TL;DR: Summarizing News Articles in Hindi Language

Saurabh Gupta, Neha Kumari
IIIT-Delhi
{saurabhg, nehak}@iiitd.ac.in

Abstract

The number of Hindi language readers on the internet is continuously increasing and is expected to rise to 38% of the Indian internet user base by 2021. This rise will result in an increase of Hindi content providers, especially news articles, and therefore we need to provide more information in lesser words. In this work, we present TL;DR, a summarization dataset of 19,912 Hindi news articles and their summaries written by authors and editors of Inshorts and Shortpedia. Scraped from Inshorts and Shortpedia webapps, these summaries are curated using extractive and mixed strategies, picking up words and phrases from articles at different rates. We perform the state-of-the-art extractive summarization strategies over this dataset and use the automatic summaries to compare with the manual summaries using different ROUGE evaluation metrics.

1 Introduction

According to a study conducted by KPMG India and Google[1], nearly 70% Indians believe that digital content available to them in their local language is more reliable. Therefore, online users who know any local language prefer to read the news available in the local language. Due to this preference bias, the content in local language and the local language users are increasing. The report also suggests that by 2021, the Indian internet user base will rise to 38% Hindi language users online. This rise will make Hindi the most popular digital language in India.

A lot of English news summarization techniques exist. However, not a lot of work has been done on summarization of local languages. For Hindi news article summaries, organizations like Inshorts[2] and Shortpedia[3] have large content teams who manually curate the summaries to serve their users. Therefore, we have used these two sources to collect the Hindi news articles and their manually curated summaries. The reason to choose them is their increasing popularity with high ratings and large number of downloads as shown in Table 1. Despite being launched later, they have a better rating than the traditional Hindi news provider apps like Dainik Jagran. Therefore, we can safely assume that people do prefer reading summaries of news articles in regional language. They are interested in using the news summary provider apps more which is evident from their App ratings in Table 1. The solution to the problem of Hindi text summarization can save time and money used for manual curation by content curators who handpick the words to add to the summaries.

In this work, we collect and release a benchmark dataset with ground truth summaries and news articles in the Hindi language. We plan to make this dataset of 19,912 news articles publicly available upon publication of the work to aid further research. We conduct experiments with three extractive text summarization techniques and compare the system generated summaries to the manually curated ground truth summaries to establish a testbed for Hindi text summarization task.

2 Background

Automatic summarization has been studied since the late 1950s, with the focus on finding ways to rank the sentences in a document by scoring them on the basis of their content. Beginning with Luhn (1958) who used statistical information like word frequency and distribution to compute a rela-

[1] https://qz.com/india/972844/indias-internet-users-have-more-faith-in-content-thats-not-in-english-study-says

[2] Summaries in 60 words, http://inshorts.com

[3] Summaries in 70 words, http://shortpedia.in

Application	Categories of news	Ratings	Downloads
InShorts (launched Sep, 2013)	12	4.6 (340k reviews)	5 M+
ShortPedia (launched Sep, 2017)	10	4.7 (172 reviews)	40k+
Dainik Jagran (launched Mar, 2012)	13	4.2 (88k reviews)	5 M+

Table 1: App statistics from Google Play Store

tive measure of significance to rank the sentences, Gambhir and Gupta (2017) explained many extractive summarization techniques such as graph theory based approaches, cluster-based methods, etc. The authors also talked about the features used in ranking sentences like the content word, cue phrase, biased word, noun, sentence location, etc.

There are some recent works (Patel et al., 2007; Gong and Liu, 2001) that suggest language independent and generic techniques to summarization. In (Patel et al., 2007), the authors created an enhanced feature vector to represent the sentences. They took into account the incomplete sentences and location features. Gong and Liu (2001) presented a similar approach and uses standard information retrieval (IR) methods to rank sentence relevance, but advanced to use latent semantic analysis technique to identify sentences that are semantically important for creating summaries. Ferreira et al. (2013) listed several methods for sentence scoring as well as evaluating the system generated summaries. They empirically proved that Textrank(Mihalcea and Tarau, 2004) is the best graph scoring algorithm for several datasets (CNN dataset, Blog Summarization dataset, SUMMAC dataset). In this work, we used graph-based methods as discussed by Mihalcea and Tarau (2004). In these methods, when a sentence refers to another, a link is created between them and a weight is assigned. The weights are further used to rank the sentences.

