Bhakti Waghmare, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

A Unified Framework for Multi-Video Summarization and Multilingual Translation Using BART

Bhakti Waghmare[1], Sharon Mavelil[2], Dr. Jasbir Kaur[3], Ifrah Kampoo[4], Mansi Rajapurkar[5]

Student of Master of Computer Applications (MCA), Guru Nanak Institute of Management Studies, Matunga, Mumbai [1,2]
Director, GNIMS B-School, Head of Information Technology and HR, Guru Nanak Institute of Management Studies, Matunga, Mumbai[3]
Assistant Professor, Guru Nanak Institute of Management Studies, Matunga, Mumbai [4,5]

Abstract- With the increasing classic of digital streaming services, video content is a common source for people to learn and transfer knowledge. With the rise of educational videos, podcasts and tutorials, knowledge has become easier to access; however, due to their extended length, learning via long-form video can often be time-consuming and inefficient. Therefore, this research addresses the redundancy and inefficiency of long-form videos by creating an automated system to generate short, concise summaries from video transcripts, thus reducing both length and associated learning time. The proposed system uses Automatic Speech Recognition (ASR) to convert audio to text from a video source and employs the BART model along with a hierarchical chunking approach to generate coherent and meaningful short summaries. Addi-tionally, because transformer-based models have a maximum context length, the system also executes iterative summarization to handle long transcripts effectively. Moreover, for ease of use, the functionality for cross-platform translation is included in the system and supports both high-resource and low-resource languages. Finally, the overall framework architecture is modu-lar in nature, leveraging FastAPI for the back end, while lever-aging Flutter for the front-end mobile app ensuring scalability and user friendliness.

Keywords— Abstractive Summarization, BART Transformer, Natural Lan-guage Processing, Multi-Video Synthesis, FastAPI, Flutter Framework, Sequence-to-Sequence Models..

I. INTRODUCTION

Today, video has become the most popular method of expanding one's academic or professional horizons. Video platforms like YouTube house a tremendous amount of technical documentation, university lecture series, and sociopolitical discourse around the world; however, even though the number of available video files continues to grow exponentially, our capability for consuming said files remains static. This has led to a situation where people are experiencing "information overload" due to the multitude of videos that need to be watched and/or listened to in order for them to gain access to valuable, informative content but have been lost in hours of unstructured audio-visual material.

Video search and discovery systems usually focus on retrieving video files but do not take into account the challenges associated with understanding the content after the video is retrieved (the lengthy process involved with having to analyze the content of the video). As a result, text summarization systems have gained popularity recently, especially concerning the use of automated abstractive summarization as a means of creating high-level (and in some cases, entirely new) summaries of what was viewed or listened to in the source video.

While LLMs have been shown to have significant promise in determining the content of video transcripts, there are still many challenges associated with the application of LLM techniques to the problem of analysis of video transcripts (e.g., many times video files contain many instances of repeated content within a transcript, so the resultant transcript

has a word count over the token limit (1024) of modern summarization models (BART)). The fact that multiple videos exist regarding a specific subject or topic also shows another significant challenge associated with the task of summarizing video footage into an automated, abstract, and comprehensive form. [1] [2]

II. RELATED WORK

Summarization is currently shifting from a framework dominated by models based on statistics like LexRank, TextRank, and others to models based on neural networks. The introduction of the Transformer model (Vaswani et al. [3]) has been pivotal in the field of summarization, as it allows models to utilize self-attention rather than recurrent units to capture long-range dependencies.

BART, or Bidirectional-and-Auto-Regressive-Transformer, is used as a denoising autoencoder to pre-train seq2seq models. BART's bidirectional encoder has full context understanding of a sentence, while allowing coherent summaries to be generated through BART's auto-regressive decoder [2]. Most of the work that has been done using BART has been on news related datasets such as CNN-Daily Mail[4], but has not yet been done on summarizing raw video transcripts.

