Hindi History Note Generation with Unsupervised Extractive Summarization

Aayush Shah[*], **Dhineshkumar Ramasubbu**[*], **Dhruv Mathew**[*], **Meet Chetan Gadoya**[*]
University of Southern California
Los Angeles, California
{aayushsh, dramasub, dkmathew, gadoya}@usc.edu

Abstract

In this work, the task of extractive single document summarization applied to an education setting to generate summaries of chapters from grade 10 Hindi history textbooks is undertaken. Unsupervised approaches to extract summaries are employed and evaluated. TextRank, LexRank, Luhn and KLSum are used to extract summaries. When evaluated intrinsically, Luhn and TextRank summaries have the highest ROUGE scores. When evaluated extrinsically, the effective measure of a summary in answering exam questions, TextRank summaries performs the best.

1 Introduction

Our task is to apply text summarization to generate notes for school students where the medium of instruction is Hindi. The motivation for this work is that students studying under the Indian Central Board of Secondary Education (CBSE) have a lack of additional resources given their medium of instruction. Online resources are limited, with most reference guide material being published in English. Given the vast quantities of information that students are made to memorize, we believe that our tool will help provide students with an outline, a text summary, that could serve as both a big picture introduction and a pre-exam study guide. We focus on this task as each year over 18 million students give the grade ten exam. As Hindi is a low resource language, we believe that such a tool could help students learn better.

From prior research (Verma et al., 2019) on comparative text summarization in English and Hindi, we see that summarization results vary drastically for different languages and subject matters. While extensive research has been done for summarization techniques in English (Luhn, 1958; Edmundson, 1969; Carbonell and Goldstein, 1998; Pal and Saha, 2014; Erkan and Radev, 2004), directly applying said methods to Hindi text performs poorly (Verma et al., 2019). Frequency, graph and feature based approaches have been investigated previously to extract summaries from Hindi text and have shown to perform well on news documents (Vijay et al., 2017). Rule based methods (Gupta and Garg, 2016), and improvements to graph based methods incorporating semantic information from the text (Kumar et al., 2015) perform well for Hindi documents from various domains.

We wish to address the task of extractive text summarization in Hindi as it applies to learning history in an education setting for school students using unsupervised algorithms. The main reason behind choosing unsupervised methods for this task is that these algorithms do not require a dedicated training set annotated by individuals with subject specific knowledge. Secondly, employing a supervised approach for a particular domain constrains the portability of the trained model to be applied on different domains. Furthermore, the efficiency or goodness of the generated summaries for a particular task rely on accurate and reliable human annotated summaries used for training.

To the best of our knowledge, there exists no work that addresses Hindi text summarization in the academic domain as a note generating tool for students. This made it difficult to compare our approaches with existing work that deals with different domains of text data. In this work, we investigate unsupervised graph, term frequency and probability based single document summarization methods. Our work will build on previous linguistic analyses (for instance, no direct way to identify proper nouns) in Hindi (Paul et al., 2013) to deal with the nuances of summarizing history written in Hindi (Garg et al., 2012). Our code is publicly available on GitHub[1].

[*] These authors contributed equally to this work.

[1] Code: https://github.com/dhineshkumar-r/Unsupervised-Extractive-Summarization-Hindi-Note-Generation

2 Materials

We used the grade 10 Hindi history textbook (NCERT, 2018-2019) prescribed by the CBSE and published by the National Council of Educational Research and Training (NCERT) as the dataset. The Textbook is available in PDF format and is about 200 pages in length. There are 8 chapters (articles) in the book. Each chapter contains around 400 sentences comprising about 18 words each. This amounts to approximately 7200 words per chapter. To evaluate generated summaries, reference summaries of length 75 sentences are manually created for each chapter using the exact sentences from the textbook (extractive summarization). The annotators drafting the reference summaries are proficient in Hindi, have studied history at a high school level and are familiar with the course content and exam structure. In order to perform an extrinsic evaluation, we considered questions from the three most recent exam papers, from 2017-2019, and their corresponding rubrics[2]. The exams contain 3 types of questions - very short answer questions (1 mark each), short answer questions (3 marks each) and long answer questions (5 marks each) requiring 1, 3 and 5 sentences from the text respectively. There are a total of 35 questions in these exam papers. A sample very short question from the 2017 examination is as follows:

'हिंद	स्वराज'	पुस्तक	के	लेखक	का	नाम	लिखिए।
‘hind	svaraaj’	pustak	ke	lekhak	ka	naam	likhie .

