A Thesis/Project/Dissertation Report

on

YOUTUBE TRANSCRIPT SUMMARIZER

Submitted in partial fulfillment of the

requirement for the award of the degree of

Master of Computer Applications

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Ms. Urvashi Suganth
Assistant Professor

Submitted By

SAKET KUMAR **(19SCSE1010116)**

AKASH KUMAR **(19SCSE1010472)**

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING / DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

DECEMBER, 202

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the YOUTUBE TRANSCRIPT SUMMARIZER, entitled **"CAPS…."** in partial fulfillment of the requirements for the award of the Bachelor Of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December 2021, under the supervision of Ms. Urvashi Suganth (Assistant Professor), Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

SAKET KUMAR (19SCSE1010116)

AKASH KUMAR (19SCSE1010472)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Ms. Urvashi Suganth

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of SAKET KUMAR (19SSCSE1010116) ,AKASH KUMAR (19SCSE1010472) has been held on __________________ and his/her work is recommended for the award of Bachelor Of Technology~~.~~

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013

Place: Greater Noida

Abstract

Enormous number of video recordings are being created and shared on the Internet through out the day. It has become really difficult to spend time in watching such videos which may have a longer duration than expected and sometimes our efforts may become futile if we couldn’t find relevant information out of it.

Summarizing transcripts of such videos automatically allows us to quickly look out for the important patterns in the video and helps us to save time and efforts to go through the whole content of the video.

The major challenge is to understand the visual semantics and convert it into a condensed format, such as caption or summary to save storage space, enables users to index, navigate, and help gain information in less time. We propose an innovative joint end-to-end solution, Abstractive and Extractive Summarization of video Sequences.

This provides a text-based video description and abstractive summary, enabling users to discriminate between relevant and irrelevant information according to their needs using various NLP and controllable text summarization models.

Automatic text summarization is the approach of generating the subset of the main text. This subset of the main text represents the complete text and the main idea of the text. Automatic Text summarization is also known as Test summarization. ATS is the important field of Natural Language Processing (NLP) and Data Mining (DM). This includes the abstractive and extractive summaries of the text.

Table of Contents

Title		Page No.
Candidates Declaration		**I**
Acknowledgement		**II**
Abstract		**III**
Contents		**IV**
List of Table		**V**
List of Figures		**VI**
Acronyms		**VII**
Chapter 1	**Introduction**	**1**
	1.1 Introduction	**2**
	1.2 Formulation of Problem	**3**
	1.2.1 Tool and Technology Used	
Chapter 2	**Literature Survey/Project Design**	**5**
Chapter 3	**Functionality/Working of Project**	**9**
Chapter 4	**Results and Discussion**	**11**
Chapter 5	**Conclusion and Future Scope**	**41**
	5.1 Conclusion	**41**
	5.2 Future Scope	**42**
	Reference	**43**
	Publication/Copyright/Product	**45**

List of Table

S.No.	Caption	Page No.
1		

List of Figures

S.No.	Title	Page No.
1	General Steps of text summarization	**9**
2	Flow chart for automatic text summarization	**17**
3	Use case diagram for automatic text summarization	**17**

Acronyms

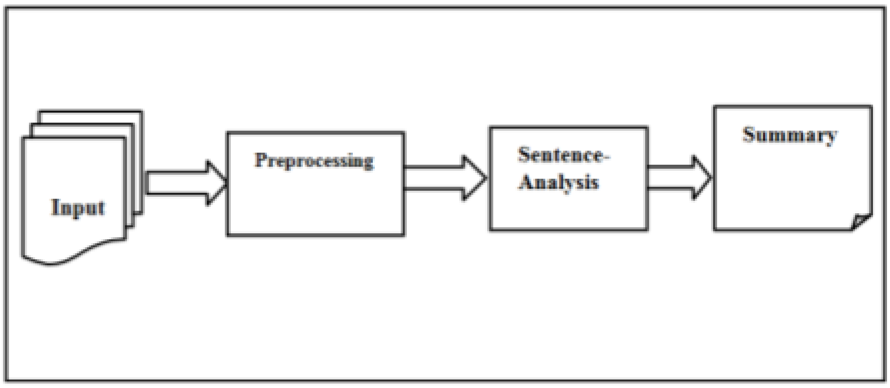
B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
MCA	Master of Computer Applications
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering
NAMAS	Neural attention model for abstractive sentence summarization
NLP	Natural Language Processing
RL	Reinforcement Learning
LSA	Latent semantic analysis
CNN	Convolution Neural Network
OOV	Out-of-vocabulary
RNN	Recurrent Neural Networks

