D^{R+T} \end{bmatrix} \quad [1]$$

where R is the number of training samples/reviews, T is the number of test samples and D is the number of feature words in the dataset.

And similarly, the class labels corresponding to L labeled training reviews are represented as a Matrix Y where C = the number of classes (in our case, 2)

$$\mathbf{Y}^L = \left[\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^L\right] = \begin{bmatrix} y_1^1, y_1^2, \ldots, y_1^L \\ y_2^1, y_2^2, \ldots, y_2^L \\ \vdots, \vdots, \ldots, \vdots \\ y_C^1, y_C^2, \ldots, y_C^L \end{bmatrix} \quad [1]$$

These matrices are then passed to a Deep Belief Network to seek the mapping function $\mathbf{X}^L \rightarrow \mathbf{Y}^L$ using the L labeled data and R+T-L unlabeled data.

5. Deep Belief Networks

Deep Belief Networks are similar to neural networks but they differ in the number of hidden layers. Deep Belief Networks have multiple hidden layers stacked one over the other to capture the complex non-linearity in the data.
They essentially disentangle the underlying variation in the data. They have been successfully applied for sentiment analysis in English with accuracy of about 71% and we are exploring the same technique for Hindi language[1].

Deep Belief Networks are used to capture the underlying non-linearity which explains the variation in data. Deep Belief Networks have multiple layers so they can easily capture the complex non-linearity in the variation of the input data. Our main motivation for using the Deep Belief Networks was that:

1. They can model complex non-linear phenomenon
2. They can exploit unlabeled data

General Deep Learning Architecture has following properties:

- One Input Layer $\mathbf{h^0}$
- N hidden layers $\mathbf{h^1, h^2, \ldots, h^N}$
- One Output Layer
- The input layer $\mathbf{h^0}$ has **D** units, equal to the number of features of sample data **x**.

We intend to seek the mapping function $\mathbf{X^L} \rightarrow \mathbf{Y^L}$ using the L labeled data and R+T-L unlabeled data.

The semi-supervised learning method based on DBN architecture can be divided into two stages[1]:

1. First, is the Pre-training of the model in which the DBN architecture is constructed by **greedy layer-wise unsupervised learning** using Restricted Boltzman Machine - RBMs as building blocks. All the unlabeled data together with L labelled data are utilized to find the parameter space **W with N layers.**
2. Second, is the Fine Tuning step in which the DBN architecture is trained according to the **negative log likelihood function using gradient descent method**. The parameter space **W is retrained by** a negative log likelihood cost function using L labelled data to fine tune the parameters of the parameter space only according to labelled data.