Name the writer of the book 'Hind Swaraj'.

A student’s response that would score one full point is as follows:

महात्मा	गाँधी	ने	'हिंद	स्वराज'	पुस्तक	लिखी ।
mahaatma	gaandhee	ne	‘hind	svaraaj’	pustak	likhee .

Mahatma Gandhi wrote the book 'Hind Swaraj'.

3 Methodology

The basic idea behind the task of extractive summarization is that individual sentences in the source document are scored and ranked to extract the top n sentences as a summary. In this work, four unsupervised methods are investigated to score and rank the input document sentences. The methods used here are KLSum (Aria and Vanderwende, 2009), Luhn Summarization (Luhn, 1958), TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004). The summaries generated by these methods are evaluated and compared against each other intrinsically and extrinsically. TextRank and LexRank are graph based approaches. KLSum utilizes a probabilistic approach. Luhn uses a naive ranking algorithm based on word significance.

3.1 KLSum

Kullback–Leibler summarization (KLSum) (Aria and Vanderwende, 2009) is a probabilistic take on the extractive summarization problem. The basic idea here is to extract a summary, a set of sentences from the source document, whose unigram distribution is as close to the unigram distribution of the source document as possible. The closeness between the source and summary document distributions is determined by the KL divergence (Kullback and Leibler., 1951) $KL(D||S)$, where $D(w)$ and $S(w)$ are the unigram distributions of the word w in the source D and summary S document respectively.

$$KL(D||S) = \sum_{w} D(w)(\log(D(w)) - \log(S(w)) \quad (1)$$

The empirical unigram distribution of a document is the term frequency of words in the given document which is computed as :

$$tf_{t,d} = \frac{\textit{\# of times the term t occurs in document d}}{\textit{Total \# of terms in document d}} \quad (2)$$

Here, $tf_{t,d}$ represents term t frequency in text document d . The term frequencies are smoothened to ensure non-zero values. Mathematically, the optimization problem is defined as below:

$$S^* = \min_{S:words(S) \leq L} KL(D||S) \quad (3)$$

Here, L is the maximum number of words in the summary S. Since optimizing the above objective is exponential in the number of sentences in the source document, a greedy approach is taken. Starting with an empty summary, the summary is extracted iteratively. At each iteration, the sentence which results in minimum $KL(D||S)$ is added to the summary until the intended number of sentences is reached.

3.2 Luhn Summarization

Luhn summarization (Luhn, 1958) is a simple and naive summarization algorithm where the relative significance of each sentence in the source document is considered for selection in the summary. The basic idea exploited in this method is that an

[2] NCERT Solutions: https://byjus.com/ncert-solutions/

author of a document writing about a concept tends to repeat the same words to represent a specific notion. When such significantly repeating words are positioned relatively closer in a document, within a sentence for example, the sentence as a whole becomes significant enough to be considered in a summary.

The relative significance of each sentence is captured with the number of significant words and their physical proximity within a sentence. Each sentence is grouped into clusters beginning and ending with significant words. These first and last significant words of clusters are significantly related if the physical distance between them, intervened by insignificant words, is under a threshold. If more than one such cluster is found in a sentence, the cluster with the highest significance factor is assigned to the sentence. The sentences are then ranked relative to the other to generate the summary. Numerically, a word is considered significant if its term frequency is more than a specified threshold. The significance factor of a cluster C in a sentence is computed as follows:

$$Significance(C) = \frac{\textit{\# of significant words in C}}{\textit{\# of words in C}} \tag{4}$$

