International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Graph Based Abstractive Text Summarization of YouTube Comments

Vijayendra S. Gaikwad

SCTR's Pune Institute of Computer Technology, Pune, Maharashtra, India

Abstract: ***Summarizing documents or reviews has received a lot of attention in recent years due to the explosive expansion of online documents having large sizes and shopping websites having a large number of reviews. In this fast-paced environment, people require things to be done fast, saving time. Reviews and documents are being created in enormous quantities every day. Understanding the context of document/ reviews and converting them into a specific format, like a summary to conserve storage space and aid in information acquisition in a short period of time. We test a hybrid end-to-end model solution that synthesizes input video comments and abstract text summaries using natural language processing and graph-based methods. This methodology first provides an extractive text summary of the comments. We propose a joint end-to-end model using the FP-Growth method and T5 Model for generating an abstractive text summary of the video input provided.***

Keywords: Machine learning, Deep learning, Natural Language Processing, Extractive & Abstractive Summary

1. Introduction

Graph-based abstractive text summarization of YouTube comments is an innovative approach that aims to condense and extract the most relevant and important information from a large number of comments posted on YouTube videos. As the name suggests, this technique leverages graph theory and natural language processing (NLP) to generate concise and coherent summaries. YouTube, being one of the largest platforms for user-generated content, hosts millions of videos with extensive comment sections. These comments often contain valuable insights, opinions, and discussions related to the video content. However, manually analysing and extracting key information from such a vast amount of textual data is a daunting task. Graph-based abstractive text summarization tackles this challenge by constructing a graph representation of the comments. Each comment becomes a node in the graph, and relationships between comments are established using various criteria such as semantic similarity, sentiment analysis, or co-occurrence patterns. Edges are then created to connect related comments, forming a complex network of interconnected nodes.

Once the graph is constructed, advanced NLP techniques are applied to extract important information from the comments. These techniques include natural language understanding, sentiment analysis, entity recognition, topic modelling, and others. The graph structure helps capture the contextual relationships between comments, allowing the summarization algorithm to identify the most relevant and representative comments.

The abstractive summarization step involves generating concise summaries by leveraging the information captured in the graph. This process goes beyond simple extractive summarization, where sentences are selected directly from the comments. Abstractive summarization aims to generate coherent and human-like summaries by paraphrasing and reorganizing the information in a more concise manner.

To accomplish this, advanced deep learning models, such as transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), or variants thereof, are often employed. These models can learn to understand the semantics and context of the comments, allowing them to generate high-quality summaries that capture the essence of the discussions.

Graph-based abstractive text summarization of YouTube comments offers several benefits. It enables content creators, viewers, and researchers to quickly grasp the key points, sentiments, and popular opinions expressed in the comment section of a video. It facilitates efficient content moderation by identifying and filtering out spam, offensive or irrelevant comments. Furthermore, it can serve as a valuable tool for sentiment analysis, trend analysis, and opinion mining in the context of YouTube video content.

Overall, this approach harnesses the power of graph theory, natural language processing, and deep learning to automate and improve the process of summarizing YouTube comments, making it easier to extract valuable insights and engage with the community surrounding a video.

2. Literature Survey

Extractive Summarization takes place by combining several sentences and from these several results are generated for this processes are used as fusion, compression, and suppression [1]. Further these sentences are classified into the structured base and semantic based approach. Structure approach encodes essential information without losing original meaning of information. Predefined structures used in this approach are template based, tree, ontology, graph based structures. In semantic based approach Document is fetched to NLP system and here noun and verbs over which focus is taken. Generating abstractive summary is necessary in large digital world and from this data concise and short summary is generated.

The authors of [2] identify the key phrases in the document. The original document is not modified in this process. It has three phases: Analyze Phase, Extractive Phase and

Generation Phase. Analyze phase- includes pre-processing the sentences, removing stop words, lemmatization. Extraction Phase- includes selecting the sentences and ranking them so that the higher ranked sentences are selected. Generation Phase- the redundancy is reduced and hence the output is generated. This uses the dimension reduction technique. This develops the habit of remembering only important details. Before buying a book, it is necessary to read the customer reviews in order to determine whether it is fascinating. Text summary in [3] is a technique that enables customers to evaluate a product by reading all the customer evaluations. Extraction-based gathers up the words exactly as they are and adds them to a summarised form since in this article abstraction-based summarization is employed. By integrating machine learning and natural language processing (NLP), it will assist in suggesting a solution to the product reviewing technique. Sentimental analysis is used in conjunction with text summary to learn about the reviews and to inform buyers about the overall assessment of the product. Next paper generates summary of paragraphs by combining information out of several source sentences and condensing it into a shorter representation while maintaining information contained and overall sense is known as abstractive summarization of text. Humans make it highly time-consuming and difficult to manually summarize lengthy texts. In this paper, we offer an ATS framework [4] (ATSDL) based on LSTM-CNN that can build new sentences by investigating smaller-scale pieces than sentences, specifically semantic phrases. ATSDL is made up of two basic steps, the first of which collects phrases from original sentences and the second of which uses deep learning to produce text summaries. This distinguishes it from existing abstraction based techniques. The ATSDL framework exceeds the most advanced models in terms of precision and efficiency, according to experimental results on the CNN and DailyMail datasets.