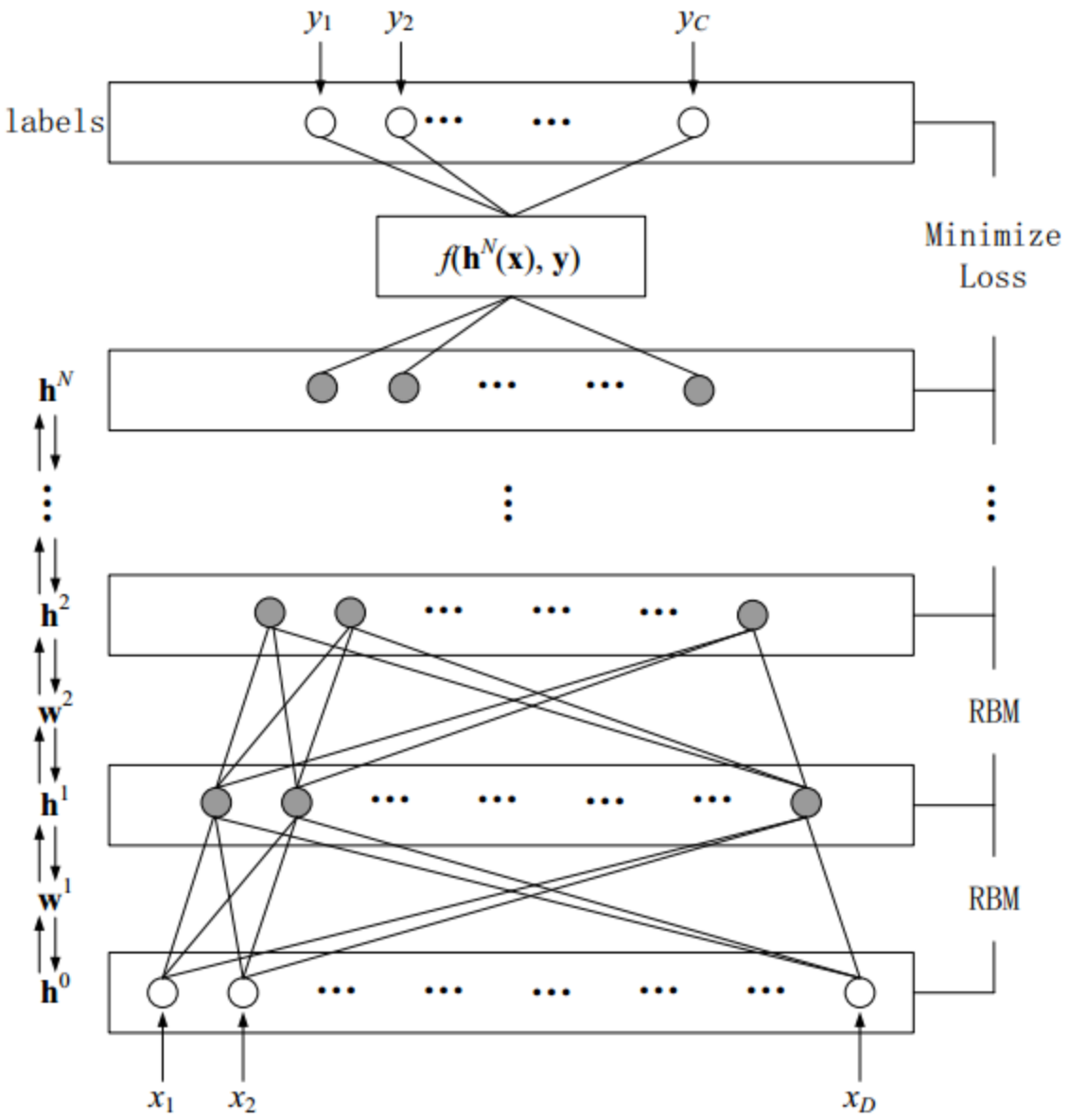

Figure 1. Deep Belief Network Architecture [1]

Energy of the state($\mathbf{h}^{k-1}$,$\mathbf{h}^{k}$) is defined as

$$E\left(\mathbf{h}^{k-1},\mathbf{h}^{k};\theta\right)=-\sum_{s=1}^{D_{k-1}}\sum_{t=1}^{D_k}w_{st}^{k}h_{s}^{k-1}h_{t}^{k}$$
$$-\sum_{s=1}^{D_{k-1}}b_{s}^{k-1}h_{s}^{k-1}-\sum_{t=1}^{D_k}c_{t}^{k}h_{t}^{k}$$

where $\mathbf{w}_{st}^{k}$ is the weight between the $\mathbf{h}^{k-1}$ which is the ***k-1***th hidden layer and $\mathbf{h}^{k}$ which is the ***k***th hidden layer .The term $\mathbf{b}_{s}^{k-1}$ is the bias for ***s***th unit of the ***k-1***th hidden layer and $\mathbf{c}_{t}^{k}$ is the bias for the **t**th unit of the ***k***th hidden layer.

The probability that the DBN model assigns to the hidden layer $\mathbf{h}^{k-1}$is given by

$$P\left(\mathbf{h}^{k-1};\theta\right)=\frac{1}{Z(\theta)}\sum_{\mathbf{h}^{k}}\exp\left(-E\left(\mathbf{h}^{k-1},\mathbf{h}^{k};\theta\right)\right)$$
$$Z(\theta)=\sum_{\mathbf{h}^{k-1}}\sum_{\mathbf{h}^{k}}\exp\left(-E\left(\mathbf{h}^{k-1},\mathbf{h}^{k};\theta\right)\right)$$

$Z(\Theta)$ is the normalizing constant[1].