3.3 TextRank

TextRank (Mihalcea and Tarau, 2004) is a graph based approach which scores sentences in the given document based on the PageRank (Page et al., 1999) algorithm. The basic principle here is that sentences within the document recommend each other and the sentences with the highest recommendation scores are considered to be in the generated summary. This involves constructing a graphical representation of the document, $G(V, E)$, where each sentence in the document is a vertex V linked to all other vertices through edges E in the undirected graph. The edge between two vertices i and j are weighted by a similarity metric w_{ij} to capture the recommendation between sentences s_i and s_j which is calculated as follows:

$$w_{ij} = \frac{\textit{\# of } w_k | w_k \epsilon s_i, s_j}{\log(|s_i|) + \log(|s_j|)} \tag{5}$$

Here, w_k are shared tokens between sentences s_i and s_j. The PageRank algorithm is run on this constructed graph until convergence to find the importance of each vertex as per the update equation below.

$$WS(v_i) = (1-d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum\limits_{v_k \in Out(v_j)} w_{jk}} WS(v_j) \tag{6}$$

In the above equation, the importance score WS of vertex v_i is a function of damping factor d, incoming edge weights to a given vertex v_i, w_{ji}, and importance score $WS(v_j)$ of neighbouring vertex v_j. The vertices are ranked based on importance and the top n sentences from the document are taken as the summary.

3.4 LexRank

Like TextRank, LexRank (Erkan and Radev, 2004) is a graph based sentence scoring algorithm based on the PageRank algorithm. However, LexRank differs in the way recommendations between sentences are computed.

$$w_{ij} = \frac{\sum\limits_{w \epsilon s_i, s_j} tf_{w,s_i} * tf_{w,s_j} * idf_w^2}{\sqrt{\sum\limits_{w \epsilon s_i} (tf_{w,s_i} idf_w)^2} \sqrt{\sum\limits_{w \epsilon s_j} (tf_{w,s_j} idf_w)^2}} \tag{7}$$

$$tf_{w,s_i} = \frac{\textit{\# of times the word w occurs in sentence } s_i}{\textit{\# of words in sentence } s_i} \tag{8}$$

$$idf_w = \log \frac{\textit{\# of sentences in the document}}{(1 + \textit{\# of sentences with term w})} \tag{9}$$

The similarity metric w_{ij}, between sentences s_i and s_j, is the idf-modified-cosine similarity (Erkan and Radev, 2004) computed between N dimensional vector representation of sentences.N is the number of unique terms in the document. For each word present in a sentence, the corresponding dimension in the N dimensional vector is set to the idf value of the word to construct the vector mapping of the sentence.

4 Results

The machine generated summaries are evaluated using intrinsic and extrinsic measures. Intrinsic (quantitative) evaluation uses ROUGE score (Lin, 2004) which is a recall based metric that compares similar n-grams in generated summaries against the handmade summaries. It is found that ROUGE based evaluation correlates with human based evaluation in comparing machine generated summaries with ideal summaries (Lin and Hovy, 2003). Hence, we consider ROUGE-1 and ROUGE-2 scores for

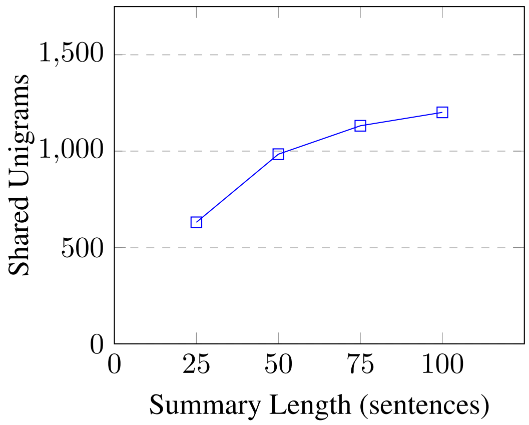

Figure 1: Overlapping unigrams vs summary length of generated summary.

this evaluation, which is the percentage of overlapping unigrams and bigrams respectively between the generated and handmade summaries.

The main idea of this work is to create a study/revision guide for students to help them understand the study material and do well on exams. Hence, the ability to answer exam questions is an indicator of a good summary. The Extrinsic (qualitative) evaluation measures how good the summary is in helping students perform well in the history exam. This is carried out by going through the summaries generated by the above mentioned algorithms and making a decision on how many points can be scored on very short and short answer questions given only the sentences in the summary. The scoring is done manually by human evaluators who refer to the examination grading rubric which is available online.