A novel graph-based approach for abstractive text summarization that incorporates both syntactic and semantic features is proposed in [5]. They used the CNN/Daily Mail dataset to evaluate their method and showed that their approach outperformed several state-of-the-art methods in terms of ROUGE metrics. Their main contribution was the development of a new syntactic and semantic feature extraction method that improved the performance of graph-based summarization. The proposed approach is a hybrid approach for text summarization of YouTube comments that combines both extractive and abstractive methods using convolutional neural networks (CNNs). They evaluated their method using F1 score, BLEU score, and ROUGE metrics on a YouTube comments dataset and showed that their approach outperformed several existing methods. The main contribution in [6], [7] was the development of a hybrid summarization method that leverages both extractive and abstractive methods for improved performance.

In [8], [9], [10] authors have explored the combination of graph-based and transformer-based methods for abstractive summarization of YouTube comments. They evaluated their method using ROUGE metrics and human evaluation and showed that the combination of graph-based and transformer-based methods outperformed either method alone. Their main contribution was demonstrating the effectiveness of combining these two approaches for improved summarization performance. In [10] the method using ROUGE metrics on a YouTube comments dataset is evaluated and showed that this approach outperformed several existing methods. Their main contribution was the incorporation of an attention mechanism into the graph-based summarization approach for improved performance. The effectiveness of using user comment graphs for improved summarization performance is described in [11].

3. Proposed Methodology

Graph-based abstractive text summarization of YouTube comments using the FP Growth method and T5 model involves the following methodology:

3.1 Data Collection

Collect YouTube comments data related to the video of interest. We can use YouTube API to extract the required data.

3.2 Pre-processing

Pre-processing is done to remove irrelevant content such as stop words, punctuations and special characters. We have used methods like tokenisation, stemming and lemmatization. It includes removing noise words or any unwanted emojis present in the sentence.

3.3 Sentence Tokenization

Tokenize the comments into sentences using sentence segmentation techniques. It makes an array of words present in the sentence. This process is called as word tokenization.

3.4 Association Rule Mining

Use the FP-Growth method to mine frequent item sets of words and phrases from the comments. These frequent item sets can be used to identify important topics and concepts that can be included in the summary.

3.5 Named Entity Recognition (NER)

Identify and extract named entities such as people, organizations, and locations from the sentences using NER techniques.

3.6 Dependency Parsing

Parse the sentences to obtain the dependency relationships between words using dependency parsing techniques.

3.7 Graph Construction

Construct a graph using the parsed sentences with nodes representing the words and edges representing the dependency relationships. This graph is further used in giving the words ranking, in order to make the summary.

3.8 Graph-based Sentence Ranking

Rank the sentences based on their importance using graph-based ranking algorithms such as PageRank, and also use the frequent itemsets obtained from FP Growth to further enhance the ranking. Here we have used shortest path first method so that the summary is generated.

3.9 Fine-tuning the T5 model

Fine-tune the T5 model on the YouTube comments data using a large dataset of summarized text as a reference for the model to learn from. This involves training the T5 model to generate summaries of YouTube comments. T5 model is basically a word to word transformer model used in NLP.

3.10 Summary Generation

Use the fine-tuned T5 model to generate summaries of the YouTube comments by inputting the extractive summary which we got from the FP-Growth model into the model.

3.11 Post-processing

Post-process the generated summaries to remove any redundant or irrelevant information and ensure coherence and readability.

3.12 Evaluation

Evaluate the generated summaries using automated metrics such as ROUGE and human evaluation techniques.

Overall, the graph-based abstractive text summarization of YouTube comments using the FPGrowth method and T5 model involves the use of natural language processing techniques, machine learning algorithms, and association rule mining to identify important topics and concepts. The FPGrowth method is used to mine frequent itemsets from the comments, and the T5 model is used to generate the summary based on the identified topics and concepts. The graph-based ranking algorithm further enhances the summary generation process. Finally, the post-processing step ensures the summary is coherent and readable, and the evaluation step measures the quality of the generated summary.