CHAPTER-1

Introduction

Introduction

With the advancement of technology, the internet is accessible through various devices like smartphones, smartwatches and within the reach of common people. That leads to the accessibility of lot of information through world wide web (WWW). More information on the internet sometimes it becomes so difficult to select only required information from large texts. Due to the information, manual summarization of information is very challenging and also time-consuming task . Thus, we need an automatic text summarization system. Summary as “a text that is produced from one or more texts, that conveys the important information text(s), and that is no longer than half of the original text(s) and usually significantly less than that”. Summaries make the task of understanding the meaning of text easier. Text summarization helps user to manage vast amount of information by condensing document and include more relevant facts into them . Text summarization process contains three steps: analysis, transformation, . The input of the system can be single or multiple documents. It depends on the user requirement. The next step is preprocessing in this step stop words removed and tokenization performed.



[19] **Fig.1** General steps of Text Summarization

1.2

A typical flow of extractive summarization systems consists of:

1. Constructs an intermediate representation of the input text intending to find salient content. Typically, it works by computing TF metrics for each sentence in the given matrix.

2. Scores the sentences based on the representation, assigning a value to each sentence denoting the probability with which it will get picked up in the summary.

3. Produces a summary based on the top k most important sentences. Some studies have used Latent semantic analysis (LSA) to identify semantically important sentences.

Automatic text summarization is an exciting research area with several applications on the industry. By condensing large quantities of information into short, summarization can aid many downstream applications such as creating news digests, report generation, news summarization, and headline generation. There are two prominent types of summarization algorithms.

First, extractive summarization systems form summaries by copying and rearranging passages from the original text. Second, abstractive summarization systems generate new phrases, rephrasing or using words that were not in the original text. Due to the difficulty of abstractive summarization, the great majority of past work has been extractive.

The extractive approach is easier because copying large chunks of text from the source document ensures good levels of grammaticality and accuracy. On the other hand, sophisticated abilities that are crucial to high-quality summarization, such as paraphrasing, generalization, or the incorporation of real-world knowledge, are possible only in an abstractive framework. Even though abstractive summarization is a more challenging task, there has been a number of advances so far, thanks to recent developments in the deep learning area.

CHAPTER- 2

Literature Survey

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.
There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document.
In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarized to be useful. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document.
It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult and non-trivial task.

Various models based on machine learning have been proposed for this task. Most of these approaches model this problem as a classification problem which outputs whether to include a sentence in the summary or not. Other approaches have used topic information, Latent Semantic Analysis (LSA), Sequence to Sequence models, Reinforcement Learning and Adversarial processes.

In general, there are two different approaches for automatic summarization: extraction and abstraction.

The extractive approach

Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version.

Extractive summarization work by identifying important sections of the text cropping out and stitch together portions of the content to produce a condensed version. Thus, they depend only on the extraction of sentences from the original text.

Thus, they depend only on the extraction of sentences from the original text. Most of the summarization research today has focused on extractive summarization, once it is easier and yields naturally grammatical summaries requiring relatively little linguistic analysis. Moreover, extractive summaries contain the most important sentences of the input, which can be a single document or multiple documents.

