Summarization of Hindi News Editorial with Sentiment Determination

A. Pandian, Rajeshwari, Abhishek Saxena

***Abstract**: Text summarization is a natural language processing application that is being researched extensively and applied further to reduce the processing time for various long-winded text-based activities. However, as NLP is still in its budding phase, the work is relatively limited to the English language, leaving regional languages rather untouched despite having an incredible following of speakers. Such is also the case with the Hindi language. In this paper, we propose to come up with an effective method of summarisation news articles in the Hindi language. Like the English variant of this application, we wish to emphasize on the important sections of a Hindi news report and summarize it within 60 to 80 words. The summarization technique will try to identify the theme of the news, named entities and numbers, title terms, etc., for constructing a keyword table. This will be further compared against a knowledge base with weighted keywords for ranking the important sentences in the relevant order and finally picking out the sentence most needed for the summary. Our goal for summarizing the Hindi news articles specifically roots from the dilemma that despite these articles being a rich source of opinionated information about various topics, they are often ignored by the readers because of their long-winded nature that makes the useful information lost in the sea of words decorated by winded introductions and linguistic ornaments like idioms. Hence, this system should enable in an effective means of summary for finding useful information along with pruning all such irrelevant details.*

***Keywords**: Natural Language Processing, Text Summarization, Sentiment Analysis, News Articles, Editorials*

I. INTRODUCTION

Text summarization and sentiment analysis are two applications of Natural Language Processing that are slowly becoming part of the daily life of the modern era by reducing the time taken for a lot of processes that otherwise required extensive human input and their gathered knowledge base. Text summarization, a broader application category of NLP allows for background uses such as monitoring media content all over the world, structuring petabytes of unstructured data in data warehouses of large companies, along with directly visible uses such as curating news report for periodic newsletters. Sentiment Analysis, however, true to its name has enabled us to decipher the overall sentiment of long-winded movie critiques to the sorting of reviews on e-commerce websites based on the reception of the chosen product.

Text summarization is a direct application of Natural Language Processing that deals with compressing text content, scalable and/or voluminous in nature, into its shorter summarized version. The summarized text is expected to highlight all the important contents of the original source by sifting the extraneous and irrelevant information.

Text summarization is a direct application of Natural Language Processing that deals with compressing text content, scalable and/or voluminous in nature, into its shorter summarized version. The summarized text is expected to highlight all the important contents of the original source by sifting the extraneous and irrelevant information.

Text summarization has two main categories of methodologies: Abstractive Summarization and Extractive Summarization.

Abstractive summarization technique involves understanding the main context of the source and reformatting in a better and more compact manner as the output. It is further classified into the structure-based approach that performs summarization based on certain features based on psychological schemas and semantic-based approach that utilizes various semantic models to derive the semantic structure of the document and extract the text accordingly.

Extractive summarization works employ a search of sentences relevant to the title theme and subheadings of the document(s). For doing so, it uses various weighted features such as Term Frequency-Inverse Document Frequency (TF-IDF), title relevance, the occurrence of names or dates, etc.

Sentiment Analysis is an application of NLP that deals with the detection of polarity, or mood of the writer. Or it can be simply said that it tries to understand and reveal the bias of document for the given title or item. It may be used for understanding discrete feelings of a particular individual, or the overall bins for a specific target.

Sentiment analysis generally employs using a bag of that are categorized under bins of positive and negative or a sentiment dictionary creation. It then compares the words in the sentences against the words available in these bins and then evaluates the result based on the emphasis of the matches and the weight of the words that were matched. The content is declared as neutral if the positive weight is equivalent to the negative weight, that ends up in a cancellation, rendering the content as neutral.

II. RELATED WORK

Hindi text summarisation and sentiment determination remain at large when it comes to getting a lot of attention by researchers and engineers due to lack of urgency for the same.

Revised Manuscript Received on April 02, 2020.

