Temporal Condensation of Tamil News

Shreenidhi S¹, Prof. Sridhar Ranganathan²
¹SCOPE-School of computer science Tamil Nadu Vellore Institute of Technology, Chennai, Tamil Nadu
²Vellore Institute of Technology, Chennai

ABSTRACT: Since the dawn of the Internet, we have been inundated with an excess of information. The volume of information available on the Internet is expected to grow exponentially. This brings a need for summarization of information. Thus, making summarization one of the most sought-after topics in the domain of natural language processing. It is essential to be informed about the vital happenings, and newspapers have been serving this purpose for a very long time. Sadly, there is a perception among the general public that no news agency today can be unequivocally trusted, the credibility of news articles is uncertain. Therefore, one has to read news articles from various sources to get an unbiased view on topic. When a query related to an event is entered in SEs like google, the search renders an overwhelming number of responses, it is humanly impossible to read all of them. In an effort to address the aforementioned problems, a condensation of news articles covering the Tamilnadu Legislative Assembly election is performed. The news articles were collected from various news sources over a period of two months. The collected articles were translated from Tamil to English. These articles included news about various events, in order to segregate Tamilnadu related news k-means clustering was performed on the dataset. The relevant news articles acquired was pre-processed to remove ambiguity and mistakes from translation. These articles were summarized individually using a linear regression model that gave importance to features such as named entities, number of words that were similar to title etc. The acquired individual summaries were summarized using BERT extractive summarizer as it would reduce redundancy. When generated summary was compared with introduction and title of the article in the absence of an introduction a precision of 0.512, recall of 0.25 and f-measure of 0.31 were obtained.

Index Terms: Named Entity Recognition, Extractive summarization, Inverted Index, BERT

I. INTRODUCTION
In an online poll conducted across major cities of India regarding the perception of public about news coverage, an overwhelming percentage of people (71%) felt that the news coverage was biased, meanwhile 80% of people also felt that coverage was unnecessarily sensationalized for certain news events while certain important events were given enough attention. Another important point that can be noted from the survey, is that people give less importance to regional news. In order to address these issues, a system that uses news data collected from different news agencies is used, thus alleviating the bias in them. These news articles are mined from various news agencies in Tamil and then converted to English, so that NLP techniques can be used with ease.

News from print media plays a vital in a democratic country by providing key information about vital happenings and issues, thus keeping people politically and socially updated and aware. With upsurge in the use of internet, numerous news articles are written everyday about any given news event, this makes capturing the key points of an event very difficult. Recent advancements in the field of machine learning and deep learning have brought a major breakthrough in the field of NLP. Techniques like Seq2Seq, RNN, BERT model etc. can be leveraged to create summary of good quality.

II. RELATED WORK
Most of the work done focusing on news summarization involves prominent use of Tf-Idf, sentiment analysis and clustering. Few news articles belonging to major news topic such as sports, health etc. were used by Mirani, T. B et al. [3] to get the first level summary. Extraction based summarization technique is used to get a separate summary for each news source. The articles are tokenized into sentences, then a for each word in the sentence, word frequency is calculated as the number of times a word appears in the sentence divided by the number of words in the sentence. Words with frequencies below or above a certain range are ignored. Sentences are assigned an importance score based on the word frequencies.
The top results are fetched as summary. This is the first level summary. Sentiment analysis is performed on the first level summary to gauge the sentiment of each news source and to validate the authenticity of the news. Text Blob library is used for the sentiment analysis, this returns a score between -1 and 1. A second level summary is formed from the first level summary. The text on which summarization has to be performed, firstly undergoes pronoun resolution. The text is then tokenized into sentences first, and later into words. Part of speech tagging is done in order to identify nouns, as nouns are used to build lexical chain using Silber and McCoy’s method. Sethi, P et al. [5] also proposes new scoring criterions that is used to identify important parts which will eventually make it to the summary. Based on the aforementioned scoring criterion, the important lexical chains are identified. Using these chains individual sentences are scored, the sentences whose score is greater than a threshold become part of the summary.

Nayeem, M. T et al. [7] propose a method to make multi-document summarization coherent. Articles are tokenized to get sentences. Similarity between sentences is computed as cosine similarity of the TF-IDf vector. Importance of sentence is calculated with Text Rank algorithm. Clustering of sentences is performed using hierarchical agglomerative clustering. Clustering serves two purposes, on limiting the number of sentences selected from a cluster redundancy can be reduced, when sentences from distant clusters are chosen, information coverage is improved. To order the sentences in summary named entity repetition is used as it is an important sign of coherence.

### III. PROPOSED SYSTEM

1. **DATA**
   
   Data is mined from websites using web mining tools, a key is added to base url, so that the urls of all the articles relevant to key can be fetched. After fetching the required urls, each page is scraped along with html structure and later only the content inside specific tags are fetched and stored. The mined data is stored as a json file.

2. **SUMMARIZATION**

   **2.1 Summary of individual article**
   
   To create summary for individual articles when a named entity is searched BERT summarizer is used. This was chosen because it does not require a training dataset and when tested against LexRank, it gave a better ROUGE score.

   **2.2 Overall summary**

   **2.2.1 K-means clustering**

   News from various states were mixed up in the dataset, to get a coherent summary, k-means clustering was used. NER was used to build a TF-IDf vector of NEs present in the the body of article.

