Text Coherence Analysis for Hindi Language

Aayush Patni **Umang** **Utkarsh Jain** **Sanchit**
Indian Institute of Technology Guwahati
Group No: 7
{patni170101001, umang170101074, jain170101075, sanch170101060}@iitg.ac.in

Abstract

In this project we look into the problem of text coherence in Hindi Language. Though the text coherence problem has been investigated thoroughly for English language, it has been unexplored in the domain of Hindi language. As such we try to come up with a neural network model to assess text coherence in an input Hindi text.

1 Introduction

Coherence in linguistics is what makes a text semantically meaningful. It is especially dealt with in text linguistics. Coherence is achieved through syntactical features such as the use of deictic, anaphoric and cataphoric elements or a logical tense structure, as well as presuppositions and implications connected to general world knowledge. Coherent text appears to be logically and semantically consistent for the reader-hearer. Our text analysis focusing on coherence is primarily concerned with the configuration of sense in the text i.e. how its single constituents are connected so that the text becomes meaningful for the addressee rather than being a random sequence of unrelated sentences and clauses. It evaluates the degree of logical consistency for text and can help document a set of sentences into a logically consistent order, which is at the core of many text-synthesis tasks such as text generation and multi-document summarizing.

Table 1 shows an example for text coherence problem.

Text 1	Text 2
मुझे किताबें पढ़ना पसंद हैं। मैं लाइब्रेरी में किताबें पढ़ना पसंद करता हूं। इसलिए मैं हमेशा लाइब्रेरी जाता हूं।	मुझे किताबें पढ़ना पसंद हैं। आज मेरा लंच छूट गया । इसलिए मैं हमेशा लाइब्रेरी जाता हूं।
label=1 (coherent)	label=0 (incoherent)

Table 1: Text coherence example

Although coherence is significant in constructing a meaningful and logical multi-sentence text, it is difficult to capture and measure as the concept of coherence is too abstract. The problem of coherence assessment was first proposed in 1980s, and since then a variety of coherence analysis methods have been developed as in (Li and Hovy, 2014), (Barzilay and Lapata, 2008), (Louis and Nenkova, 2012) and (Cui B.). However all the research into this topic has been mostly constrained to English language with no significant efforts in Hindi language.

Broadly the goal of the project can divided in two categories:

- Create a labelled data set in Hindi language for training of models.
- Try to come up with a novel solution based on deep learning for analysing text coherence on Hindi text and analyse its performance

2 Method

In this section, we discuss how we propose to achieve different goals of the project. The lack of text coherence research in Hindi domain persists as a major challenge to our work as there is very little background work. We have generated our own extensive data set for training as well as leveraged some other existing data sets which we found suitable for us. Our Text Coherence Analysis

Model is mainly inspired by previous works done in English domain. Here we discuss in detail our methodology for each of the goals.

2.1 Data set Generation

For our project we require a labelled data set of coherent and incoherent Hindi text to train our model. We are restricting our sample space to a single paragraph of text consisting of few sentences. We have combined two corpora for our model training. Our first corpus is custom generated. It consists of collection of news articles from Doordarshan news archive. This data is obtained using web scrapping of the dd-news website. Our second corpus is leverages an existing data set consisting of Hindi Wikipedia articles, available on Kaggle. For both data set we have applied relevant size restrictions so that its is neither too long nor short.

The underlying assumption is that the original sentence ordering in the source document is coherent. Hence each of the original documents are labeled as coherent text. To generate incoherent dataset, for each document, we split the document into its constituent sentences and generate a new document which consist of a random permutation of original sentences. We assume that such random permutation destroys the coherence structure, hence the new document is labelled incoherent.

2.2 Text Coherence Analysis Model

We present a novel deep coherence model(DCM) based on convolutional neural networks to learn coherence for the given text. We study the text coherence problem with a new perspective of learning sentence distributional representation and text coherence modeling simultaneously. In particular, word embeddings are first explored to generate sentence matrix for each sentence, and then sentence models map sentences to distributional vectors in parallel, which are used for learning coherence between them. Further, interactions between sentences are captured by computing the similarities of their distributional representations. Finally, the sentence vectors and their corresponding similarity scores are concatenated together to estimate the text coherence.

The model divides the document into paragraphs called cliques of n-sentences. Cliques of the original document are coherent; labeled 1. It then generates **change accordingly** 3 permutations of the document, where cliques form the permuted documents are believed to be non-coherent; labeled 0. The document's coherence score is hence the product of all of its cliques. Figure 1 shows how a 3-clique model would look like.