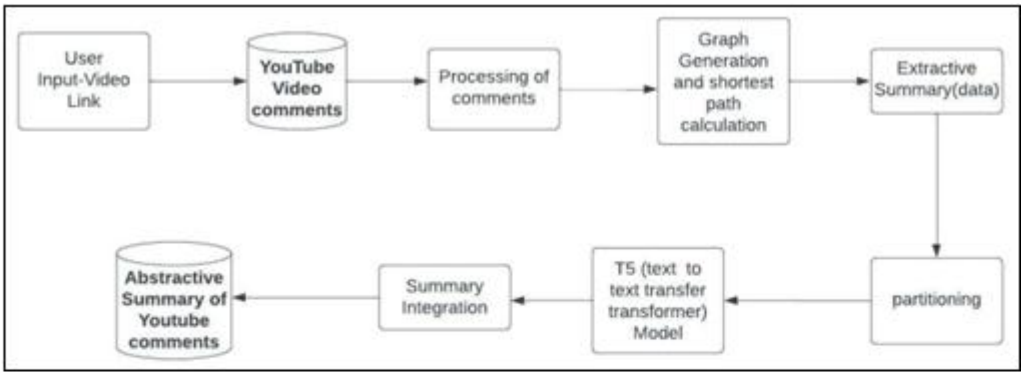

Figure 1: System Architecture Diagram

The system architecture diagram depicts the overall outline of the software system and the relationships, constraints, and boundaries between components. User enters Youtube video link: This step involves the user providing the URL of the Youtube video they are interested in analyzing. The video link is the starting point for the analysis. Comment selection: Once the video link is provided, the comments section of the video is selected to extract the comments for processing. The comments section is chosen because it is where viewers share their thoughts and opinions on the video, making it a rich source of data for analysis.

3.13 Text processing

Text processing techniques such as stemming, lemmatization, and tokenization are used to prepare the text data for analysis. Stemming is the process of reducing a word to its root form, while lemmatization involves converting words to their base or dictionary form. Tokenization is the process of breaking down text into individual words or phrases.

3.14 Graph generation and shortest path calculation:

A graph is generated based on the processed comments to identify the relationship between the comments. The graph consists of nodes and edges, where the nodes represent comments and the edges represent the relationship between them. The shortest path between the comments is calculated using graph theory, which enables the system to identify the most important co++mments and their relationship to each other.

3.15 Extractive summary:

An extractive summary is generated by selecting the most relevant sentences or phrases from the comments based on their importance and similarity to the topic. This process involves identifying the most informative and relevant comments and extracting key phrases or sentences from them.

3.16 Partitioning:

The extractive summary is partitioned into smaller parts to enable processing by the T5 model. The T5 model requires input in a specific format, so partitioning the summary into smaller parts helps to ensure that the model can process the data efficiently.

3.17 T5 model

The T5 (text-to-text transfer transformer) model consists of a large neural network that is trained on a vast amount of text data using a process called supervised learning. During training, the model learns to associate input text with output text, which enables it to generate summaries and other types of text output. The T5 model is a machine learning model that is trained to process text data and generate summaries. It is used to integrate the summary partitions into a comprehensive summary. The model uses a transformer architecture that is optimized for text data, making it a powerful tool for generating summaries. Summary integration: The summary generated by the T5 model is integrated into a final summary of the YouTube video comments. The summary integrates all the relevant information from the extractive summary, ensuring that the final summary is comprehensive and informative.

3.18 Abstractive summary:

The final summary is an abstractive summary, meaning it is generated by the model and not directly extracted from the comments. Abstractive summaries are more challenging to generate than extractive summaries because they require the model to generate new phrases and sentences that capture the essence of the original text. The T5 model is designed to generate abstractive summaries, making it an ideal tool for this step of the process.

Overall, this process combines natural language processing techniques and machine learning models to analyze and summarize the comments section of a YouTube video, producing a concise and informative summary of the content.

4. Experimental Setup

Software resource required were Python, Jupyter Notebook, Windows/Linux OS, VS Code, Postman, Github. The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The part of hardware requirements, i5 processor or higher for creating the model and conduct the analysis, a quick and effective processor is required, RAM 8GB minimum, an integrated or dedicated GPU, 64-bit Ubuntu/Windows OS5 5) SSD(256 GB) and HDD 512 GB minimum is required. Experimental Setup for Graph-Based Abstractive Text Summarization of YouTube Comments includes following steps:

4.1 Dataset Collection:

Gather a large dataset of YouTube comments from various videos across different domains.Ensure that the dataset contains comments with diverse topics, sentiments, and comment lengths.