The works that study summarization of local languages (Thaokar and Malik, 2013; Gupta and Lehal, 2012) proposed methods to summarize Hindi and Punjabi languages respectively. The system proposed by Thaokar and Malik (2013) uses features like Average TF-IS[sentence]F, sentence length, sentence to sentence similarity, etc. followed by a genetic algorithm (Goldberg, 1989) which helps select the most relevant (highest ranked) sentences. Gupta and Lehal (2012) presented a similar approach where the final sentences were determined by a feature-weight equation with top-ranked sentences selected as the summary. Gupta and Lehal (2012) show promising results for Punjabi language summarization. In another work (Gupta and Kaur, 2016), the authors tried to summarize the Punjabi text document in a hybrid based approach. The authors used manually converted Hindi unicode-based corpus as well as unicode punjabi newspaper AZIT.

3 Collecting News Articles and Summaries

The TL;DR dataset was collected from the official web-app of Inshorts and Shortpedia. To create the dataset, we performed a web-scale crawling of news articles and their corresponding summaries from different pages representing different categories like sports, politics, business, etc. The crawling process respected the rules mentioned in the robots.txt for each website. We observed the collected summaries to find out that they are either extractive or mixed in nature. However, there is no meta-data that clearly states the nature of a particular summary. The extractive/mixed nature of summaries played a vital role in selecting the existing systems as discussed in Section 4.

3.1 Content Scraping

To crawl the news articles and their corresponding summaries, we created a list of all categories that are available on Inshorts and Shortpedia. Each category has its own web-page on the web-app. Each page consists of a lot of news titles and their summaries. We used a headless browser, phantomjs[4], to simulate a browser like behavior to load more news titles and their respective summaries. We used Beautiful Soup[5] to extract HTML body

[4] http://phantomjs.org/

[5] https://www.crummy.com/software/BeautifulSoup/

	Inshorts	Shortpedia
Number of dataset instances used for the experiment	14,649	5,263
Number of sentences in summaries	3-8	3-9
Number of words in summaries	50-70	60-110
Number of sentences in news articles	12-465	10-180
Number of words in news articles	330-10,593	92-4,944
Avg. number of sentences in summaries	3.6 ± 0.5	2.5 ± 2.2
Avg. number of words in summaries	60 ± 1	85 ± 7
Avg. number of sentences in news articles	8 ± 13	20 ± 12
Avg. number of words in news articles	425 ± 495	400 ± 323

Table 2: Dataset Statistics

content. Beautiful Soup allowed us to navigate and search through the HTML to extract the titles and the summaries directly from the content; and a source URL to fetch the actual news. Using the source URL and the Newspaper3k[6] module, we fetched the text from the original news article. The data are collected over a period of 4 months.

3.2 Building the Datasets

We created a dataset of Hindi news articles and their manually curated summaries from Inshorts and Shortpedia. We collected over 17,549 and 11,896 news articles having Hindi summaries of length between 50-70 and 60-110 words for InShorts and Shortpedia respectively. We discovered two categories of instances in the dataset. First, where the news article is in English and the summary is in Hindi. Second, where both the news articles and summaries are in Hindi. We filtered the latter instances, where both the news article and summary is in Hindi, using language detection from text[7] to perform extractive text summarization. We performed the experiments on 14,649 and 5,263 news articles from Inshorts and Shortpedia respectively. More information about news articles and their summaries in the dataset are given in Table 2.

4 Performance on Existing Methods

Empirically, Ferreira et al. (2013) proves that the Graph based techniques like Textrank, Lexrank performs better in the task of extractive summarization. Given the extractive and mixed nature of the collected manual summaries, we generated the summaries using the following methods:

Lead3 A very commonly used automatic text summarization strategy is to pick up the first k sentences and treat them as a summary. As shown in Table 2, the average number of sentences in summaries is 2.5 and 3.6. Therefore, we choose k as the mean of these two values, i.e., 3 as the average manual summary over both datasets is also three sentences long.

TextRank TextRank(Mihalcea and Tarau, 2004) is a graph-based ranking model for automatic text summarization. The algorithm first identifies text units that first define the text at hand and adds them as vertices in the graph. In the next step, the relations that connect these text units are identified, and edges are drawn between them. This process is iterated till convergence, and the vertices are sorted based on their final scores. The values attached to each sentence are used for ranking or selection decision. We have used the Gensim implementation of TextRank(Rehurek and Sojka, 2010) for our work.

LexRank LexRank(Erkan and Radev, 2004) is a stochastic graph-based method where relative importance of the textual units are computed for Text Summarization. This method first identifies the most important sentences in the document based on eigenvector centrality in a graph representation of sentences. The results of this approach have shown that degree-based methods outperform centroid-based methods. We have taken an implemention[8] of this method and modified it to make it work with the Hindi news articles.

