Review Of Text Summarization Techniques using NLP For Transcripts And Articles

Bhushan Aher
Computer engineering
Modern Education Society's College of Engineering.
Pune, India.

Prof. Shobha Raskar
Computer engineering
Modern Education Society's College of Engineering. Pune, India

Rohit Ushir
Computer engineering
Modern Education Society's College of Engineering. Pune, India

Onkar Chaudhari
Computer Engineering
Modern Education Society's College of Engineering. Pune, India

Naresh Barule
Computer Engineering
Modern Education Society's College of Engineering Pune, India

Abstract

A vast number of video recordings and articles are generated and distributed on the internet every day. It has become quite difficult to devote time to watching films or reading articles that may last longer than expected, and If we are unable to get relevant information from them, our efforts may be futile. Summarizing such films' transcripts or summarising articles saves time and effort by allowing us to rapidly identify key patterns in the video. The research is entirely based on NLP, a cutting-edge artificial intelligence approach that aids in language recognition, summarization, and other tasks. This research focuses on the algorithms that aid in the summarization of the generated texts and transcripts. This paper shows how to summarise texts using various algorithms, as well as how to convert audio to text for films without transcripts and summarising it to offer a content summary utilising extractive and abstractive text summarization methodologies. Natural Language Processing (NLP) is an artificial intelligence area concerned with analysing, comprehending, and creating natural human languages so that computers can manage written and spoken human language without using computer-driven language.

Keywords—: NLP, artificial intelligence, text summarization techniques, abstractive, extractive

INTRODUCTION

We must first understand what a summary is before moving on to text summarization. A summary is a text that conveys information by combining one or more texts. The original text has a wealth of useful information. A condensed version is also available. The goal of artificial text summarization is to provide the original content in a condensed, semantically rich format. The most significant benefit of adopting a summary is that it cuts down on reading time. Text summarising approaches include extractive and abstractive summarization. deciding on key phrases Extractive summarising is the process of condensing paragraphs and other parts of the original content into a shorter version. The process of comprehending and articulating the important concepts in a document is known as abstractive summarization

Extractive and abstractive text summaries are the two types of text summaries. The primary idea of the book is simply communicated to the consumer through an inductive summary. The length of the generated summary is usually 5 to 10% of the original text. On the other hand, informative summary approaches provide a quick overview of the primary material. The useful summary should be 20 to 30% of the main content's length.

Main steps for text summarization: There are three main steps for summarizing documents. These are topic identification, interpretation and summary generation.

- The text's most important information is highlighted as a subject, cue words, and word frequency are examples of topic identification strategies. The mostly used and effective topic identification procedures are the ones that are based on sentence positioning.
- Summary production necessitates a comprehension of abstract summaries. This process combines many subjects to create a general content

I. LITERATURE REVIEW

I. REVIEW OF LITERATURE

In recent years, there has been a rise of important works on text summarising. A prior study focused on text summaries of single documents. When compared to prior techniques, technology has advanced, as has computational power, allowing for a faster, more effective, and more precise type of document processing.

To generate a summary of the reviews, Ravali boorugu and Dr. G. Ramesh proposed an extractive based technique that uses different text summaries types such as summarization depending on input, purpose, and output type. Single There are further options for document text summarization (SDTS) and multi-document text summarization (MDTS). In the category of input type dependence, they used single document and multi document techniques. There are generic, domain-specific, query-based approaches, and extractive and abstractive-based summarising techniques in the category of based on purpose. They discussed the most recent research in this topic and offered alternative ways for extractive summarization in the early phases.

A text summarising technique that was automatic was proposed by Adhika Widyasari, Edy Noersasongko, and Abdul Syukur. It has a summary that is generated automatically. From 2015 through 2019, they wanted to find and analyse methodologies, datasets, and trends in artificial text summarising research. This involves automatic text summary based on concepts. Single document, multi document, interactive, abstract, supervised learning, and unsupervised learning are the different types of machine learning techniques. They attempted to increase their performance and obtain quick results. A comparison of the most commonly used text summarising approaches and algorithms.

For summarizing the texts, Surabhi Adhikari, Rahul, and Monika suggested algorithms that are natural language processing based. It includes a variety of papers that show how machine learning may be used to summarise text. Using techniques like as Naive Bayes, Random Forest, and the support vector algorithm, we can classify spam on Twitter and generate EXT text summaries. Convolutional neural networks (CNN), recurrent neural networks, and other deep learning subjects (RNN). The k-nearest neighbour Newtonian technique, the artificial bee colony, and the human learning algorithm are among the other algorithms mentioned.