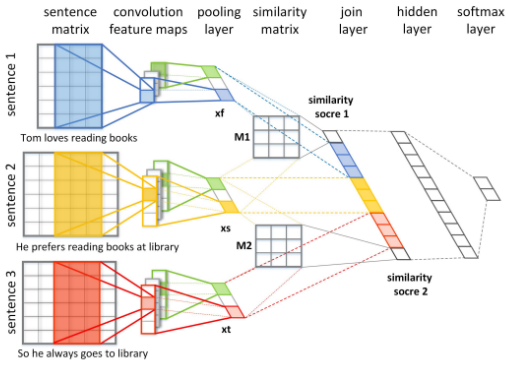


Figure 1: CNN model for text coherence measurement

The convolutional layer maps the sentence into an advanced representation where fine-grain features are captured, then the similarity is computed according to the equation shown below, after which the similarity matrices are learned as well.

$$Sim(fs) = x_f^T M_1 x_s$$

The similarity equation measures the similarity between the first sentence of the clique (x_f), and the second sentence (x_s), using trained similarity matrix (M_1). The same equation is applied to second and third sentences, (x_s) and (x_t). The similarity between the first and the third sentences is inferred from the first two scores since it is

a transitive relation, so there is no need to calculate it.

Finally, the dense layers are responsible for determining whether the sentence is coherent or not.

3 Implementation Details

3.1 Dataset Generation

All possible non-overlapping cliques of 3 sentences were generated from each coherent document. These cliques were stored with label 1 indicating coherent text. Each document was shuffled 10 times and again all possible cliques were generated. These cliques were labelled 0 indicating incoherent data.

3.2 CNN Model

The generated dataset was passed as input through a 100 dimensional Embedding layer. We have used pre-trained Fasttext word embeddings and thus this layer is set to be not trainable. After that a 1-D convolution was applied with 100 filters of width 3 and MaxPooling was applied to it. Once a vector representation of each sentence in clique was generated, this was passed to a 100×100 Similarity matrix to get the similarity scores of adjacent sentences in each clique. Then these scores were passed to a MLP model to get the final class output. Adam was used as optimizer and negative log likelihood loss function was used to train the model for 20 epochs. Further 5-fold cross validation was used along with ROC score as a metric to evaluate the model's performance. The whole architecture was coded using Tensorflow and Keras libraries.

4 Results

The first segment of the project was labelled dataset generation. Raw data that has been collected consists of two types of articles, news articles and Wikipedia articles which adds up to more than thirty thousand documents. News articles have been collected from the DD news website using web scraping. Wikipedia article dataset is obtained from kaggle dataset (Gaurav (2020)) of clean articles.

The coherent and incoherent datasets are generated from raw data. The underlying assumption is that the articles in raw data are coherent. To generate the incoherent data, sentences in the articles of raw dataset are randomly shuffled.

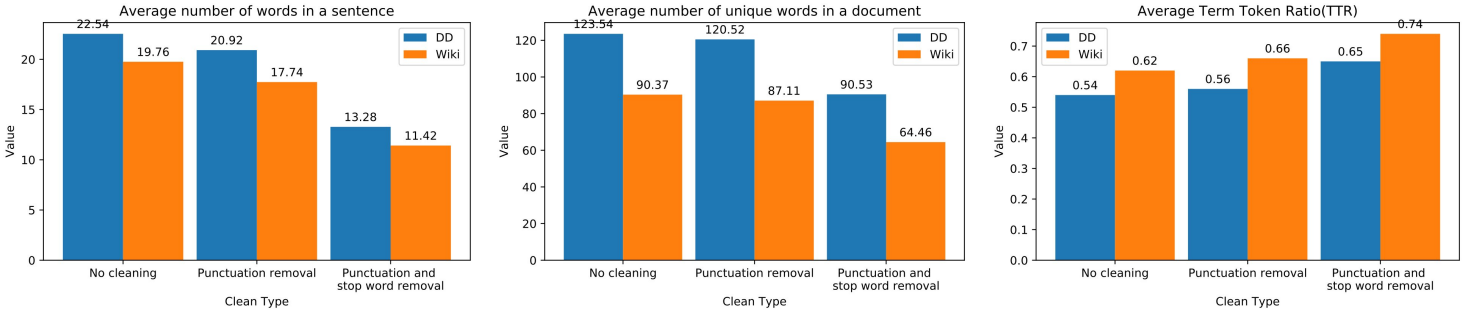
Following this, data was analyzed to know basic properties like number of sentences, number of words etc. of the generated dataset. Analysis was done thrice with different types of preprocessing and separately for News and Wikipedia articles. In first case no preprocessing was done and thus punctuation marks were considered as tokens in this case. In second case, punctuation marks were removed from the data as they are less relevant to text coherence. Finally all the words in the data were stemmed and stop words were removed.

Figure 2 shows the comparison of average number of words in a sentence, average number of unique words in a document and average term token ratio (TTR) for DD news articles(DD in graph) and Wikipedia articles(Wiki in graph) and for different types of data cleaning. Table 2 shows the summary of statistics for both the datasets.

