A Comprehensive Technique for User Activity Based Twitter Content Summarization

Ayushi Gupta, Devyani Keskar, Madhur Firodiya, Siddhi Hagawane

Abstract: Going through thousands of comments in order to understand opinion of people on a particular post ingests a lot of time and resources of the user. By developing this system, we aim that user gets updated with summarized information of all such events in a time constrained manner. It involves merging multiple opinions stated on the social platform and summarizing it to provide the gist of the topic in order to improve ergonomic experience. For this purpose, our system displays both abstractive and extractive summary of the content. Extractive summary generation makes use of Page rank algorithm and abstractive summary generation makes use of RNN (LSTM).

Keywords: Text analysis, Tweets, Live streaming, Filter based analysis, Anomaly detection

I. INTRODUCTION

Summarization is the process of generating short, fluent, and most importantly accurate summary of a respectively longer text document. The main idea behind summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. As online textual data grows, summarization methods have potential to be very helpful because more useful information can be read in a short time. Social media platforms like twitter, facebook, instagram generate large amount of data through post and comment. Every post on this platform have comments and scrolling through these comments requires the user to spend a lot of time. [4] If the gist of public opinion on a particular post can be provided to user, he can state his opinion in comparatively short span, thereby enabling him to invest the same amount of time for some other work. Generation of summary is not just limited to comments and captions on posts but also can be used for summarizing large amount of data available on the internet through blogs, Wikipedia and research papers. Taking into consideration a subset here we design a system to generate summary of the tweets and replies on the twitter. [1] We also wish to further expand it for recapitalizing the content on other social media platforms. Such a summarization system can also be used in e-commerce applications for generating the gist of thousands of product reviews which can then be used to analyse the sale of the product and understand the public sentiment about likability of product.

It can also help the customer to buy the perfect product. [17] Here we propose a system which is a web application for summarizing tweets on twitter and ensure that they can be summarized effectively and efficiently. In order to achieve these goals, we developed the following objectives:
- Research current technologies and progress associated with tweet summarization
- Perform live streaming to collect dated tweets.
- Implement algorithms and models for different methods of tweet summarization.
- Evaluate the models and tune them if necessary.
- Build and host web application which takes tweets as input and produce summary as output.

II. LITERATURE SURVEY

We carried out the literature survey by going through various research works done in the field of summarization. Some of the drawbacks identified are Linguistic constraint, implementation complexity, underlying limitations of methods applied, semantic & syntactic constraints etc. To overcome these shortcomings, we have proposed a system that combines the benefits of multiple models using a "hybrid approach" which implements semantic as well as syntactic approach for summarization.

A. Research Work

[1] Koustav Rudra, Siddhartha Banarjee "Extractive summarization is applied to extract important tweets and then abstractive summarization is applied to improve readability".

[7] N. Moratanch and S. Chitrakala, "A Survey on Extractive Text Summarization". In this paper, author has described the word level features and sentence level features. In this paper author have categorized all extractive summarization methods into unsupervised and supervised methods and have explained each method and have depicted few evaluation metrics.

Akshil Kumar et al. In this paper author has analyzed and compared the performance of three different algorithms. Firstly, the different text summarization techniques explained. Extraction based techniques are used to extract important keywords to be included in the summary[12]. For comparison three comparison three keyword extraction algorithms namely TextRank, LexRank, Latent Semantic Analysis (LSA) were used. Three algorithms are explained and implemented in python language. The ROUGE 1 is used to evaluate the effectiveness of the extracted keywords. The results of the algorithms compared with the handwritten summaries and evaluate the performance. In the end, the TextRank Algorithm gives a better result than other two algorithms.

Revised Manuscript Received on April 12, 2020.
Ayushi Gupta, Student, Dept. Of Computer Engg., MITCOE, Pune
Email: ayushing9@gmail.com
Devyani Keskar, Student, Dept. Of Computer Engg., MITCOE, Pune
Email: devyani.keskar@gmail
Madhur Firodiya, Student, Dept. Of Computer Engg., MITCOE, Pune
Email: madhur.firodiya15@gmail.com
Siddhi Hagawane, Student, Dept. Of Computer Engg., MITCOE, Pune
Email: siddhi.hagawane22@gmail.com

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958, Volume-9 Issue-4, April 2020