The probability of turning on unit *t* is a logistic function of the states of $\mathbf{h}^{k-1}$ and $\mathbf{w}_{st}^{k}$

$$p\left(h_{t}^{k}=1\,|\,\mathbf{h}^{k-1}\right)=\mathrm{sigm}\left(c_{t}^{k}+\sum_{s}w_{st}^{k}h_{s}^{k-1}\right)$$

The probability of turning on unit *t* is a logistic function of the states of $\mathbf{h}^{k}$ and $\mathbf{w}_{st}^{k}$

$$p\left(h_{s}^{k-1}=1\,|\,\mathbf{h}^{k}\right)=\mathrm{sigm}\left(b_{s}^{k-1}+\sum_{t}w_{st}^{k}h_{t}^{k}\right)$$

The logistic function is given by:

$$\mathrm{sigm}(\eta)=1/1+e^{-\eta}$$

Optimization problem is formulized as:

$$\arg\min_{\mathrm{h}^{N}}\quad \mathrm{f}\left(\mathrm{h}^{N}\left(\mathbf{X}^{L}\right),\mathbf{Y}^{L}\right)$$

$$\mathrm{f}\left(\mathrm{h}^{N}\left(\mathbf{X}^{L}\right),\mathbf{Y}^{L}\right)=\sum_{i=1}^{L}\sum_{j=1}^{C}\mathrm{T}\left(\mathrm{h}_{j}^{N}\left(\mathbf{x}^{i}\right)y_{j}^{i}\right)$$

6. Experiments

We ran the code for multiple configurations of number of neurons in the hidden layers. We also experimented with multiple number of hidden layers and found the following configuration for the deep belief network to be best.

- Five layer network, One Input, Three Hidden and One Output Layer
  --Neurons in the Input Layer is equal to the vocabulary size
  --Neurons in Hidden Layer= [100,50,20]
  --Neuron in Output Layer = [1, 0] (positive and negative)

We intend to seek the mapping function $X^L \rightarrow \mathbf{Y}^L$ using the L labelled data and R+T-L unlabeled data. We made use of Theano Library[1] for its implementation of deep belief networks.

7. Results

We verify the performance of semi-supervised learning with different percentage of labeled data by testing the DBN on validation and the held out test data set. The results of running the above experiment on IIT-Bombay movie review dataset and our manually annotated movie reviews from Jagran website are then plotted in Figure 2 and Figure 3 respectively.

As seen from these graphs, there is a significant change in the accuracy as the % of training size is increased. In general, the validation set accuracy increases (from minimum of 50% to a maximum of 76% on IIT-B movie dataset) with increase in number of labeled training data. Where as, the accuracy on test set although initially increases with increase in training size (from minimum of 50% to maximum of 64% on IIT-B movie dataset), it drops slightly after some point since the DBN has overfit its model while trying to minimize the loss on validation set.

Also, another point to note is that DBN reaches decent performance even with very few labeled reviews. As seen from the figures below, the DBN obtains relatively high accuracy (of 50%) even with just 10% (~30) labeled reviews as training set!

