Extractive Text Summarisation in Hindi

Sakshee Vijay, Vartika Rai*, Sorabh Gupta, Anshuman Vijayvargia & Dipti Misra Sharma*

Kohli Center on Intelligent Systems (KCIS),
International Institute of Information Technology, Hyderabad
{sakshee.vijay, vartika.rai}@research.iiit.ac.in,
{audace.sourabh, avijayvargia}@gmail.com & dipti@iiit.ac.in

Abstract

With immense amount of data growing on web in Hindi, a text summariser would be helpful in summarising Government data, medical reports, news, and research articles. Hindi is the fourth most-spoken first language in the world. Hindi written in the Devanagari script is the official language of the Government of India. There is no public dataset for extractive summarisation available in Hindi and thus a dataset of 24253 News articles was extracted and the extractive summaries results were evaluated on various parameters with manual gold summaries of exactly 60 words each.

Index Terms: Dataset, Text Summarisation, Extractive, Hindi

1. Introduction

Extractive Text summarisation is a task of selecting few phrases which hold important information and discarding redundant information from a textual document or article. An automatic summarisation tool is required because of immense growth in the amount of data available on the World Wide Web(www). The need of automated summary in many domains such as news article summary, document summary, reviews summary, government documents summary, medical reports summary is growing and need attention in Hindi. The main aim of a text summariser is to reduce the reading time. A lot of relevant work has been done in other languages where the processing of text and respective tools are already present.

The correct use of heuristics and features yield better results in extractive summarisation. Hindi being a free word order language makes it difficult to choose linguistic features based on structure of sentences. So statistical and linguistic features are used to get extractive summary of news articles.

The rest of the paper is organized as follows: Section 2 presents a brief Literature review of the text summarization task, in Section 3 we describe our dataset, in Section 4 in detail our Methodology, discussing the features and strategy to get extractive summary in Hindi, in Section 5 we relate the computational results obtained, and in Section 6 the application of our proposal to a reference document collection and finally, in Section 7, we present some conclusions and outline some future research work, that can be done later.

2. Literature Survey

The earliest work in summarisation was done by Luhn and its also the most cited work in summarisation where he proposed that the frequency of a particular word in an article provides an useful measure of its significance [3]. As a first step, words were stemmed to their root forms, and stop words were deleted. Luhn then compiled a list of content words sorted by decreasing frequency, the index providing a significance measure of the word. On a sentence level, a significance factor was derived that reflects the number of occurrences of significant words within a sentence, and the linear distance between them due to the intervention of non-significant words. Later on sentence position was also added as an feature by Baxendale [8].

Lin and Hovy (1997) [9] studied the importance of a single feature, sentence position. Just weighing a sentence by its position in text, which the authors term as the position method, arises from the idea that texts generally follow a predictable discourse structure, and that the sentences of greater topic centrality tend to occur in certain specifiable locations (e.g. title, abstracts, etc). However, since the discourse structure significantly varies over domains, the position method cannot be defined as naively as in Baxendale [8].

Thaokar [6] was one of the first one to work on Hindi dataset and proposed an approach for summarizing Hindi text document using semantic graph and particle swarm optimization algorithm. It uses Hindi Wordnet to tag appropriate POS of word for checking SOV of the sentences which uses six statistical and two linguistic features and uses genetic algorithm to optimize the summary generated based on the text feature terms with less redundancy, with F1 score 50.01, which was very less with respect to the amount of data given.

[1] proposed three major algorithms, fuzzy classifier, neural network and global search optimization (GSO). The fuzzy classifier and neural network are used for generating sentence score. The GSO algorithm is used with the neural network, in order to optimize the weights in the neural network. A hybrid score is generated from fuzzy method and neural network for each input sentences. Finally, based on the hybrid score from fuzzy classifier and neural network, the summary of the given input records are generated. The approach achieved an average precision rate 0.90 and average recall rate of 0.88 for compression rate 20%. Many simple rule based approaches are shown by [2] where the accuracy is shown as 91%, which is not the baseline as the dataset is quiet small.

3. Dataset

The data is taken from website which has currently humans summarising news in 60 words manually. These Language Specialists are the native speakers of Hindi, thus provide us with the gold data. The link to the website is https://www.inshorts.com/hi/read. Articles for every category (sports, politics, world, entertainment, etc,.) are extracted along with their gold summary, with a total of 24253 article summary pair,as shown in the Table 1. Each summary is of con-

* Equal Contribution

978-1-5386-1981-0/17/$31.00

sistent 60 words length. Such a huge corpus of News articles in Hindi along with their 60 words gold human extracted summary doesn't exist for Hindi. Hence the dataset is of high importance and one of its kind.