Current research on MDS also identifies minimizing redundancy as an area of difficulty. The model has to consider overlaps in information in the three transcriptions to produce a single summary that includes all of the different perspectives from the video transcriptions. This framework is based on Nallapati et al. [5], by introducing a hierarchical attention mechanism to summarizing video data.

III. SYSTEM ARCHITECTURE

The proposed architecture is flexible and scalable, allowing for the substitution of any one of its components (e.g., the summarization model).

A. Data Acquisition Layer

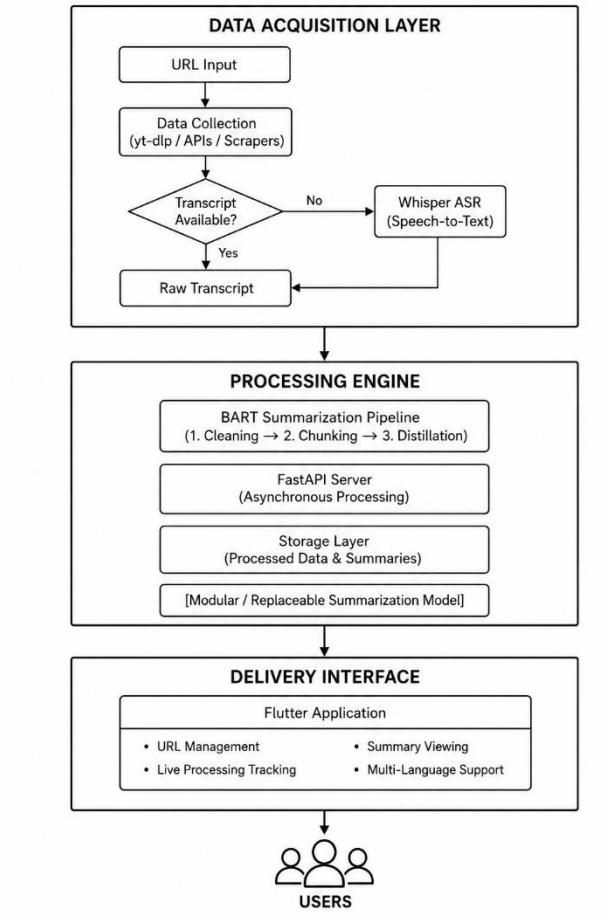
The data collection process begins in the URL Input Module, where the URL link is fed in. The use of fast scrapers and APIs like yt-dlp enables the system to collect data and generate the transcript automatically. Should there be no transcript, the system supports integration with Whisper ASR, which converts audio files to text automatically[6].

B. Processing Engine

The core part of the system, the BART Summarization Logic, processes the unstructured transcript data into structured output through the employment of a 3-stage method (Cleaning, Chunking, and Distillation), running on a non-blocking FastAPI server to allow simultaneous processing [10].

C. Delivery Interface

The users will have access to the Flutter Application and perform the following actions. Manage the list of URLs, track the processing process live and view the generated summaries in English and other languages.



IV. METHODOLOGY

The methodology is framed around tackling the "Length Constraint" problem within typical Transformer architectures.

A. Text Normalization

Transcripts often have noise. We implement a regex based cleaning suite, removing non-lexical filler words ("uh", "um", "like" etc.) and handling common ASR mistakes. The performance of the BART model is strongly influenced by the grammatical accuracy of the input, making the normalization extremely important.

B. Hierarchical Recursive Summarization

Since BART has a context window limit of 1024 tokens, simply chopping up the transcript is harmful. The benefit of lecture notes is often that important information can be late in the lecture, so truncating them is an issue. For our implementation, we employ "Semantic Chunking":

$$\mathbf{S^{(final)}} = \mathbf{fBART}\left(\mathbf{\Sigma\, fBART(Chunk_i)}\right)$$

Here the transcript T is broken down into n chunks. Each chunk is passed through the model and generates a "local" summary. The local summaries are concatenated to produce an intermediate text I. Finally a "Global Pass" is carried out on I, ensuring context is present through the text [7].

Multi-Video Contextual Merging

If several URLs are input into the model, they are treated as a single body of text. Inter-video Synthesis is then performed, where a point that is made in video A is enhanced with information from video B, yielding a rich summary relative to creating an individual summary for each video [8].