[1] http://deeplearning.net/software/theano/

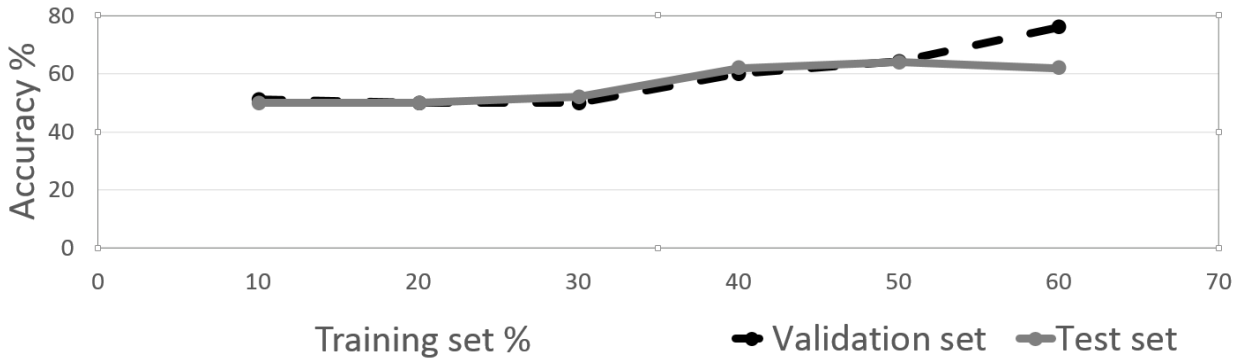

Figure 2: Validation and Test accuracy of DBN on IIT-Bombay hindi movie dataset, with different % of labeled reviews for training.

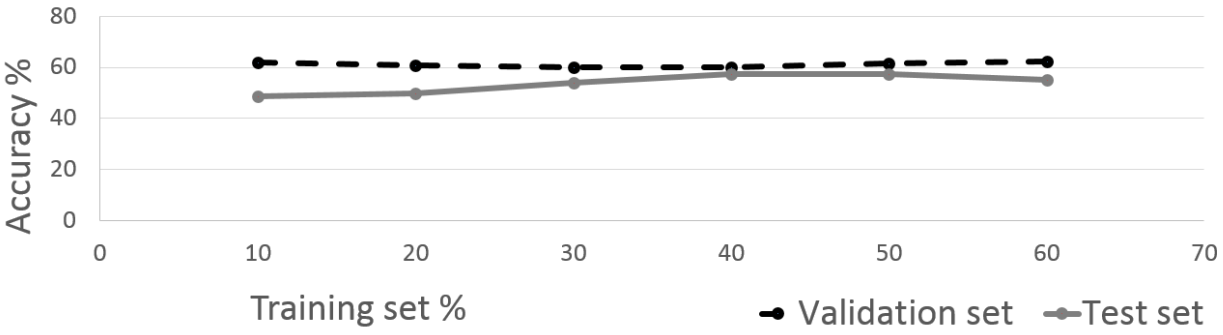

Figure 3: Validation and Test accuracy of DBN on our manually annotated Jagran website movie review dataset, with different % of labeled reviews for training.

8. Conclusion and Future Work

- Semi-Supervised methods give promising result for Hindi Language, given that it is a morphologically rich language and has its own challenges in sentiment analysis.
- In the paper we referred [1], they report 71% accuracy using DBN on English Language and 76% using active deep learning. And we are able to achieve a maximum of 64% accuracy on Hindi language using DBN.
- Since DBN shows good performance with very little annotated data, this semi-supervised approach can be be easily and quickly set up for a different domain (such as product reviews) or even another low resource language (such as Bangla, Tamil, …)
- We could further explore the area of active deep learning where after the semi-supervised step we get a list of critical unlabeled reviews which can be labeled to get better accuracy.

- We could expand the semi-supervised approach to handle negation rules, which is not possible by our current unigram count model. For examples, the following sentence गानों में मौलिकता **नहीं** है is incorrectly marked as positive, and the sentence उनकी पकड़ कहीं भी कमजोर **नहीं** होती is incorrectly marked as negative by our tool since the word **नहीं** negates the polarity.

Acknowledgement

We thank Prof. Amitabha Mukherjee for his valuable support throughout the project and guiding us when required.

References

[1] Shusen Zhou, Qingcai Chen, and Xiaolong Wang. Active deep networks for semi-supervised sentiment classification. *Coling,* 2010.
[2] Piyush Arora, Sentiment Analysis For Hindi Language, *MS Thesis IIIT-H,* 2013.
[3] D. Das and S. Bandyopadhyay. Labeling emotion in bengali blog corpus a fine grained tagging at sentence level. In *Proceedings of the Eighth Workshop on Asian Language Resources*, pages 47–55, Beijing, China, August 2010. Coling 2010 Organizing Committee.