The data was cleaned after extraction in the following way:

1. Majority of the articles were written in English summarised in Hindi. These articles were ignored, not taken into consideration.
2. The Hindi summary referring to an article in form video or any other non-textual description were ignored. Even tweets as the full article were ignored.
3. All summaries were manually written, so while comparing it with the main article, usually the words did not match, because human annotators use a similar word from the hindi dictionary.
4. Articles containing English words were not taken, but articles containing words like "computer" in english were kept as this is a commonly used term in hindi as well.
5. External links to other news articles were removed.
6. Author details were cleared from the article.

Categories	Number of Articles
World	2678
Technology	560
Startup	45
Sports	1900
Science	85
Politics	5623
National	6799
Miscellaneous	1009
Different(Hatke)	405
Entertainment	2905
Business	1987
Automobile	257
Total	**24253**

Table 1: Categories of Articles

4. Methodology

Features to select these important informative structures can be of two categories, statistical based on the frequency of some elements in the text, and linguistic extracted from a simplified argumentative structure of the text.

There are three main steps to summarise the text : preprocessing, processing and postprocessing. In preprocessing, the article is brought to a structural form. In processing, the main summary is generated. In post processing, summary is formatted in a readable and precise form from the processed data. The final summary is provided to the user after these three steps.

The amount of data to be considered as meaningful and to be taken in phrases of extractive summarisation is reduced in the first preprocessing step, and thereby the dimensionality of article is reduced by eliminating stop words and case folding, followed by stemming. Stop words are words which do not add to the semantics of article and usually words like preposition. But stop words like pronouns, referring to other nouns are kept. Pronouns usually refer to some important noun of an article. Case folding includes converting all words to same kind of letter case. Stemming gives the root form of words, giving syntactically similar words. Processing step includes Word Level and Sentence Level Features.

1. **Word Level Features**
   (a) *Frequency based (tf-idf)*: As shown intially by [3] the most important and frequently used measure is term frequency and inverse document frequency. In text summarization we can employ the same idea: in this case we have a single document d, and we have to select a set of relevant sentences to be included in the extractive summary out of all sentences in d. The use of tf-idf thus can be seen for single document as well. Term frequency is calculated for both unigrams and bigrams. While calculating bigram frequency, we only calculated the bigrams for proper nouns. The bigram frequency is lower than the unigram frequency so we applied normalizations to bigram frequency with scale to unigram frequency in order to use it as a word level feature.
   (b) *Length of word*: The frequency of smaller words would be larger than longer words. In order to negate this occurrence, we use the length of word.
   (c) *Occurrence in heading of Articles*: More weightage is given to words occurring in headings as compared to other words, as those are important.
   All the above weights are calculated and normalised to a scale of 0-1.
2. **Sentence Level Features**
   (a) *Sentence Length*: The length of sentence varies a lot within an article and thus, ratio of the number of words occurring in the sentence over the number of words occurring in the longest sentence of the document.
   (b) *Sentence position* within an article and in paragraphs as well. We use here the percentile of the sentence position in the document, as proposed by [7]; the final value is normalized to take on values between 0 and 1.
   (c) *Presence of verb in a Sentence*: If a sentence is complete, it does contain an axillary verb or a main verb in a sentence, which helps in measuring importance of a sentence.
   (d) *Similarity to headline of article*: The cosine similarity of every sentence with the headline is calculated and taken into consideration.
   (e) *Referring pronouns*: While removal of stop words, we usually ignore the pronouns. In order to get actual score of a sentence, the proper nouns to which the pronouns are referred should be considered.
   (f) *Cohesion similarity score of a sentence*: Various sentences and their similarity with each other are considered. A similarity score of sentences is computed in the following way : The summation of similarity score s with every other sentence in the document is considered and in the end, the sentence with maximum cohesion similarity score in the document.

(g) *Sentence-to-Centroid Cohesion*: This feature is obtained for a sentences As follows: first, we compute the vector representing the centroid of the document, which is the arithmetic average over the corresponding coordinate values of all the sentences of the document; then we compute the similarity between the centroid and each sentence, obtaining the raw value of this feature for each sentence. The normalized value in the range [0, 1] for s is obtained by computing the ratio of the raw feature value over the largest raw feature value among all sentences in the document. Sentences with feature values closer to 1.0 have a larger degree of cohesion with respect to the centroid of the document, and so are supposed to better represent the basic ideas of the document.

5. Results

Experiment Results are of the form :

Precision = No. of summary sentences extracted matching to human summary / Total No. of sentences in extracted summary multiplied by 100%. Precision is also called accuracy of the system, which indicates how good a system is to select sentence as a summary sentence.