Length of the summary is an important factor to be considered when generating summaries. The challenge is to balance recall and precision, i.e. to capture as much important information as possible from the whole document while avoiding the inclusion of superfluous information. Such a summary with the right length should make for a faster and better learning experience for students. Fig. 1 shows the relationship between the length of summary, in sentences, with respect to the shared unigrams with the reference summary. While longer summaries have more overlap, the decrease in slope indicates that a decreasing percentage of added sentences match the reference. Thus, 75 sentences was selected as the model summary length.

Algorithms	ROUGE-1	ROUGE-2
LexRank	0.56	0.25
TextRank	0.72	0.44
Luhn	**0.74**	**0.45**
KLSum	0.39	0.17

Table 1: Intrinsic Evaluation: Comparison of ROUGE scores for LexRank, TextRank, Luhn and KLSum summaries compared against the reference summaries.

Algorithms	Exam scores
LexRank	40.8%
TextRank	**53.1%**
Luhn	38.8%
KLSum	46.9%
Reference Summaries	**67.3%**

Table 2: Extrinsic Evaluation: Comparison of exam scores for reference summaries, LexRank, TextRank, Luhn, and KLSum summaries.

4.1 Intrinsic Evaluation Results

Table 1 describes the ROUGE scores of different algorithm generated summaries when compared to the human generated reference summaries. We see Luhn based summaries have the highest ROUGE-1 and ROUGE-2 scores of 0.74 and 0.45 when compared to other algorithms. In this case, ROUGE-1 and ROUGE-2 follow a similar distribution.

4.2 Extrinsic Evaluation Results

The generated summaries were evaluated based on their ability to answer questions on three years' (2017-2019) history exam papers in Hindi. We compare their exam scores with the baseline, the exam scores of the hand generated reference summaries. This comparison is done by evaluators who have studied Hindi and history at a high school level while referring to the grading rubric provided by the CBSE board. It is important to note that the full text is sufficient to answer all of the exam questions scoring 100%. The reference summaries scored 67.3% outperforming summaries of TextRank scoring 53.1%, LexRank scoring 40.8%, Luhn scoring 38.8% and KLSum scoring 46.9% as shown in Table 2.

5 Discussion

When evaluated extrinsically on question answering ability, we see that human generated reference summaries are able to score better on exam questions when compared to machine generated summaries. Among the unsupervised approaches, TextRank scores the most on exam questions, 53.1%. Since TextRank is able to answer approximately 80% of exam questions that the reference summaries answer, we believe that note generation by TextRank provides a good supplementary study tool for students.

We observed the impact of Hindi on the ROUGE metric. The presence of stop words, ambiguous pronouns and other commonly used connecting terms in Hindi artificially raise the n-gram overlap without adding useful information. For example, consider the two sentences below:

नेहरू	ने	गाँधी	से	सत्याग्रह	का	महत्व	को	समझा	।
neharoo	ne	gaandhee	se	satyaagrah	ka	mahatv	ko	samajha	.

Nehru learnt the importance of non-violence from Gandhi.

गाँधी	ने	अंग्रेज़	को	भारत	से	निकलने	का	आग्रह	किया ।
gaandhee	ne	angrez	ko	bhaarat	se	nikalane	ka	aagrah	kiya .

Gandhi urged the British to leave India.

The two sentences have completely different meanings, sharing only subject, Gandhi in common. The English sentences have two unigrams in common, 'Gandhi' and 'the' out of a total of thirteen unique unigrams, approximately a 15% overlap. On the other hand, the Hindi sentences have a total of five unigrams in common out of a total of fourteen unique unigrams, approximately a 36% overlap. This aspect of the Hindi language, with an abundance of connecting terms, would also raise the ROUGE metric of sentences which need not convey useful information.