Retrieved from: D7827049420/2020©BEESP
DOI: 10.35940/ijeat.D7827.049420

Published By: Blue Eyes Intelligence Engineering & Sciences Publication
The preprocessing paradigm is used to extract the necessary contents from the collected comments and reduce it in order to make the processing easy.

iii) Extractive Analysis Module
It will generate an extractive summary for a set of tweets and replies to those tweets. The extractive summary will then be used to generate abstractive summary.

iv) Abstractive Analysis Module
The abstractive summary i.e the interpreted summary of the original tweets along with its replies is generated by this module. It is displayed to the user along with the extracted ranked tweets on the system dashboard.

v) Database Module
The excel file storing the monthly tweets along with the summary will be converted into json file and mongodb will be used to import the file.

vi) User Interface
The dashboard will be the platform where user inputs will be recorded to process and deliver the output back to the users on a single click.

The UI will also take in user feedback on every result generated to calculate and improve the accuracy of the system.

Following constraints will be taken into consideration while designing the architecture:
- After every month, the monthly activity of the requested users till date will be updated
- Summary generated will be limited to max 1000 chars.

III. SYSTEM DESCRIPTION

Our system is primarily based on extractive as well as abstractive analysis for which the following system architecture is proposed. It has Majorly the following components/modules:

i) Tweets Extraction Module
Tweets along with its corresponding replies are retrieved by live streaming using the Twitter api for the “userhandle” provided as input by the user.

These are written in an excel file to be used for further processing.

ii) Preprocessing Paradigm
The data is summarized using two different methods i.e. Extractive and Abstractive. These methods will be used to process both monthly activity of a specified user and collective data of tweets tweeted and replies given to the user in past seven days i.e.recent data.

Extractive Summary Generation: Produces a subset of the sentences from the original text.Page-Rank algorithm is used for this purpose.

1. \( J = [L_1, L_2, L_3, \ldots \ldots n] \)

   Where \( n \) belongs to \( J \).

2. sentence[\( i \)] = [L_1, L_2, L_3, \ldots \ldots L_n] \)

3. \( \text{Sentence}[\cdot] - \text{stopwords} \)

4. Construct word vectors \( T_1 = \{w_1, w_2, w_3, \ldots \ldots w_n\} \)

   \( T_n = \{w_1, w_2, w_3, \ldots \ldots w_n\} \)

5. Construct similarity matrix using cosine similarity

   \( \text{Cosine_similarity} = 1 - \frac{\text{m.n}}{\|\text{m}\| \|\text{n}\|} \)

   Where \( m \) = weight of sentence \( 1 \)

   \( n \) = weight of sentence \( n \)

6. Construct a graph \( G = \{\text{Sn}, \text{En}\} \)

   where \( \text{Sn} \) = Sentence

   \( \text{En} \) = Similarity Score

7. \( \text{ranked_sentences}[\cdot] = [R_1, R_2, R_3, \ldots \ldots R_n] \)

   Where \( R_1 > R_2 > R_3, \ldots \ldots > R_n \)

The excel file is read from tweets. Sentence Tokenizer will be used for tokenizing the sentences and vectors will be created. Word embeddings are used to convert phrases into numerical formats, thus helping the networks to learn better. It also provides certain characteristics of the words used in vocabulary. We have used GloVe for word representation in our code which is provided by Stanford. We have limited our dimensions to 100. This helped us to assign weights to the sentences. Now based on weights assigned, similarity matrix based on cosine similarity is constructed. Using this matrix sentences are ranked and top 5 sentences are given as output if the number of tweets is greater than 5 else n tweets are given as output if number of tweets is less than 5(n<5). The output is written in the same excel sheet provided as input.

Abstractive Summary Generation: Reproduces important material in a new way after interpretation. Examines text using advanced natural language techniques to generate a new shorter text.[4] It uses recurrent neural network along with LSTM.