Abstractive summarization

Abstractive summarization methods aim at producing summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text — parts of which may not appear as part of the original document, that conveys the most critical information from the original text, requiring rephrasing sentences and incorporating information from full text to generate summaries such as a human-written abstract usually does. In fact, an acceptable abstractive summary covers core information in the input and is linguistically fluent.

Thus, they are not restricted to simply selecting and rearranging passages from the original text.

Abstractive methods take advantage of recent developments in deep learning. Since it can be regarded as a sequence mapping task where the source text should be mapped to the target summary, abstractive methods take advantage of the recent success of the sequence to sequence models. These models consist of an encoder and a decoder, where a neural network reads the text, encodes it, and then generates target text.

In general, building abstract summaries is a challenging task, which is relatively harder than data-driven approaches such as sentence extraction and involves complex language modeling. Thus, they are still far away from reaching human-level quality in summary generation, despite recent progress using neural networks inspired by the progress of neural machine translation and sequence to sequence models.

CHAPTER-3

Functionality/Working of Project

An example is the work of Alexander et al, which proposed a neural attention model for abstractive sentence summarization (NAMAS) by exploring a fully data-driven approach for generating abstractive summaries using an attention-based encoder-decoder method. Attention mechanism has been broadly used in sequence to sequence models where the decoder extracts information from the encoder based on the attention scores on the source-side information. The code to reproduce the experiments from the NAMAS paper can be found here.

Example output of the attention-based summarization of Alexander et al. The heatmap represents a soft alignment between the input (right) and the generated summary (top). The columns represent the distribution over the input after generating each word.

Recent studies have argued attention-based sequence to sequence models for abstractive summarization can suffer from repetition and semantic irrelevance, causing grammatical errors and insufficient reflection of the main idea of the source text. Junyang Lin et al propose to implement a gated unit on top of the encoder outputs at each time step, which is a CNN that convolves all the encoder outputs, in order to tackle this problem.

Based on the convolution and self-attention of Vaswani et al., a convolutional gated unit sets a gate to filter the source annotations from the RNN encoder, in order to select information relevant to the global semantic meaning. In other words, it refines the representation of the source context with a CNN to improve the connection of the word representation with the global context. Their model is capable of reducing repetition compared with the sequence to sequence model outperforming the state-of-the-art methods. The source code of paper can be found here.

Other methods for abstractive summarization have borrowed the concepts from the pointer network of Vinyals et al to addresses the undesirable behavior of sequence to sequence models. Pointer Network is a neural attention-based sequence-to-sequence architecture that learns the conditional probability of an output sequence with elements that are discrete tokens corresponding to positions in an input sequence.

For example, Abigail See et al. presented an architecture called Pointer-Generator, which allows copying words from the input sequence via pointing of specific positions, whereas a generator allows generating words from a fixed vocabulary of 50k words. The architecture can be viewed as a balance between extractive and abstractive approaches.

In order to overcome the repetition problems, the paper adapts the coverage model of Tu et al., which was proposed to overcome the lacking coverage of source words in neural machine translation models. Specifically, Abigail See et al. defined a flexible coverage loss to penalize repeatedly attending to the same locations, only penalizing the overlap between each attention distribution and the coverage up to the current time step helping to prevents repeated attention. The source code for the model can be found here.

The Pointer-generator model. For each timestep in the decoder, the probability of generating words from the fixed vocabulary, versus copying words from source using a pointer is weighted by a generation probability p_{gen}. The vocabulary distribution and attention distribution are weighted and summed to obtain the final distribution. The attention distribution can be viewed as a probability distribution over the source words, that tells the decoder where to look to generate the next word. It is used to produce a weighted sum of the encoder hidden states, known as the context vector.

Other studies in abstractive summarization have borrowed the concepts from the reinforcement learning (RL) field to improve model accuracy. For example, Chen et al. proposed a hybrid extractive-abstractive architecture using two neural networks in a hierarchical way, that selects salient sentences using an RL guided extractor from the source and then rewrites them abstractively to generate a summary.