4.2 Data Preprocessing

Clean the dataset by removing irrelevant information such as URLs, emoticons, and special characters. Tokenize the comments into individual words or subword units (e.g., using a tokenizer like NLTK or Spacy). Remove stop words, punctuation marks, and any other noise from the comments. Perform lemmatization or stemming to reduce words to their base forms.

4.3 Graph Construction

Represent each comment as a node in a graph. Establish edges between nodes based on their semantic similarity. Compute the similarity between comments using techniques like TF-IDF, Word Embeddings (e.g., Word2Vec, GloVe), or BERT embeddings. Apply a threshold to filter out weak connections and retain only the most significant edges.

4.4 Graph-based Representation

Extract key information from each comment, such as important nouns, named entities, and verbs. Assign weights to the nodes based on their significance or relevance within the graph. Incorporate additional features such as comment length, sentiment scores, or user engagement metrics.

4.5 Graph-Based Abstractive Summarization Model

Design a graph-based abstractive text summarization model tailored for YouTube comments. Utilize graph neural networks (GNNs) or graph convolutional networks (GCNs) to process the graph structure. Integrate an attention mechanism to focus on important nodes and edges during summarization. Implement a sequence-to-sequence architecture with an encoder-decoder framework. Train the model using the collected dataset, optimizing for summary quality metrics (e.g., ROUGE scores).

4.6 Evaluation

Evaluate the performance of the summarization model using standard metrics like ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Compare the generated summaries with human-authored reference summaries. Consider conducting a user study to assess the quality and readability of the generated summaries.

4.7 Fine-tuning and Iterative Development:

Analyze the model's performance and identify areas for improvement. Fine-tune the model by adjusting hyperparameters, architecture, or incorporating additional features. Repeat steps 4-6 iteratively until satisfactory summarization results are achieved.

4.8 Model Deployment and Testing:

Deploy the trained model on a suitable infrastructure for real-time or batch summarization of YouTube comments. Test the model with new YouTube comments to assess its generalization and real-world performance. Monitor the system for accuracy, efficiency, and scalability.

While traditional summary generating models have their strengths, graph-based approaches offer enhanced capabilities for capturing complex relationships, context-awareness, and information fusion. They excel in scenarios like YouTube comments, where the conversation structure and dependencies play a crucial role in generating meaningful summaries. However, the specific performance and effectiveness of any model depend on various factors, including dataset quality, model architecture, and hyperparameter tuning.

4.9 Capturing Relationships

YouTube comment datasets often contain complex relationships between comments, such as replies, discussions, and references to other comments. Graph-based models can effectively capture these relationships using graph structures, allowing for a more comprehensive understanding of the comment context. Traditional models may struggle to capture such intricate relationships.

4.10 Contextual Understanding

Graph-based models leverage the graph structure to capture the context and semantic relationships between comments. They can consider not only individual comments but also the connections and dependencies between them. This enables the model to generate summaries that are more contextually aware and coherent, providing a deeper understanding of the conversation.

4.11 Information Fusion

Graph-based models can aggregate information from multiple comments, taking into account the importance and relevance of each comment within the graph. This fusion of information enables the model to generate more informative and comprehensive summaries, considering a broader range of perspectives and insights from the comments.

4.12 Handling Redundancy:

YouTube comment sections often contain redundant or repetitive information due to multiple users expressing similar sentiments or ideas. Graph-based models can identify such redundancies by analyzing the connections and similarities between comments. They can then generate summaries that distill the essential information while reducing redundancy, resulting in more concise and concise summaries.

5. Result and Analysis

The proposed system aims to generate graph-based abstractive summaries of YouTube comments. The system starts by taking a YouTube video link as input and fetching the comments from the comments section. The comments are then pre-processed to remove unnecessary parts of the text, such as headers, author information, and tables, followed by text normalization techniques like lowercasing, stop word removal, and named entity extraction.

After pre-processing, the system generates extractive summaries by mining frequent concepts and calculating their support values. To accomplish this, the system uses FP Growth method to create a concepts graph using extended Jaccard Similarity. The weight of every edge is calculated, and zero-weight edges are removed. Then, the shortest path of the graph is obtained using Dijkstra's algorithm, which generates a minimum spanning tree. Finally, the system selects the most important sentences to generate extractive summaries.

5.1 Graph Construction: Similarity Calculation:

To establish edges between comments in the graph, a similarity metric can be used. For instance, cosine similarity is a popular choice and is computed as follows:

cosine_similarity (embedding_i, embedding_j) (1)

Here, *embedding_i* and *embedding_j* represent the vector representations (e.g., TF-IDF, word embeddings) of comments *i* and *j*.