1. \( \text{Ti} \rightarrow \text{Ci} \)

   Where \( i \) = \{1, 2, \ldots \ldots n\}

   \( T = \text{Tweet} \)

   \( C = \text{Integer representation of Tweet} \)

2. \( x = \{x_1, \text{Ti} \mid x \text{ is longest sample}\} \)

3. \( E \rightarrow \text{Encoder} \rightarrow \text{Decoder} \)

4. \( \text{Hi} = [n][k] \)

   Where \( H = \text{OneHotEncoder} \)

   \( n = \text{No. of character in longest tweet length} \)

   \( k = \text{No. of characters in our dictionary} \)

5. \( E_1 = \text{Encoder Input Sentence} \)

   \( D_2 = \text{Decoder Input} \)

   \( \text{Sample T3 = Target} \)

6. Clean the data and append “start” and “end” to T3

7. Convert word \( \rightarrow \) indexed numbers [using dictionary]

8. Convert word \( \rightarrow \) fixed length vector using [embedding layer]

Figure 2. Working of LSTM

Forget Gate: It uses a sigmoid function which compresses the values so that they are between 0 and 1. This helps in regulating the network.

2. Sigmoid activation function: It compresses the values between 0 and 1. It helps to identify relevant values. The values close to 0 are eliminated and the ones close to 1 are kept for further use.

In the diagram,

\( C_{i-1} \) : Previous cell state

\( C_i \) : New cell State

\( h_{i-1} \) : Previous hidden state

\( X_i \) : Current input
The output of forget is a forget vector.

Input Gate: The input gate is used to update the cell state. The previous hidden state \((h_{t−1})\) and the current input \((x_t)\) pass through the sigmoid function to keep important information and forget the irrelevant information by normalizing values between 0 and 1. The network is passed through tanh function. Output of tanh function is multiplied with the output of sigmoid function in order to get the information that the next hidden state should hold.

V. RESULT AND DISCUSSION

We have checked the precision of summary generated by system by conforming it from general public and considering their feedback while calculating the accuracy. The summary was presented to the users which was generated by our system for a particular user handle. We provided two ways of feedback, one being a poll with “yes/no” option and the other with star ratings. If a ‘no’ is received from the user, it is classified as a “negative feedback”. However, if the user marks ‘yes’, it is furthermore validated with star rating which ranges from 1-5. This further parameter ensures proper validation of the result generated.

In star ratings, 2 or less than “2 stars” indicate “unsatisfied opinion” of the user regarding summary even if he has marked ‘yes’ initially. “3 stars” indicate “neutral opinion” and “greater than 3 stars” would indicate a “satisfied opinion.”

The accuracy of the system is calculated on the basis of received feedback with the help of conditional probability.

\[
P(\text{Satisfied} | Y \text{ is } \text{Yes}) = \frac{\text{Probability of No of user having opinion ‘Yes’ \\& ‘Stars’>2}}{\text{Probability of No of user having opinion ‘Yes’}} \times 100
\]

For verifying the above, we took feedback of 50 people on the summary generated for a particular user handle of twitter. Out of which, 42 people voted yes & further out of them 35 gave satisfactory ratings(3 stars and above) and the remaining 7 people gave ratings less than or equal to 3.

\[
P(\text{Satisfied} | Y \text{ is } \text{Yes}) = \frac{35}{42} \times 100 = 83.33\%
\]

regulated by passing the previous hidden state and current input through tanh activation function that normalizes the values between -1 and 1. The output of tanh activation function is pointwise multiplied with the output of sigmoid function to decide what important information from the tanh output has to be kept.

Cell state: To formulate the next cell state, the current cell state \((C_{t−1})\) gets pointwise multiplied with the output of forget gate which is a forget vector that drops values close to 0. The output of the input gate is pointwise added with it to update the cell state to new values that the neural network is expected to remember and which it concludes to be important.

Output gate: It is used to formulate the next hidden state. The previous hidden state \((h_{t−1})\) and the current input \((x_t)\) is passed through a sigmoid function. The new cell state \((C_t)\) This survey resulted in an accuracy of 83.33%.

The expected output for a given tweet (retrieved input) along with its top replies in Extractive summarization is as follows:

VI. CONCLUSION

Summarization systems that are currently existing in the market make use of either statistical approaches or linguistic approaches. Statistical techniques begin with basic features such as term frequency (TF-IDF) and gradually extend to positional features and contextual features in order to ensure high quality summary. The linguistic techniques rely on semantic analysis and adopt Lexical databases to find the association between textual units. This technique generates cohesive summary as compared to statistical techniques using low level features. To achieve benefits of both these approaches, our system makes use of a hybrid approach including statistical as well linguistic techniques. Our system generates both extractive as well abstractive summary. Using this approach, our system generates a summary of the respective user’s activity monthly along with the summary of the replies given to the user in the past 7 days.