[6] https://newspaper.readthedocs.io/en/latest/

[7] We have used the lang detect package from PYPI (https://pypi.org/project/langdetect/)

[8] https://github.com/siddrtm/Document-Summarization

Table 3: The average F1-score of ROUGE-1, ROUGE-2, and ROUGE-L metrics for the chosen methods: Lead3, Textrank and Lexrank on the two datasets, Inshorts and Shortpedia.

Dataset	Evaluation	Lead3	TextRank	LexRank
Inshorts	Rouge - 1	0.11	0.31	**0.33**
	Rouge - 2	0.05	0.10	**0.10**
	Rouge - L	0.08	**0.23**	0.19
Shortpedia	Rouge - 1	0.33	0.35	**0.35**
	Rouge - 2	0.14	0.13	**0.16**
	Rouge - L	0.24	**0.28**	0.25

5 Evaluation

We employed ROUGE-1, ROUGE-2 versions of ROUGE-N (N stands for n-grams) and ROUGE-L (L stands for Longest Common Subsequence) (Lin, 2004) as our quantitative evaluation metrics. ROUGE-N calculates the number of n-grams overlapping between the system summary and the manual summary. We selected ROUGE-1/ROUGE-2 to see how many lexical units/pairs of lexical units overlap to see whether the methods defined in Section 4 were able to capture the important lexical units from the news articles. ROUGE-L computes lexical units that overlap to measure the quality of summaries. We chose ROUGE-L as another evaluation metric because the techniques we were using involve extracting the sentences and sorting them based on their position in the document to create a summary. We did not impose a maximum limit on the length of summaries we generate. ROUGE-L do not require predefined n-gram length and naturally captures the sentence level structure. We used F1-score variants of ROUGE-1, ROUGE-2 and ROUGE-L because we do not impose a word limit to the summaries, and want the scores to account for different summary lengths. We use the default configuration of the Lin (2004) ROUGEv1.5.5 implementation. Source article text and manual summaries for all methods are tokenized using the corpus reader available with NLTK's Indian module[9].

6 Results

Table 3 shows results for the summarization methods discussed in Section 4 on Inshorts and Shortpedia news articles. Lower average F1-score of LEAD3 for all metrics with Inshorts articles show that the first three sentences do not contain the most relevant information. However, this is not the case with the news articles collected from Shortpedia.

The average F1-score of ROUGE-1 or unigram overlap is maximum for both Inshorts and Shortpedia in the case of Lexrank because it takes into account the relative importance of textual units. The method is scoring sentences and extracting the ones with the most important lexical units. Similar pattern is seen with average F1-score of ROUGE-2 or the bi-gram overlap as well. However, the score in case of bi-gram overlap is lower as compared to the score in case of unigram overlap. This is happening because the automatic summaries are extractive while the manual summaries are both extractive and mixed in nature.

As evident from Table 3, for all cases, Textrank method produces the most successful summaries with higher ROUGE-L scores. The averge F1-score of ROUGE-L is best with Textrank for both Inshorts and Shortpedia. Textrank method ranks sentences based on their linkability and is able to capture the sentence level structure outperforming the other chosen methods by a small margin.

7 Conclusion and Future Work

We have introduced a new dataset of 19,912 Hindi news articles and their corresponding manual summaries for Hindi text summarization task. We also experimented with state-of-the-art graph based techniques used for extractive summarization. The dataset and the empirical results with ROUGE scores establishes a testbed for further research in Hindi text summarization task.

Generating abstractive summaries is a very complex task and requires a very large amount of data. At this point, we have limited the scope of this work to extractive summarization techniques. As we are collecting more data, the future directions include topic modelling and generation of abstractive summaries for Hindi news articles.

[9] https://www.nltk.org/_modules/nltk/corpus.html

References

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Frederico Luiz Gonalves de Freitas, George D. C. Cavalcanti, Rinaldo Lima, Steven J. Simske, and Luciano Favaro. 2013. Assessing sentence scoring techniques for extractive text summarization. *Expert Syst. Appl.*, 40:5755–5764.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.*, 47(1):1–66.

David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 19–25, New York, NY, USA. ACM.

Vishal Gupta and Narvinder Kaur. 2016. A novel hybrid text summarization system for punjabi text. *Cognitive Computation*, 8(2):261–277.

Vishal Gupta and Gurpreet Lehal. 2012. Automatic punjabi text extractive summarization system. In *Proceedings of COLING 2012: Demonstration Papers*, pages 191–198.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Alkesh Patel, Tanveer Siddiqui, and U. S. Tiwary. 2007. A language independent approach to multilingual text summarization. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 123–132, Paris, France, France. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

C. Thaokar and L. Malik. 2013. Test model for summarizing hindi text using extraction method. In *2013 IEEE Conference on Information Communication Technologies*, pages 1138–1143.