An examination into abstractive text summary algorithms was given by Parth Dedhia, Hardik Pachgade, and Meghana Naik. In contrast to extractive approach, abstractive technique is a more advanced summary technique that also produces grammatically acceptable summaries. They employed RNN to perform abstractive text summarization.Long Short Term Memory is one of the most common types of memory (LSTM).It also has a thorough overview of LSTM, including equations, input types, and model topologies.

NEED FOR TEXT SUMMARIZATION

The digital age's rapid proliferation of data needs the development of automatic text summarising technologies that enable users to quickly extract insights from it. If you want to extract specific information from an online news article or video, you may need to sift through the content and spend a significant amount of time weeding out unnecessary information before you find what you need. As a result, computerised text summarizers capable of extracting key information while removing inessential and unnecessary information are becoming increasingly significant. As a result, extraction becomes extremely important.

II. EXTRACTIVE TEXT SUMMARIZATION

This approach has two stages: pre-processing and processing. The act of presenting the original content in an organised fashion is known as pre-processing. The following stages are frequently included: a) Detecting sentence boundaries, b) Stop word deletion, and c) Stemming The processing stage finds and calculates the factors that define sentence relevance before utilising the weight learning approach to give weights to these factors. The feature-weight equation is used to get the final score for each sentence. For the final summary, the top-ranking sentences are chosen.

III. ABSTRACTIVE TEXT SUMMARIZATION

Abstractive summarization, on the other hand, examines the entire material and uses powerful natural language algorithms to recreate the original content in a new and optimised fashion. The newly produced content is more concise and, more importantly, includes the most significant information from the original text. Unlike extractive approaches, which might result in disfluent sentences, abstractive summaries provide fluent, grammatically correct sentences.

In our thoughts, we generate a semantic representation of the document. Then, using words from our extensive vocabulary (words we commonly use) that fulfil the semantics, we write a concise summary that includes all of the document's important points. As you can see, constructing this form of summarizer could be difficult because Natural Language Generation is required.

IV. EXTRACTIVE TEXT SUMMARIZATION WITH SUMMY

Many text summarising methods are included in the Sumy library. Instead of writing your own algorithm, you can import one.

1. LexRank: A sentence that is similar to many other sentences in the text has a higher likelihood of being crucial in LexRank algorithms. The Lex rank technique assumes that a statement is backed by other like sentences, and hence gets ranked higher. The more important the information in the summary paragraph is, the higher the rating.

2. The LSA technique is an unsupervised learning tool for extractive text summarization. It uses singular value decomposition (SVD) on a matrix of term-document frequency to extract semantically important words. The text will be summarised using these algorithms after importing the lsa from sumy and passing the document

3. Luhn: This algorithm's technique is based on TFIDF, according to Luhn (Term - frequency- inverse document frequency). It's useful when both low- and high-frequency words (stop words) aren't relevant. The highest-scoring sentences are used in the summary, and this is how sentences are graded.

4. TextRank: TextRank is a genism-developed extractive summarization approach. It works on the principle that words that appear more frequently have more weight. As a result, sentences with a lot of repetition are necessary. The system assigns a score to each sentence in the text based on this. The top-scoring sentences are included in the summary.

V. ABSTRACTIVE SUMMARIZATION WITH TRANSFORMERS

Hugging Face's transformers are compatible with a variety of popular models, including GPT-2, GPT-3, BART, Open AI, GPT, and T5.

Model T5 encoder-decoder transformer Text-to-text conversion is used for all linguistic concerns. Provide the text to the transformer for decoding. Before encoding the output to obtain the original text, it will carry out the steps.

A seq2seq paradigm that combines a bidirectional encoder and an auto-regressive decoder is the bidirectional and auto-regressive transformer, or BART.

The GPT-2 transformer, created by open AI with a process similar to BART, is another key competitor in text summarization.