FUTURE SCOPE

- **Relocate linguistic restriction:** Currently system can generate summary for only “English” text. System can be extended to generate summary for other language text as well if online lexical database for other languages are available.
- **Product Review summary:** Proposed System can generate summary for comments on twitter posts. It can be extended to generate summary for product reviews on various e-commerce sites.
- **Extension to other social media application:** It is also possible to generate summary for comments on posts of other social media applications like Instagram, hike, Facebook.

Figure 3. Extracted Tweets And Replies

| Sheet number : 1 |
|------------------|
| [altk-data] Downloading package stopwords to [altk-data] C:|Users\mp\AppData\Roaming\altk data... |
| [altk-data] Package stopwords is already up-to-date! |

Figure 4. Extractive Output
REFERENCES

1. Koustit Rudra, Siddhartha Banerjee, "Summarizing Situational Tweets in Crisis Scenario", (2016)
2. Xiaohua Li, Yi Tong, Wei Fu, Ru Zhou, Ming Liu, "Graph-based Multi-tweet Summarization Using Social Signals", (2016)
3. Hiro Tsutsumi, Nishino Masaaki, Yoshida Yasuhisa, Suzuki Jun, Yasuda Norihito, and Nagata Masaaki, "Summarizing a Document by Trimming the Discourse Tree", IEE/ACM Transactions On Audio, Speech, And Language Processing (2015)
4. Sarda A.T. and Kulkarni A.R., "Text Summarization using Neural Networks and Rhetorical Structure Theory", International Journal of Advanced Research in Computer and Communication Engineering (2017)
5. Renjith S.R, Sony P, "An Automatic Text Summarization for Malayalam Using Sentence Extraction", Proceeding of 27th IRF International Conference (2015)
6. Subramaniam Manjula, Dalal Vipul, “Test Model for Rich Semantic Graph Representation for Hindi Text using Abstractive Method.” IRJET (2015)
7. N. Moratanch and S. Chirakala, "A Survey on Extractive Text Summarization", IEEE International Conference on Computer, Communication, and Signal Processing (2017)
8. Arpita Sahoo and Dr. Ajit Kumar Nayak, "Review Paper on Extractive Text Summarization" (2018)
9. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. ArXiv e-prints (2017)
10. Koustit Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, Prasenjit Mitra "Summarizing Situational Tweets in Crisis Scenario" (2016)
11. Ankit Kumar, Zixin Luo, Ming Xu, "Text Summarization using Natural Language Processing" (2018)
12. Alexander M Rush, Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization", 2015.
13. Z. J. Fu, X. L. Wu, Q. Wang, and K. Ren, "Enabling central keyword-based semantic extension search over encrypted outsourced data, " IEEE transactions on information forensics and Security, vol. 12, no. 12, pp. 2986-2997, 2017.
14. C. Chen, X. J. Zha, P. S. Shen et al., “An efficient privacy-preserving ranked keyword search method,” IEEE Transactions on Parallel and Distributed Systems, 2015.
15. Broenlee, J., “A Gentle Introduction to Text Summarization”, March 02, 2018.
16. Dalal V. & Malik, L. G., "A survey of extractive and abstractive text Summarization techniques" In Emerging Trends in Engineering and Technology (ICETET), 2018.
17. Nallapati, R., Zhou, B., Gultehre, & Xiang, B. “Abstractive Text Summarization Using sequence-to-sequence RNN’s and Beyond” (2016)
18. Radhakrishnan, P., “Attention Mechanism Network” Hacker Noon (2017)

AUTHORS PROFILE

Ayushi Gupta, student of BE, Computer Engineering, MITCOE
"Fake Email & Spam Detection Using Naive Bayes & User Feedback Approach"-Springer, ICCSA 2019
"Fake News classification on Twitter using Flume, N-gram analysis and Decision Tree machine learning technique"-Springer, ICCSA 2019
"Human Gait Analysis based on Decision Tree, Random Forest and kNN Algorithm" -JCET, Jan 2020.Her area of interests are Data Science ,analysis & its applications

Devyani Keskar, student of BE, Computer Engineering, MITCOE, “Fake News classification on Twitter using Flume, N-gram analysis and Decision Tree machine learning technique”-Springer, ICCSA 2019
“Fake Email & Spam Detection Using Naive Bayes & User Feedback Approach”-Springer, ICCSA 2019
Her area of interests are Big Data & programming