Dr. A. Pandian*, Associate Professor, Computer Science department, SRM Institute of Science and Technology, Chennai, India. Email:apandiansrm@gmail.com

Rajeshwari, Student, Computer Science Department, SRM Institute of Science and Technology, Chennai, India. Email:klhn.rajeshwari@gmail.com

Abhishek Saxena, Student, Computer Science Department, SRM Institute of Science and Technology, Chennai, India. Email: 5abhisheksaxena@gmail.com

Manisha Gupta, Dr.Naresh Kumar Garg proposed summarisation of Hindi documents using rule based approach. Various criteria present in the document were chosen as conditions for selecting sentences in the summary. These criteria were then given weights based on their importance in the overall summary. The sentences with the highest scores were finally filtered out and the summary was obtained. This research accomplished the elimination of sentences filled with deadwood words and phrases. The sentences with the highest sum scores were chosen to appear in the summary. [1] Archana N.Gulati and Dr.S.D.Sawarkar made use of scoring based on fuzzy logic algorithms, where extracted features are used as input in the fuzzifier. The output decides the degree of importance. The final summary is determined by picking the sentences with high and very high importance scores. [2] Shashi Pal Singh, Ajai Kumar, Abhilasha Mangal, Shikha Singhal presented a system for bilingual automatic text summarisation using unsupervised deep learning. Their work made use of a Restricted Boltzmann Machine (RBM) to aid in the creation of a feature matrix using weighted features against sentences, according to which the sentence scores were calculated. The summary is generated by sorting the scores in descending order and picking the first sentence by default. [3] Chang-Shing Lee, Zhi-Wei Jian, and Lin-Kai Huang used the applications of fuzzy ontology to perform summation on Chinese news articles. The technique makes use of fuzzy ontology to describe the domain knowledge. The preprocessing step generates useful terms based on the news corpus, which are used as input for the fuzzifier. The system was able to produce a summary of Chinese news articles in English. [4] Jianwu Wu proposed news summarisation of Chinese news articles via soft clustering algorithm. The technique makes use of a sentence clustering algorithm, that makes use of sentence similarity for clustering, until the query sentence is clustered. The results produced, however, are not human friendly, or so to say, not easy to read by the targeted audience.[5]

Vandana Jha, Savitha R., Sudhashri S Hebbar , P Deepa Shenoy and Venugopal K R proposed a multi domain sentiment aware dictionary for sentiment determination of Hindi text. A large word data set taken from multiple source domains and a target domain which is used to build a multiple domain sentiment aware dictionaries for classifying unknown target domain reviews as positive or negative. The dictionary explores a variety of ranges of positive and negative remarks. The output produced includes the sentiment of the provided input and the degree of this sentiment. [6]. Prof.Sumitra Pundlik, Prachi Kasbekar, Gajanan Gaikwad presented a system for class-based sentiment analysis based on multi class classification. The technique involved classification using HindiSentiWordNet (HSWN) and LM-Classifier. The SentiWord Classification separates the body of words into positive, negative and neutral categories using HSWN. Hindi documents can thus be classified into multiple categories according to their ontology and polarity. [7] Charu Nanda, Mohit Dua, Garima Nanda presented the idea of sentiment analysis of movies reviews provided in Hindi language using machine learning algorithms. The technique makes use of classification using Support Vector Machines (SVMs) and the Random Forest Algorithm to identify the features of the text by searching for patterns. This allows for classification of reviews into negative and positive categories. [8] Santosh Kumar Bharti, Korra Sathya Babu and Rahul Raman implemented a Context-based Sarcasm Detection System for Hindi Tweets. The technique involved using POS Tagging and the aforementioned HSWN. News and tweets from the corpus are fed into the Sarcasm Detection Engine (SDE), and sarcasm is identified if the tweet contradicts its context.[9] Mukesh Yadav and Varunakshi Bhojane created a Semi-Supervised Mix-Hindi Sentiment Analysis Engine using Neural Networks. The technique used Neural Networks for the classification task, and then made predictions using pre-classified words. Words could be classified as positive or negative. If a classification for a word could not be found, it was stored for later processing. This allows for sentiment analysis in various domains, such as health, business and tourism.[10]

III. EXPERIMENTAL PROCEDURE

The proposed implementation will be carried out with the following steps in order: inputting data, pre-processing of the raw input, weighing the pre-processed input using weights, using the weights with Restricted Boltzmann Machine (RBM), further processing phase and finally obtaining the summarized input. A deeper description of these phases is as follows:

Summarization Section:

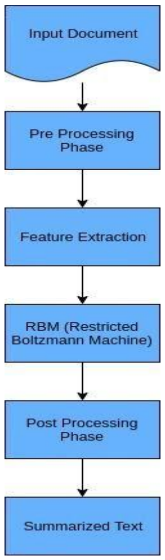

Fig 1: Summarization Phase, Diagonal based feature extraction for handwritten character recognition system using neural network J. Pradeep, E. V. R. Srinivasan, S. Himavathi

A. Input Phase: The very first step of the operation where the raw contents are inputted either directly through some corpus or by converting an image to text using an OCR tool.

B. Pre-Processing Phase: This is the second phase of the summary operation, which is further broken down into the following steps to get usable input for the next step:

a. **Sentence Segmentation** - The raw sentences are broken down into individual sentences in this step.

b. **Tokenization**: The individual sentences are further broken down into its constituent words for cherry-picking in this step along with calculating the term frequency of each term in the input.

c. **Stop Word Removal**: There are certain words in all languages that are constantly repeated in order coherent grammatically correct sentences. However, these words offer no merit to the actual count of tracking the word frequencies as they generally mean nothing to the actual content. Hence, these words are removed in this step.

C. Feature Extraction: This is the third step in the operation where the determined constituents are given certain weights based on the chosen features for determining the relevance of the word/sentence to the document theme. The obtained values will be then used for forming the feature matrix to be used in the next step. Some of the features chosen for this step are:

a. **Term Frequency:** This feature gives weight to the term based on the number of its repeated occurrence in the entire document. The term frequency (Tf) for a term is calculated as follows

$$\mathrm{TF(t)} = \frac{Number\ \ of\ term\ \ t\ appears\ \ a\ document}{Total\ \ number\ \ of\ terms\ \ the\ document}$$

D. Sentence to Sentence Similarity Feature: A sentence (S) is compared against another sentence (S’) and their similarity factor is calculated. If S’ scores high in this feature then it will have less weight and vice versa. Sentence to sentence similarity (SS_Similar) is calculated as:

$SS\_Similar_i$ = i=1Njsimilar (sentence$_i$, sentence$_j$) where 1 <= i <= N.

E. Number Token Feature: This feature is calculated by counting the number of numerical values in the sentence against the total word count of the associated sentence. The formula for Number token feature (NF) is given as follows:

$$\mathrm{NF_i} = \frac{numeric\ \ value\ \ term\ \ \ \ i}{sentence\ \ word\ \ count}$$

, where numeric_value_term is the said feature term in the sentence i.

F. Sentence Length Feature: This feature for deleting shorter sentences immediately that may not hold any useful tokens at all directly. Sentence length (SL) is calculated as:

SLi = 0 (if the number of sentences is less than a chosen threshold value)

G. Proper Noun Feature: This feature identifies the presence of any proper noun contents and gives them weight.

e. **Named Entity Feature:** This feature is used for recognizing the weight of the identified object / proper noun based on the repetitive occurrences.

f. **Unique Term Feature:** This feature recognizes the terms with the least number of occurrences in the document and gives the associated sentence weight as the term may describe an important process/event.

g. **Relevance to Title Feature:** This feature is used to identify the similarity of the terms in the sentence against the title or the title theme and give according to the relevance with it.

h. **Recognition of idioms:** If a sentence is recognized as an idiom from the bag of idioms, then that sentence will be immediately removed from the expected sentences list. The feature Sentence Idiom (SI) is given as:

SFi = 0 (if the sentence has a strong match against an idiom)

D. Sentence Matrix Generation: This is the fourth step in the operation where the features from the previous step are considered as columns while each of the sentences is considered as a row for the 2-D matrix in this step where rows with sentences S = (s1, s2, s3, s4, …… , sN) and columns with features F = (f1, f2, f3, f4, ……, fx). For our model, we will be having the nine features like the column for this 2-D matrix.