The next phase was training the discussed model. We have used Area Under Cover(AUC) for the Receiver Operating Characteristics(ROC) curve evaluation metric for checking the model's performance. It tells how much model is capable of distinguishing between classes. The model was trained on two varaints of the dataset. Dataset-I consists of equal

Data set	DD News Articles	Wikipedia Articles
No. of documents	14162	18099
No. of sentences	158945	145935
No. of words	3581832	2883057

Table 2: Data set statistics



(a) (b) (c)

Figure 2: Comparison plots for a)Average number of words in a sentence, b) Average number of unique words in a document and c) Average term token ratio per document, for DD news and Wikipedia articles and different data cleaning types.

number of training examples belonging to both classes which we call balanced data. Dataset-II is an imbalanced dataset with ratio of 1 : 10 between coherent and incoherent classes. Oversampling technique was used to make Dataset-II. Model trained on second dataset outperformed those trained on the first dataset as shown by the ROC plots(Figure 3 and Figure 4).

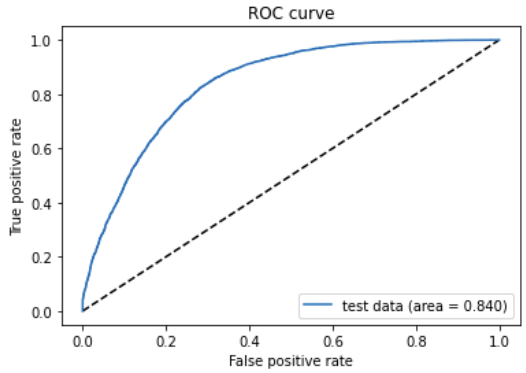


Figure 3: ROC Curve for balanced data

Higher AUC score indicates the model is better at distinguishing classes. AUC score for balanced data is on average 0.840 whereas for imbalanced data using oversampling it is 0.906. Accuracy for test set in each fold can be seen in Table 3.

5 Conclusion

In this project, we have tried to address the problem of coherence in Hindi Language. We are doing it in two phases - creation of labelled data sets in Hindi and proposal

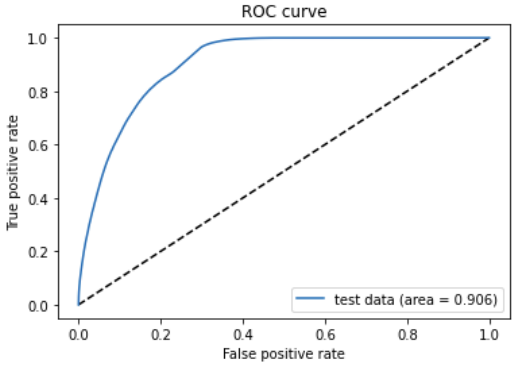


Figure 4: ROC Curve for unbalanced data with oversampling

Fold Number	Balanced Data	Unbalanced Data
1	77.72	82.61
2	78.02	82.89
3	77.52	83.23
4	76.89	83.01
5	77.83	82.76

Table 3: Test accuracy for balanced and unbalanced data in 5-fold cross validation

of a solution based on deep learning for analysing text coherence on Hindi text with its performance analysis. We have collected coherent dataset from DD news archive and Hindi Wikipedia articles and generated incoherent dataset by doing random permutation of sentences in mentioned coherent dataset. We have also performed basic analysis(number of sentences, number of words etc.) on our dataset. Following this, we trained a deep coherence model based on convolutional neural networks for text coherence analysis. The model tries to learn

the text coherence for cliques from the sentence ordering and their similarities. Text coherence on the basis of sentence ordering is a relatively new way for text coherence analysis.

6 Future Work

There is a need for more benchmark datasets for Hindi language so that the model can be trained better. Further we observe that increasing data using oversampling improves the accuracy. But due to limited computational power, we could not train the model with more data than the one for which results have been presented in this report. Our model learns text coherence from the ordering of sentences. Hindi text coherence can explored using some other approaches such as entity based coherence approaches, centering framework theory, etc. as discussed by (Abdolahi and Zahedi). The use of network architecture search to find the best architecture for MLP model added after the similarity calculation can also be investigated. In this model, pre-trained embeddings have been used. Training embeddings from scratch might give better results because in that case out of vocabulary (OOV) words will be taken care of.

References

Mohamad Abdolahi and Morteza Zahedi. *An overview on text coherence methods.*

Regina Barzilay and Mirella Lapata. 2008. Entity grid. *Modeling Local Coherence: An EntityBased Approach.*

Wikipedia contributors. 2020. Coherence (linguistics). *Wikipedia, The Free Encyclopedia.*

Zhang Z Cui B., Li. Y. Dcm. *Text coherence analysis based on deep neural network.*

Gaurav. 2020. Hindi wikipedia articles-55k. *Kaggle.*

Jiwei Li and Eduard H. Hovy. 2014. Recursive and recurrent nns. *A Model of Coherence Based on Distributed Sentence Representation.*

Annie Louis and Ani Nenkova. 2012. Hmm. *A Coherence Model Based on Syntactic Patterns.*

.

Note: Code base couldn't be uploaded due to size limitations. The whole code base can be found at github repository: `https://github.com/umangk279/NLP-Hindi-Text-Coherence.git`