CHAPTER- 4

Results and Discussion

In other words, the model simulates how humans summarize long documents first using an extractor agent to select salient sentences or highlights, and then employs an abstractor — an encoder-aligner- decoder model — network to rewrite each of these extracted sentences. To train the extractor on available document-summary pairs, the model uses a policy-based reinforcement learning (RL) with sentence-level metric rewards to connect both extractor and abstractor networks and to learn sentence saliency.

Reinforced training of the extractor (for one extraction step) and its interaction with the abstractor.

The abstractor network is an attention-based encoder-decoder which compresses and paraphrases an extracted document sentence to a concise summary sentence. Moreover, the abstractor has a useful mechanism to help directly copy some out-of-vocabulary (OOV) words.

The convolutional extractor agent

The extractor agent is a convolutional sentence encoder that computes representations for each sentence based on input embedded word vectors. Further, an RNN encoder computes context-aware representation and then an RNN decoder selects sentence at time step t. Once the sentence is selected, the context-aware representation will be fed into the decoder at time t + 1.

Thus, the method incorporates the abstractive approach advantages of concisely rewriting sentences and generating novel words from the full vocabulary, whereas adopts intermediate extractive behavior to improve the overall model's quality, speed, and stability. The author argued model training is 4x faster than the previous state-of-the-art. Both source code and best pre-trained models were released to promote future research.

Other recent studies have proposed using a combination of the adversarial processes and reinforcement learning to abstractive summarization. An example is Liu et al. (2017), whose work proposes an adversarial framework to jointly train a generative model and a discriminative model similar to Goodfellow et al. (2014). In that framework, a generative model takes the original text as input and generates the summary using reinforcement learning to optimize the generator for a highly rewarded summary. Further, a discriminator model tries to distinguish the ground truth summaries from the generated summaries by the generator.

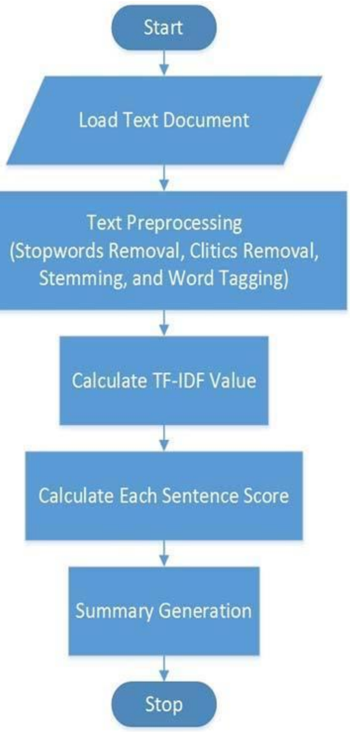
The discriminator is implemented as a text classifier that learns to classify the generated summaries as machine or human-generated, while the training procedure of generator is to maximize the probability of discriminator making a mistake. The idea is this adversarial process can eventually let the generator to generate plausible and high-quality abstractive summaries. The author provided supplementary material here. The source code is available in this github repo.

In short

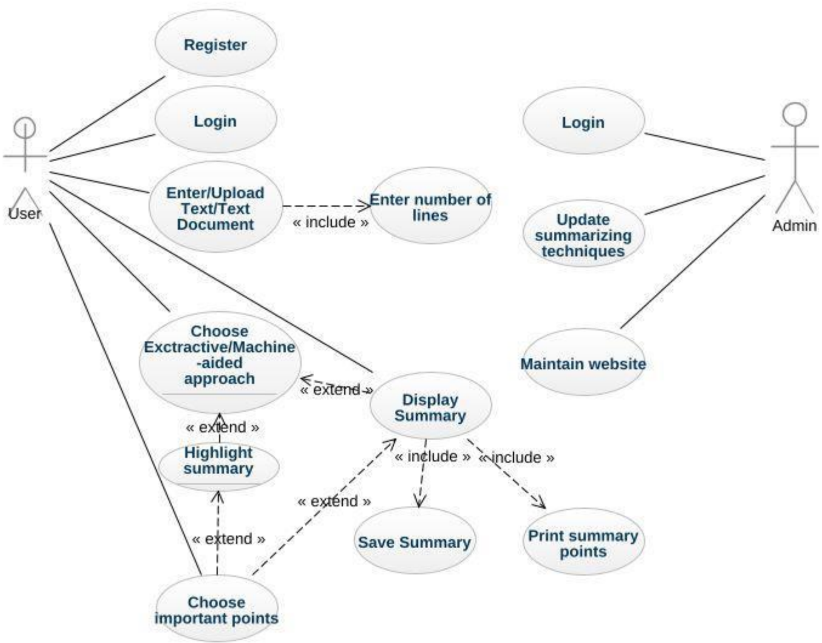
Automatic text summarization is an exciting research area with several applications on the industry. By condensing large quantities of information into short, summarization can aid many downstream applications such as creating news digests, report generation, news summarization, and headline generation. There are two prominent types of summarization algorithms.