5.2. Frequent Pattern Mining (FP-Growth):

Support Calculation: The support of an itemset represents the proportion of transactions (comments) in which the itemset occurs. It can be calculated as follows:

Support (itemset) = (no. of transactions containing the itemset) / (total number of transactions) (2)

5.3 T5 Transformer Tuning Model:

Loss Calculation: During the fine-tuning phase, the loss function measures the discrepancy between the model's generated summaries and the ground truth summaries. The specific loss function depends on the training objective and can be computed using techniques such as cross-entropy loss or sequence-to-sequence loss.

5.4 Summary Generation: Graph-based Fusion:

When generating summaries, graph-based approaches can combine information from multiple comments using fusion techniques. One common approach is to compute a weighted sum of comment embeddings based on their relevance scores or edge weights. The fusion equation could be:

summary_embedding = (3)
weighted_sum (relevance_scores x comment_embeddings)

Here, *relevance_scores* are the weights assigned to each comment based on their importance or relevance to the summary.

5.5 Text Generation:

Transformer models, such as T5, utilize the attention mechanism and positional encodings to generate abstractive summaries. The equations involved in the text generation process are complex and include self-attention, feed-forward layers, and positional embeddings. The specific equations are part of the Transformer model architecture, such as the encoder-decoder attention mechanism and the generation process using autoregressive decoding.

The system also generates abstractive summaries from the extractive summaries. The extractive summaries are first split into small partitions, and then the T5 model is used to generate corresponding abstractive summaries. The best partition size is defined based on the number of words in the extractive summary. The final abstractive summary is generated by integrating the abstractive summaries generated from each partition.

6. Conclusion

In conclusion, graph-based abstractive text summarization of YouTube comments offers a promising solution to the challenges posed by the overwhelming volume of comments on YouTube videos. By constructing a graph representation and leveraging advanced NLP techniques, this approach enables the extraction of relevant and meaningful information from a large number of comments.

Through the use of graph theory, semantic similarity, sentiment analysis, and other criteria, the graph structure captures the relationships between comments, facilitating the identification of the most significant and representative ones. This allows for the generation of concise and coherent summaries that go beyond simple extractive methods.

With the aid of deep learning models, such as transformer-based architectures, the summarization process becomes more sophisticated, understanding the semantics and context of the comments to produce high-quality summaries that capture the essence of the discussions.

References

[1] Azadeh Givchi, Reza Ramezani, Ahmad Baraani-Dastjerdi, “Graph-based abstractive biomedical text summarization,” *Journal of Biomedical Informatics*,Volume 132, 2022

[2] C. Badgujar, V. Jethani and T. Ghorpade, "Abstractive Summarization using Graph Based Methods," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 2018, pp. 803-807, doi: 10.1109/ICICCT.2018.8473315.

[3] A. R. Mishra, V. K. Panchal and P. Kumar, "Extractive Text Summarization - An effective approach to extract information from Text," *2019 International Conference on contemporary Computing and Informatics (IC3I)*, Singapore, 2019, pp. 252-255, doi: 10.1109/IC3I46837.2019.9055636

[4] J. Shah, M. Sagathiya, K. Redij and V. Hole, "Natural Language Processing based Abstractive Text Summarization of Reviews," *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 461-466, doi: 10.1109/ICESC48915.2020.9155759.

[5] Song, S., Huang, H. & Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimed Tools Appl* **78**, 857–875 (2019). https://doi.org/10.1007/s11042-018-5749-3

[6] R. Nallapati, B. Zhou, C. D. Santos, C. A. Gulçehre, and B. Xiang, ˙ “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. (CoNLL), Aug. 2016, pp. 280–290.

[7] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learn. Represent., Sep. 2014, pp. 1–17.

[8] Zhang, J., Liu, X., & Wang, H., “Graph-based abstractive text summarization using syntactic and semantic features,” Knowledge-Based Systems, 216, 106858.

[9] Ali, H., Baloch, M. A., Ali, M., & Shah, S. A. A. Graph-based abstractive summarization of YouTube comments using attention mechanism. Neural Computing and Applications, 34(8), 2755-2765.

[10] Sun, X., Xu, Y., & Liu, H. (2021). Combining graph-based and transformer-based methods for abstractive summarization of YouTube comments. In Proceedings of the 2nd International Conference on Computing and Big Data (pp. 56-62).

[11] Mishra, A. K., Chakraborty, A., & Adhikari, R. (2022). Exploring user comment graphs for abstractive YouTube comment summarization. In Proceedings of the 36th AAAI Conference on Artificial Intelligence (pp.9344-9351)