E. Deep Learning Process: The created feature matrix is then used by the Restricted Boltzmann Machine (RBM) for recalculation of features to pick out the more important ones and create hidden features through backpropagation. The final matrix after the deep learning processing will have certain score against each of the sentences in the matrix which will be used for determining the importance of a sentence in the summary. We have pursued the ideal route of selecting the sentences with the highest scores for our summary.

Now following this, the summarized text will be used for determining the sentiment of the article. This will be performed through the following steps:

In both the training and predicting phases, the means of inputting the text will remain the same. Hence all the steps through steps A to C will be repeated with the respective changes made into them for accommodating this next step. However, following this, it will include the following steps for sentiment analysis:

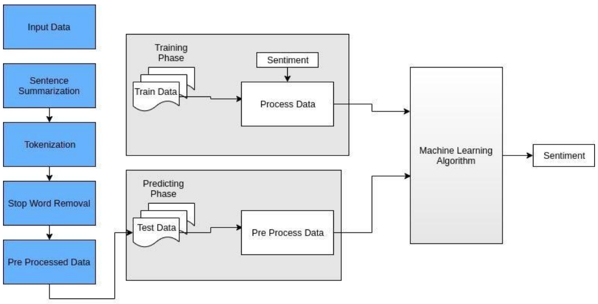

Fig 2: Sentiment Determination Phase, Source: Unknown

A. **Creation of Bag of Words:** A bag of words will be created for the Hindi language with positive and negative bins manually. This will correspond to the Sentiment block in the above diagram. The bag of words will be created using a dataset that comes with already labelled attributes for the generally used words while describing objects of concern.

B. Feature Selection: Features will be again selected for performing the machine learning algorithm. We will be using the TFIDF feature used for the summarization process again and use the following new features:

a. Positive Sentiment match: For determining the value of this feature, the words in the sentence will be matched against the positively labelled terms in the bag of words and weighted as per the chosen threshold value.

b. Negative Sentiment match: It will be calculated similarly, with the exception of matching the words in the sentences against the negatively labelled terms.

We will not be performing

any feature selection as we only have three features going into the SVM which are essential for determining the sentiment.

Machine Learning Algorithm: Machine Learning algorithm, here a Support Vector Machine (SVM) will be used for training the model with positive and negative news articles. SVM is an ideal choice for this process due to its robustness in determining the results under ideal conditions. Since our features are linearly separable, their separation on the hyperplane will be done was done with relative ease.

IV. RESULT AND DISCUSSION

To evaluate the methods and the techniques, Hindi news articles were used. Most of the proposed methodology was successfully executed. For implementation, textblob Python library has been used.

Challenges faced while processing the model were finding the suitable library for carrying out the natural language processing operations, ending delimiter of Hindi language as some of the sources prefer suing the actual “purna viram” (|) whereas others use the “.” Symbol to end the sentences and lack of consolidated sources of training sets. Other challenges were that the algorithm fails to separate the ending token and the preceding word in the absence of whitespace between them and in case of poor spacing, the algorithm may consider the same token twice.
The discrepancy causes the further pre-processing actions to enable uniform testing.

TABLE I: Meta data analysis

S.No	Meta Data	Input 1	Input 2	Input 3	Input 4	Input 5
1	MD 1	15	12	8	20	14
2	MD 2	1031	342	262	1528	830
3	MD 3	44	24	13	82	38
4	MD 4	900	199	136	1369	736

*Input (as given in Table I) – Dataset articles
*MD 1 – Number of words in Heading
*MD 2 – Number of words in Article
*MD 3 – Number of sentences
*MD 4 – Number of stop words removed (important words)

TABLE II: Processing Time for each feature

S No	Features	TIME(S)				
		Input I	Input 2	Input 3	Input 4	Input 5
1	F1	0.023	0.013	0.014	0.022	0.021
2	F2	0.012	0.006	0.010	0.019	0.011
3	F3	0.074	0.032	0.022	0.118	0.061
4	F4	0.016	0.008	0.004	0.025	0.015
5	F5	0.016	0.007	0.004	0.027	0.014
6	F6	0.013	0.005	0.003	0.024	0.011