Algorithm:

1. Submit a video or text file as an input.

2. Submit the request to the server.

3. Converting the input file to a video or text.

4. Using speech to text, create a transcript for the video.

5. Use the proper algorithm/method.

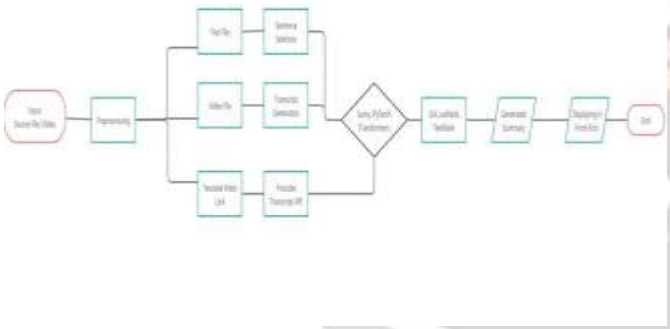
6. Send the summary version's front-end version..

7. Show the summary.

VI. TEXT SUMMARIZATION HISTORY

Extractive summarizers previously relied heavily on sentence evaluations in the original document. Statistical methodologies or linguistic strategies are used by the most frequent and current text summarising systems. The sentences are weighted using high frequency terms, standard keywords, the Cue Method, Title Method, and Location Method.

DATA FLOW



VII. IMPLEMENTATION

I. Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a Python programming environment for statistical natural language processing with data from humans (NLP).

It contains tokenization, parsing, categorization, stemming, tagging, and semantic reasoning text processing techniques.It also includes a recipe book and a book that explains the principles underlying the NLTK's fundamental language processing tasks, as well as graphical demonstrations of the data and sample data sets.

Steven Bird, Edward Loper, and Ewan Klein created the Natural Language Toolbox as an open source Python programming language toolkit for use in development and education.

It takes a hands-on approach to computational linguistics and Python programming basics, making it perfect for linguists with little or no programming expertise, engineers and academics interested in computational linguistics, students, and instructors.

The Penn Treebank Corpus, Open Multilingual Wordnet, Problem Report Corpus, and Lin's Dependency Thesaurus are among the more than 50 corpora and lexical sources included in NLTK.

II.SPACY

SpaCy is my go-to library for Natural Language Processing (NLP) operations.That, I believe, is true for the vast majority of NLP professionals out there!

SpaCy distinguishes out among the several NLP libraries available today. If you've used spaCy for NLP, you'll understand exactly what I'm talking about. If you're not familiar with spaCy's abilities,. you're about to be enthralled by how versatile and multi-functional this library is. The features it offers, the convenience of use, and the fact that the library is always maintained up to date are all factors that favour spaCy.

spaCy's Statistical Models

These are the models used by SpaCy's power engines. SpaCy can use these models to do things like tag parts of speech, recognise named entities, and parse dependencies. The different statistical models in spaCy, as well as their specifications, are given below:

• en core web sm: Multi-tasking in English CNN received OntoNotes training. 11 MB in size

• en core web md: Multi-tasking in English Glove vectors were trained on Common Crawl and CNN was trained on Onto Notes. 91 megabytes

• en core web lg: Multi-tasking in English GloVe vectors were trained on Common Crawl and CNN was trained on Notes. 789 megabytes

spaCy's Processing Pipeline

When working with spaCy, the initial step is to give a text string to an NLP object. This object is a collection of text pre-processing activities that the input text string must go through.

spaCy in Action

Let's get our hands dirty with spaCy right now. This section explains how to use spaCy to perform various NLP tasks. Part-of- Some of the most frequent NLP tasks are speech tagging, dependency parsing, and named entity recognition.

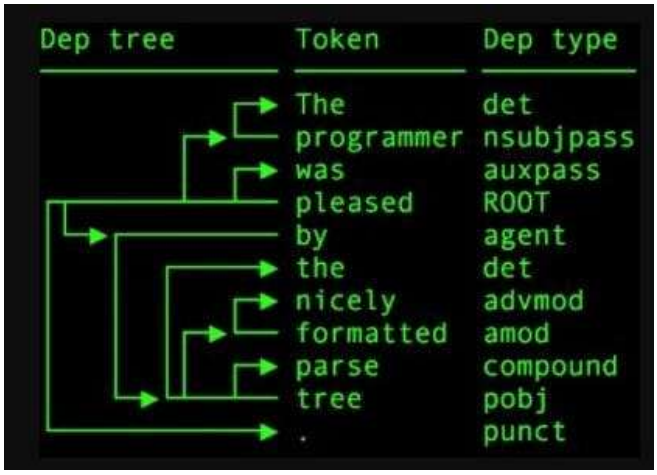
1. Part-of-Speech (POS) Tagging using spaCy

The parts of speech in English grammar describe the purpose of a word and how it is used in a sentence. Nouns, pronouns, adjectives, verbs, and adverbs are the most common components of speech in English.