The transcripts retrieved from numerous videos are first normalized and then gathered into a unified corpus for multi-video summarization. The combined transcript is put to semantic chunking where similar content segments are grouped together irrespective of the original video source.

At the intermediate summarization phase, local summary generation of local chunks is done and they are merged and analysed for overlapped and complementary and unique information. Excess articles are removed, maintaining topic diversity The last global summarization pass integrates information from all input videos for generating a unified and comprehensive summary that reflects different views on the same topic.

C. Translation and Language Mapping

Once we have created the summary in English, we translate it into the preferred language depending upon the selection made by the user. We employ the lru_cache for fast translation to ensure speediness in several scenarios and also convert all special characters in non-latin alphabets (Hindi, Tamil, etc.) to those of the destination language[9].

V. IMPLEMENTATION DETAILS

A. Backend Stack

Python 3.10 and FastAPI are employed for the backend part. Inference for the model is handled using the Transformers library by Hugging Face[12]. In order to optimize the performance when inference is running on the CPU, Py Torch's quantization mechanism is used to reduce the model's size to about 400MB (original BART Large)[14].

B. Frontend Stack

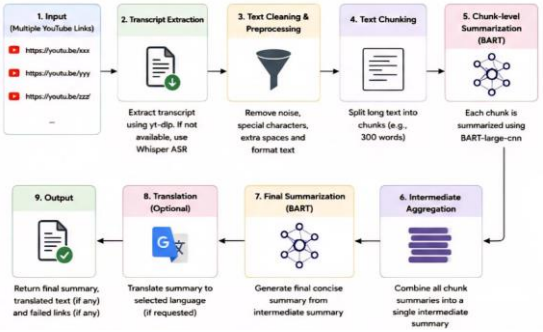
The frontend using the Flutter architecture follow the BLoC (Business Logic Component) architectural pattern to manage the synchronization of the state of the UI such as loading screen, error message and other elements, with the REST API based backend response.

VI. PROCESSING PIPELINE

The system makes summaries from many YouTube videos. It does this by following a set of steps. First people give the system one or more YouTube video links. Then the system gets the words from the videos using yt-dlp subtitles or Whisper ASR if it cannot find subtitles[13].

The system makes the words, from the videos clean. Breaks them down into smaller parts[15]. This helps the system work well with videos that have a lot of talking. The system then uses the BART-large-CNN model to make a summary of each small part. After that the system puts all these summaries together to make one summary. The system then makes this summary even shorter and clearer.

The system can also translate the summary into other languages if people want it to. When the system is done it gives people the summary. Tells them if any of the video links they gave it were not working. The system gives them the YouTube video summary. Says which video links were bad. The system does all this to make a YouTube video summary.



VII. EXPERIMENTALS AND ANALYSIS

The thorough experiments were conducted on 3 types of video records. The three types are technical tutorials, news reporting and university lectures. Two measurement criteria used by us are ROUGE score and process time.

A. Dataset Description

The experimental evaluation was conducted using a collection of YouTube videos belonging to three primary categories: technical tutorials, news reports, and university lectures. The selected videos varied in duration and transcript length to simulate real-world usage scenarios.

The dataset included both manually generated subtitles and automatically generated transcripts obtained through speech recognition systems. Videos in English constituted the majority of the dataset, while multilingual translation experiments were conducted using selected samples translated into Hindi and Tamil. This diverse dataset enabled evaluation of the framework across varying content structures, vocabulary complexity, and transcript lengths.

B. Semantic retention analysis

Preserved Key Points (PKP) refers to the percentage of important informational units from the original transcript that are retained within the generated summary. These key points include central concepts, critical facts, definitions, and conclusions presented throughout the source material. PKP is used as an indicator of semantic retention and measures how effectively the summarization framework preserves essential information while reducing content length.

According to our results' analysis, there is a very clear difference. In comparison to other models used head truncation, this recursive pass method had 94% preserved PKP whereas other model only had 58% PKP. This indeed shows that the recursive pass has remembered some information from the end of the transcript.