* Input (as given in Table II) – Dataset articles

*F1 – Total number of words
*F2 – Total number of important words
*F3 – Total number of sentences
*F4 – Term frequency
*F5 – Inverse Document Frequency
*F6 – Term Uniqueness

V. CONCLUSION

As much as summarization is a problem much worked upon, idiom elimination happened to be one of the major hurdles that we faced as idioms often included terms that had a positive hit on the used features. RBM proved to be an effective way of dealing with the problem as it neatly included all the features that were designed to deal with the problem. In future, we would look for more features and expand the number of features to see what kind of changes it brings to the accuracy of the summary. We aim to reach for higher accuracy levels, albeit at the cost of higher processing time currently. In future, we would like to balance the overall aspect of time vs accuracy dilemma for better robust results.

REFERENCES

1. Manisha Gupta, Dr.Naresh Kumar Garg, “Text Summarization of Hindi Documents using Rule Based Approach”, International Conference on Micro-Electronics and Telecommunication Engineering (2016)
2. Shashi Pal Singh, Ajai Kumar, Abhilasha Mangal, Shikha Singhal, “Bilingual Automatic Text Summarization Using Unsupervised Deep Learning”, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016
3. Archana N.Gulati, Dr.S.D.Sawarkar, “A Novel Technique for Multi Document Hindi text summarization Based Approach”, 2017 International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017)
4. Chang-Shing Lee, Zhi-Wei Jian, and Lin-Kai Huang, “A Fuzzy Ontology and Its Application to News Summarization”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 35, NO. 5, OCTOBER 2005
5. Jianwu Wu, “Web News Summarization via Soft Clustering Algorithm”, 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery
6. Vandana Jha , Savitha R. , Sudhashri S Hebbar , P Deepa Shenoy and Venugopal K R, “HMDSAD: Hindi Multi-Domain Sentiment Aware Dictionary”, 2015 Intl. Conference on Computing and Network Communications (CoCoNet'15), Dec. 16-19, 2015, Trivandrum, India
7. Prof.Sumitra Pundlik, Prachi Kasbekar, Gajanan Gaikwad, “Multiclass Classification and Class-based Sentiment Analysis”, 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India
8. Charu Nanda, Mohit Dua, Garima Nanda, “Sentiment Analysis of Movie Reviews in Hindi Language using Machine Learning”, International Conference on Communication and Signal Processing, April 3-5, 2018, India
9. Santosh Kumar Bharti , Korra Sathya Babu, Rahul Raman,“Context-based Sarcasm Detection in Hindi Tweets”, unpublished
10. Mukesh Yadav, Varunakshi Bhojane, “Semi-Supervised Mix-Hindi Sentiment Analysis using Neural Networks”, unpublished.

AUTHORS PROFILE

Dr. A. Pandian received his MCA degree from Bharathidasan University, Tiruchi. He received his M.Tech degree from Punjabi University, Patiala, Punjab and M.Phil. degree from Periyar University, Salem. He has completed Ph.D. (Computer Science & Engineering) in SRM Institute of Science and Technology, Chennai. He has over twenty-four years of experience in teaching. He is working as Associate Professor in the Department of Computer Science & Engineering), SRM IST, Chennai. His areas of interest are text processing, information retrieval and machine learning. He is a member of ISTE, IAENG, IACSIT and ISC. He has published more than thirty papers in many international conferences and refereed journals of repute. Also, he filed four patents in the Intellectual Property Rights of India..

Rajeshwari is a fourth-year student currently pursuing her Bachelor's in technology with specialization in Computer science engineering from SRM institute of science and technology. She has completed courses like Machine Learning, Python Programming et cetera from Coursera. Skilled in programming, she looks forward to work in a reputed company in the IT industry. Rajeshwari's work in the past has been mostly in Machine Learning..

Abhishek Saxena is a fourth-year student currently pursuing her Bachelor's in technology with specialization in Computer science engineering from SRM institute of science and technology. He has completed courses like Android Application Development, Java Programming et cetera from Coursera and Udacity. Skilled in programming, he looks forward to work in a reputed company in the IT industry. Abhishek's work in the past has been mostly in Mobile Application Development.