Recall = No. of summary sentences extracted matching to human summary / Total no of sentences in human summary multiplied by 100%. Recall represents completeness of a sentence. A recall value of 1 indicates that all the actual summary sentences in the document are selected by the system. It is also called sensitivity of the system.

$$F_1 score = \frac{2(precision * recall)}{(precision + recall)} \quad (1)$$

And $GScore$, the Geometric mean of precision and recall is calculated as,

$$Gscore = \sqrt{precision * recall} \quad (2)$$

Performance	Score
Recall	70
Precision	62

Table 2: Results.

Article length	Summary Length
1076 words(maximum)	60
150 words(average)	60

Table 3: Compression Ratio.

Rather than the compression ratio being a fixed variable, every summary length was fixed to 60 words, for varying article length.

6. Application with Sentiment Analysis

Text summarisation can be used to determine the sentiment of an article in less time effectively. Since the summaries generated are more shorter and precise version of the article, it can be used to get the sentiment of the article in less time. In future work, the model would be to test rigorously what is the accuracy of sentiment of the article and summary generated respectively, depicting how similar they are. Text summarisation in Hindi is not available for free online. It will hosted online for public use as well as along with the sentiment analyser, for the varied uses with the increase in the amount of online content in Hindi.

7. Conclusion and Future work

In Hindi, a proper baseline for the task of extractive summarisation was never created for a huge dataset of 24253 news articles. With this dataset and baseline, we would now apply machine learning techniques for extractive text summarisation. For Abstractive text summarization, which is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage and increases the readability of summary, we would train a neural network on all news articles and also individually on various category of news which have high number of articles. Both model results will be compared for further analysis. Abstractive Summarisation in hindi using Rich semantic graph is proposed by Subramaniam [5]. Abstractive text summarisation using sequence to sequence RNNs as proposed by Nallapati [4] is proposed for hindi as well. But no results are provided. The analysis of results with correct accuracy and precision with our model would be calculated, along with the comparision with above two models.

8. References

[1] Anitha, J and Prasad Reddy, PVGD and Prasad Babu, MS. An Approach for Summarizing Hindi Text Through a Hybrid Fuzzy Neural Network Algorithm. *Journal of Information & Knowledge Management*, 13(04):1450036. 2014. World Scientific.

[2] Kaur, Dawinder and Kaur, Rajbhupinder Automatic Summarization of Text Documents Written in Hindi Language 2014.

[3] Luhn, Hans Peter. The automatic creation of literature abstracts *IBM Journal of research and development*, 2(2)159–165. 1958. IBM.

[4] Nallapati, Ramesh and Zhou, Bowen and Gulcehre, Caglar and Xiang, Bing and others. Abstractive text summarization using sequence-to-sequence rnns and beyond *arXiv preprint arXiv:1602.06023*, 2016.

[5] Subramaniam, Manjula and Dalal, Vipul. Test Model for Rich Semantic Graph Representation for Hindi Text using Abstractive Method. 2015.

[6] Chetana Thaokar and Latesh Malik. Test model for summarizing hindi text using extraction method. *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, 1138–1143. 2013.

[7] Witten, Ian H and Paynter, Gordon W and Frank, Eibe and Gutwin, Carl and Nevill-Manning, Craig G KEA: Practical automatic keyphrase extraction. *Proceedings of the fourth ACM conference on Digital libraries*, 254–255. 1999. ACM.

[8] Baxendale P Machine-made index for technical literature - an experiment. *IBM Journal of Research Development*, 2(4):354361. 1958.

[9] Hovy, E. and Lin, C. Y. Advances in Automatic Text Summarization *Mani, I. and Maybury, M. T., editors*, 81–94. 1999. MIT Press.

[10] Dalal, Vipul and Malik, Latesh G. A survey of extractive and abstractive text summarization techniques *Emerging Trends in Engineering and Technology (ICETET), 2013 6th International Conference on*, 109–110. 2013. IEEE.

[11] Edmundson, Harold P. New methods in automatic extracting *Journal of the ACM (JACM*, 16(2):264–285. 1969. ACM.

[12] Gaikwad, Deepali K and Mahender, C Namrata. A Review Paper on Text Summarization *International Journal of Advanced Research in Computer and Communication Engineering*, 5(3). 2016. ACM.

[13] Neto, Joel Larocca and Freitas, Alex A and Kaestner, Celso AA. Automatic text summarization using a machine learning approach *Brazilian Symposium on Artificial Intelligence*, 205–21. 2002. Springer.