We reevaluated the importance of the ROUGE score for the chosen task. We notice that a good ROUGE score is not a good indicator of a summary's ability to serve as a study aid. This is evident from the extrinsic evaluation. Luhn summarization, which has the highest ROUGE-1 score (0.74), performs poorly on the question answering task scoring only 38.8%. Conversely, KLSum having the lowest ROUGE-1 score (0.39) performs better than Luhn summarization extrinsically, obtaining approximately 47%. This relationship between the ROUGE and exam scores of the summaries can be confirmed by the Spearman's rho and Kendall's tau coefficients (Yue et al., 2002) , which are -0.4 and -0.33 respectively. The negative coefficients indicate a weak correlation between the summary's ROUGE score and its question answering ability. This shows that, in addition to ROUGE, it is important to formulate evaluation mechanisms that align with chosen application to evaluate machine generated summaries.

We noticed that machine generated summaries have sentences with ambiguous subjects. While the algorithms may identify an important fact, it cannot attribute it to a subject. Consider the following sentence:

स्कूल	का	प्रिंसिपल	एक	कोलोन	था	।	उसने	लड़की	को	स्कूल	से	निकाल	दिया	।
scool	ka	prinsipal	ek	kolon	tha	.	Usne	ladakee	ko	scool	se	nikaal	diya	.

The school's principal was a colonist. He expelled the girl from the school.

When the machine generated summary contains only the second sentence it is able to answer the question "What caused the school rebellion?" (expelling the girl from school) but cannot identify the subject (school's principal) who carried out the action without the preceding sentence. This is a structure we see often in Hindi where one sentence in English corresponds to two in Hindi. In the English version of the text, the fact is stated as

"The principal, also a colon, expelled her."

As the input text documents to the models were not pre-processed, we observed models treating the same entity differently. For instance, the tokens Gandhi and Gandhi-ji. The addition of an honorific suffix 'ji' results in both terms being treated as different. Since the rule of removing the suffix 'ji' applies only to proper nouns, we cannot generalize this as a stemmer rule.

We believe that a TextRank based summarization tool would prove effective for other subjects whose exam questions test factual knowledge like Geography or Biology. However, further testing is required before its portability can be validated. Also, we believe the project would benefit from an Entity Recognizer, as a pre-processing step, to solve both ambiguous subjects problems and the ambiguity caused by the honorific suffixes in the summaries. Nevertheless, we believe that this project represents a step in the right direction towards providing a note generation tool for students in Hindi medium schools.

References

Haghighi Aria and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

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

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.

Navneet Garg, Vishal Goyal, and Suman Preet. 2012. Rule based hindi part of speech tagger. In *Proceedings of COLING 2012: Demonstration Papers*, pages 163–174.

Manisha Gupta and Naresh Kumar Garg. 2016. Text summarization of hindi documents using rule based approach. In *2016 international conference on micro-electronics and telecommunication engineering (ICMETE)*, pages 366–370. IEEE.

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. pages 79–86.

K Vimal Kumar, Divakar Yadav, and Arun Sharma. 2015. Graph based technique for hindi text summarization. In *Information Systems Design and Intelligent Applications*, pages 301–310. Springer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 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*, pages 404–411.

NCERT. 2018-2019. *Bharat Aur Samkalin Vishwa-2*. National Council of Education Research and Training. ISBN: 81-7450-712-4.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Alok Ranjan Pal and Diganta Saha. 2014. An approach to automatic text summarization using wordnet. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 1169–1173. IEEE.

Snigdha Paul, Nisheeth Joshi, and Iti Mathur. 2013. Development of a hindi lemmatizer. *arXiv preprint arXiv:1305.6211*.

Pradeepika Verma, Sukomal Pal, and Hari Om. 2019. A comparative analysis on hindi and english extractive text summarization. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):1–39.

Sakshee Vijay, Vartika Rai, Sorabh Gupta, Anshuman Vijayvargia, and Dipti Misra Sharma. 2017. Extractive text summarisation in hindi. In *2017 International Conference on Asian Language Processing (IALP)*, pages 318–321. IEEE.

Sheng Yue, Paul Pilon, and George Cavadias. 2002. Power of the mann–kendall and spearman's rho tests for detecting monotonic trends in hydrological series. *Journal of hydrology*, 259(1-4):254–271.