First, extractive summarization systems form summaries by copying and rearranging passages from the original text. Second, abstractive summarization systems generate new phrases, rephrasing or using words that were not in the original text. Due to the difficulty of abstractive summarization, the great majority of past work has been extractive.

Flow Chart for automatic text summarization



Use Case Diagram



CHAPTER-5

Conclusion and Future Scope

As the availability of data in the form of text increasing day by day. It becomes so difficult to read the whole textual data in order to find the required information which is both difficult as well as a time-consuming task for a human being. So, at that time ATS performs an important role by providing a summary of a whole text document by extracting only the useful information and sentences. There are different approaches of text summarization. The real-world applications of text summarization can be: documents summarization, news and articles summarization, review systems, recommendation systems, social media monitoring, survey responses systems. The paper provides a literature review of various research works in the field of automatic text summarization. This research area can be explored more by looking in existing systems and working on different and new techniques of NPL and Machine Learning

Refrences

1. L. He, E. Sanocki, A. Gupta, and J. Grudin, "Comparing presentation summaries: Slides vs. reading vs. listening," Conference on Human Factors in Computing Systems (CHI 2000), April 1-6 2000, The Hague, Netherlands, pp. 177–184.

2. Youtube videos.
3. https://huggingface.co/docs/transformers/master/en/main_classes/pipelines#transformers.SummarizationPipeline
4. https://huggingface.co/docs/transformers/index

5. Wikipedia. (2020, December 15). *The crown (TV series)*. Wikipedia, the free encyclopedia. Retrieved December 15, 2020, from https://en.wikipedia.org/wiki/The_Crown_(TV_series)

6. HuggingFace. (n.d.). *Summary of the tasks — transformers 4.0.0 documentation*. Hugging Face – On a mission to solve NLP, one commit at a time. https://huggingface.co/transformers/task_summary.html#summarization

7. GitHub. (n.d.). *Abisee/CNN-dailymail*. https://github.com/abisee/cnn-dailymail

8. Papers With Code. (2019, October 29). *Papers with code – BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. The latest in Machine Learning | Papers With Code. https://paperswithcode.com/paper/bart-denoising-sequence-to-sequence-pre

9. Giacaglia, G. (2020, October 5). *Transformers*. Medium. https://towardsdatascience.com/transformers-141e32e69591

10. Wikipedia. (2019, August 25). *Transformer (machine learning model)*. Wikipedia, the free encyclopedia. Retrieved December 16, 2020, from https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*, 5998-6008.

12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … & Uszkoreit, J. (2020). An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

13. Wikipedia. (2019, October 10). *BERT (language model)*. Wikipedia, the free encyclopedia. Retrieved December 21, 2020, from https://en.wikipedia.org/wiki/BERT_(language_model)

14. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

15. HuggingFace. (n.d.). *BERT — transformers 4.1.1 documentation*. Hugging Face – On a mission to solve NLP, one commit at a time. https://huggingface.co/transformers/model_doc/bert.html

16. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.