   **2.2.2 Linear Regression**

   To get summary from individual articles, a linear regressor was used, this model was chosen because features that we want to stress upon can be decided by us.

   **2.2.3 BERT Extractive summarizer**

   Second level summary was obtained by passing first level summary to BERT Extractive Summarizer model.

---

**Fig 1. Result of k-means clustering**

**Fig 2. Features extracted to train Linear Regression**

3. **BUILDING A WEB APPLICATION**

   Web application was built using Django a python based framework that follows MVT template.

---
In MVT architecture, components are loosely coupled, hence it is easier to make changes. The Controller which acts communicates with both model (data and logic) and view (presentation layer) needs separate code to be written in MVC model. In MVT this part is taken care by the framework itself.

**4. MAKING USER-FRIENDLY UI**

1. **For user to see summary of an event**
   
   When an event is selected by a user, all the news articles related to the event is summarized and displayed in the order of occurrence. Finally, an overall summary will be displayed in Tamil. To speed up the process intermediate results of each day is stored and gets modified regularly to be current.

2. **Query and summary**
   
   All articles related to the query are fetched and summary is displayed chronological order. In the context of election, named entities are of significance importance, hence those are identified and an autocomplete feature is added to make the search easy.

   UI was built using Html, bootstrap and jQuery. Bootstrap and Javascript were downloaded separately and added to static folder (Django folder).

**5. VISUALIZATION**

Visualizations can be done with the help of Chart.js. Chart.js can be easily installed and the module can be set-up in Django settings. Visualization was performed to interpret the emotions exhibited towards various Named entities.

**IV. EXPERIMENTAL SETUP**

**1. DATA ACQUISITION**

News articles were scrapped from 3 Tamil News Websites with help of Beautiful Soup, a web mining library in Python. The websites are Hindutamil, Dinamani and Dailythanthi.

The scraper was run using google colaboratory.

The scraped article was stored in the form of json with five attributes.
2. WEB APPLICATION
The web application was developed using Django. The necessary python packages (rest_framework, chart.js, djongo) were installed and configured in Django framework’s settings.py

3. DATABASE
MongoDB was used as database as it is scalable. It is also a nosql database, which allows unstructured, non-relational data to be stored with ease. MongoDB server was hosted locally.

When user requests a query, a http request is made with the query appended to the url, the query is then looked up in the inverted index, the relevant articles are fetched from database using ids stored in the inverted index.

V. RESULTS AND DISCUSSION

Fig 4. Autocomplete aiding querying

When a piece of text is entered in search box, all named entities that have it as substring are suggested by the autocomplete.

Fig 5. Summary of individual articles for a given query

Fig 6. Final summary in Tamil

Fig 7. Result of VADER in Gujarat Municipal elections (sentiment analysis on the entire text)

Fig 8. Result of aspect based sentiment analysis in Gujarat Municipal elections

On comparing Fig 6 and Fig 8, we can conclude that aspect based sentiment analysis gives better insight.

VI. CONCLUSION
The technology stacks that was used for development of this was found to be suitable for the project. The generated summary
from BERT extractive method gave a precision of 0.52, recall of 0.25 and f-measure of 0.31. Django framework can be used to build large projects with multiple dependencies in a short span of time as it is loosely coupled. The response time will be improved if the process of summarizing is parallelized. In real-time a single query will be requested multiple times, in such cases, the results can cached when a query is requested for first time and served readily when a request is made later.

REFERENCES
1. Sharma, Parul, and Teng-Sheng Moh. "Prediction of Indian election using sentiment analysis on Hindi Twitter." In 2016 IEEE international conference on big data (big data), pp. 1966-1971. IEEE, 2016.
2. Liu, Mingrong, Yicen Liu, Liang Xiang, Xing Chen, and Qing Yang. "Extracting key entities and significant events from online daily news." In International Conference on Intelligent Data Engineering and Automated Learning, pp. 201-209. Springer, Berlin, Heidelberg, 2008.
3. Mirani, Tarun B., and Sreela Sasi. "Two-level text summarization from online news sources with sentiment analysis." In 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), pp. 19-24. IEEE, 2017
4. Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv: 1508.04025 (2015).
5. Sethi, Prakhar, Sameer Sonawane, Saumitra Khanwalkar, and R. B. Keskar. "Automatic text summarization of news articles." In 2017 International Conference on Big Data, IoT and Data Science (BID), pp. 23-29. IEEE, 2017.
6. Nasukawa, Tetsuya, and Tohru Nagano. "Text analysis and knowledge mining system." IBM system journal 40, no. 4 (2001): 967-984.
7. Nayeem, Mir Tafseer, and Yllias Chali. "Extract with order for coherent multi-document summarization." arXiv preprint arXiv: 1706.06542 (2017).
8. Feldman, Ronen, and Ido Dagan. "Knowledge Discovery in Textual Databases (KDT)." In KDD, vol. 95, pp. 112-117. 1995.
9. Konchady, Manu, and James Sanger. Text mining application programming. Vol. 1. Boston: Charles River Media, 2006.
10. "Speech and Language processing" by Dan Jurasky and James H. Martin [ third edition]