C. Translation accuracy

The accuracy of translation of the generated transcript in the other two languages were tested. Though failed to correctly translate some difficult tech terms, the informational content is well-kept and even those non-native speakers can understand the central ideas of the video.

Video Count	Source Tokens	Summary Tokens	Inference Time(s)
1	4,200	320	11.2
2	8,500	410	18.4
3	14,000	550	26.1
5	22,000	680	42.5

VIII. LIMITATIONS AND FUTURE WORK

The designed framework's ability to automatically summarize videos and translate from

one language to another has been demonstrated. In particular, the video transcript quality has a strong influence on the model and any poorly written transcript will yield a poorer summary. Moreover, the use of transformer models has high computational requirements increasing inference speed when carried out on CPU hardware. Furthermore, it will take longer to process more transcripts and the quality of the summary will depend upon the accuracy of the translation done through external services.

There are a number of potentials for further improvements. Consequently, future versions of the framework must include real-time summarization, summarization aware of the speaker, and transformer fine-tuning. The framework may also be extended to support audio summarization while eliminating the requirement to feature transcripts as input data. Users can also have the option of regulating the length of the summarization adaptively. Sentiment-aware summarization can also become an important segment of the system. Moreover, a cloud implementation may also be worth developing.

IX. CONCLUSION

The current research describes the methodology of automated video summarization and multilingual translation with the use of BART transformer architecture. The designed framework includes such stages as the extraction of transcripts, hierarchical summarization, semantic aggregation, and translation to provide efficient knowledge extraction from video content. In contrast to other summarization methods that work only with one document, the framework is capable of analyzing multiple video transcripts and providing the output that helps the user get an overview of the subject based on several sources. In order to overcome the issue related to token limitations associated with transformers, a chunk-based hierarchical summarization is used to ensure proper processing of long documents.

At the same time, the multilingual translation feature provides a means to increase the convenience and accessibility of the summaries, thus enhancing the efficiency of knowledge acquisition. The choice of technologies for the backend framework and the frontend application (FastAPI and Flutter, respectively), also allows to ensure that the system will be scalable, practical, and easy-to-use. Thus, the designed system provides an effective solution to the issue of efficient video summarization and translation.

REFERENCES

1. R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 280–290, 2016
2. M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7871–7880, 2020.
3. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 5998–6008, 2017.
4. T. Wolf et al., "Transformers: State-of-the-art natural language processing," *Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pp. 38–45, 2020.
5. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *International Conference on Machine Learning (ICML)*, PMLR, pp. 28490-28508, 2023
6. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 4171–4186, 2019.

7. N. Zhang, J. Tang, and J. Lin, "Hierarchical transformers for long document summarization," *IEEE Access*, vol. 9, pp. 13204-13212, 2021.
8. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.
9. L. Wang, L. Li, and J. Rao, "Multi-document summarization via deep learning techniques: A survey," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2145–2170, 2022.
10. FastAPI Documentation. "FastAPI Framework, high performance, easy to learn, fast to code, ready for production." Available: https://fastapi.tiangolo.com/ [Accessed: June 2026].
11. Flutter Documentation. "Flutter UI toolkit for building beautiful, natively compiled applications." Available: https://flutter.dev/ [Accessed: June 2026].
12. Hugging Face Transformers Documentation. "State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX." Available: https://huggingface.co/docs/transformers/ [Accessed: June 2026].
13. YouTube Data API Documentation. "Google Developers: YouTube Data API v3." Available: https://developers.google.com/youtube/ [Accessed: June 2026].
14. B. Wu, A. Chang, and D. Zhou, "Quantization and training of low-bitwidth convolutional neural networks for computer vision and NLP," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4251-4265, 2022.
15. Yihong Liu, Haotian Ye, Leonie Weissweiler, Renhao Pei, Hinrich Schütz, "*Crosslingual Transfer Learning for Low-Resource Languages Based on Multilingual Colexification Graphs*".