POS tagging is the process of assigning POS tags to all of the words in a sentence automatically.It helps with downstream NLP activities including feature engineering, language understanding, and information extraction.

2. Dependency Parsing using spaCy

Every sentence has a grammatical structure, which can be extracted with the aid of dependency parsing. It can alternatively be viewed as a directed graph, with nodes corresponding to the words in the sentence and edges between nodes corresponding to the word dependencies.



3. Identified Entity Using spaCy for recognition

Entities are words or collections of words that represent information about everyday objects like people, places, and organisations. These things have official names.

4.Using spaCy for rule-based matching

SpaCy's arsenal now includes rule-based matching. With this spaCy matcher, you can apply user-defined rules to find words and phrases in the text.

It's basically a more advanced version of Regular Expressions.

The spaCy matcher looks for lexical aspects of the word, such as POS tags, dependency tags, Regular Expressions, on the other hand, look for words and phrases utilising text patterns.

VIII. CONCLUSION

Despite the fact that artificial text summarization is a relatively new topic, recent research has focused on developing trends in biomedicine, product assessments, education domains, emails, and blogs. This is due to the plethora of information on these issues available, particularly on the Internet. In the realm of natural language processing, automated summarization is a hot topic (Natural Language Processing It comprises constructing a summary of one or more texts automatically. Document summaries that are extractive or abstractive select a few key lines, chapters, or paragraphs from the source text mechanically. Text summarising techniques based on neural networks, graph theory, fuzzy logic, and clustering have all been successful in generating readable document summaries to some extent. Both extractive and abstractive methods have been investigated. The majority of summarising methods employ extractive techniques. The abstractive process is analogous to human summaries. Summarizing abstracts currently requires a lot of time and effort

ACKNOWLEDGMENT

This research and the related work would not have been feasible and possible without the assistance of the guides, Prof. Shobha Raskar and Prof. Jaya Mane.

REFERENCES

[1] Parth Rajesh Deshia, Hardik Pradeep Pachgade: Study on abstractive text summarization techniques, 2020. Emerging advances in information technology and engineering are the focus of this international conference. 2020.

[2] Rahul, Saurabhi Adhikari,Monika: NLP based machine learning approaches for text summarization, 2020. fourth international conference on computing methodologies and communication. 2020.

[3] Ravali Boorugu and Dr. G. Ramesh: a survey on NLP based text summarization for summarizing product reviews.Second international conference on inventive research and computing application. 2020.

[4] Dr. Gajula Ramesh, Dr.J.Somasekar, Dr. Karanam Madhavi, Dr. Gandikota Ramu, Best keyword set recommendations for building service-based systems International Journal of Scientific and Technology Research, volume 8, issue 10, October, 2019.

[5] Adhika Widyasari, Edy Noersasongko, abdul syukur.International conference on information and communication technology. 2019.

[6] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 Int. Conf.
Data Sci. Commun. IconDSC 2019, pp. 1–3, 2019, doi:
10.1109/IconDSC.2019.8817040

[7] Prabhudaas Janjanam and Pradeep Reddy: Text summarization-an essential study. Second international conference on computational intelligence in data science.2019.

[8] T. Jo, "K nearest neighbor for text summarization using feature similarity," Proc. - 2017 Int. Conf. Commun.
ICCCCEE 2017, pp. 1–5, doi:
10.1109/ICCCCEE.2017.78667059. Control. Comput. Electron. Eng. ICCCCEE 2017, pp. 1–5, doi:
10.1109/ICCCCEE.2017.78667059.

[9] B. Mutlu, E. A. Sezer, and M. A. Akcayol, "Multidocument extractive text summarization: A comparative assessment on features," Knowledge-Based Syst., vol. 183, p. 104848, 2019, doi:
10.1016/j.knosys.2019.07.019.

[10] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, "Query-oriented text summarization using sentence extraction technique," 2018 4th Int. Conf. Web Res. ICWR 2018, pp. 128–132, 2018, doi:
10.1109/ICWR.2018.8387248.

[11] L. Cuiling, "Text Automatic Summarization Generation Algorithm for English Teaching," 2016 Int. Conf. Intell. Transp. Big Data Smart City, p. 2016, 2017