The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 9 Issue 1, 2021
pp: 31 - 36

Follow Me Summarize an Article Using Adaptive Text Summarization API with IBM Cloud

[1]Sankala Priyanka, [2]Pailolu Gari Sarika, [3]Penugonda [4]Neelika Talari Indu
[1,2,3,4] UG Student, Department of ECE,
Dr K V Subba Reddy College Of Engineering For Women, Kurnool, Andhra Pradesh, India

Abstract
The process of creating a condensed version of a text document that retains important information and the overall meaning of the source text is known as text summarization. Automatic test summarization emerges as an important method for quickly and easily locating pertinent information in a large body of text. There are two categories of approaches to text summarization: abstractive and extractive. The two approaches to text summarization are thoroughly examined in this paper. The difficulty of reading it in a short amount of time increases with the amount of public content. Additionally, it takes time and effort to comprehend all of this information in a meaningful way. If summaries are available, reading them is an effective method for skimming the text. You can save time by using this application to find a summary of lengthy articles. The article can be accessed directly via the URL or website, or you can provide your article and receive a summary

1. Introduction

The task of producing a concise and fluent summary while maintaining key information and the overall meaning is known as summarization. The process of making a concise, accurate, and easy-to-understand summary of a lengthy text document is known as text summarization. For a text document, it is the process of distilling the most crucial information. The goal of text summarization is to make a summary of a large corpus with key points that describe the whole thing.

Applications that use text summarization are used by everyone. The platform that publishes articles on daily news, entertainment, and sports is the target audience for many of these applications. We prefer to read the article's summary before beginning to read the entire article due to our busy schedules. Reading a summary helps us determine our area of interest and provides a quick overview of the story.

There is a lot of textual material, and it keeps getting more and more every day. Consider the web, which is made up of web pages, news articles, status updates, blogs, and many other things. The only way we can get around the unstructured data is to use search and look at the results.

A lot of this text data needs to be broken down into shorter, more focused summaries that only include the most important details. This will allow us to better navigate the text and check to see if the larger documents contain the information we need.

Digital documents' textual information quickly turns into huge amounts of data. The vast majority of these numerous documents are unorganized: It has not been organized into conventional databases and has no restrictions. Because of this, processing documents is a pointless activity, primarily as a result of the absence of standards. Prior to moving on to the text summarization, we must first understand what a summary is. A summary is a shorter version of a text that is derived from one or more texts and conveys important information from the original text. The presentation of the source text as a shorter version with semantics is the aim of automatic text summarization. The most significant benefit of using a summary is that it cuts down on reading time. There are two types of text summarization techniques: abstractive and extractive summarization. Selecting significant sentences, paragraphs, etc. is an extractive summarization technique. by condensing portions of the original document into a shorter form. An abstract summary is one in which the main ideas in a document are understood and then expressed in straightforward natural language. Text summarization can be broken down into two distinct groups: revealing and instructive. Inductive summaries only convey to the reader the text's central idea. This kind of summarization typically takes up 5 to 10% of the main text. The informative summarization systems, on the other hand, provide concise information about the main text. The informative summary takes up 20 to 30 percent of the main text.

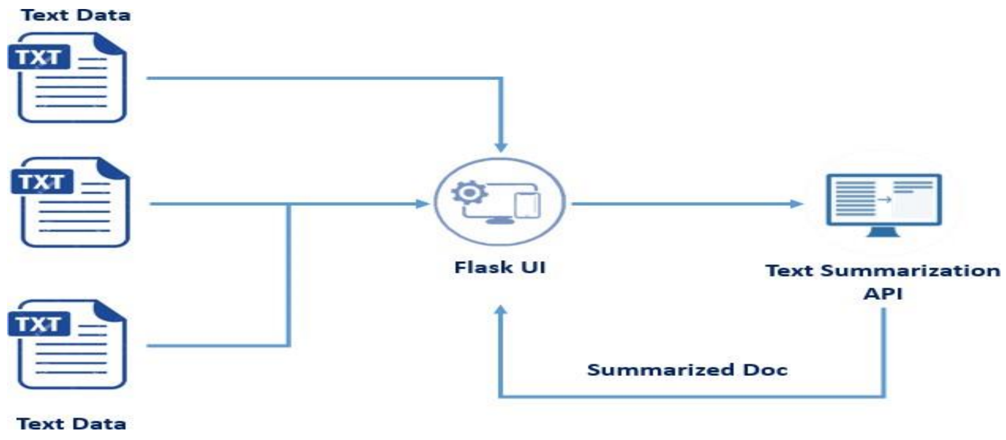
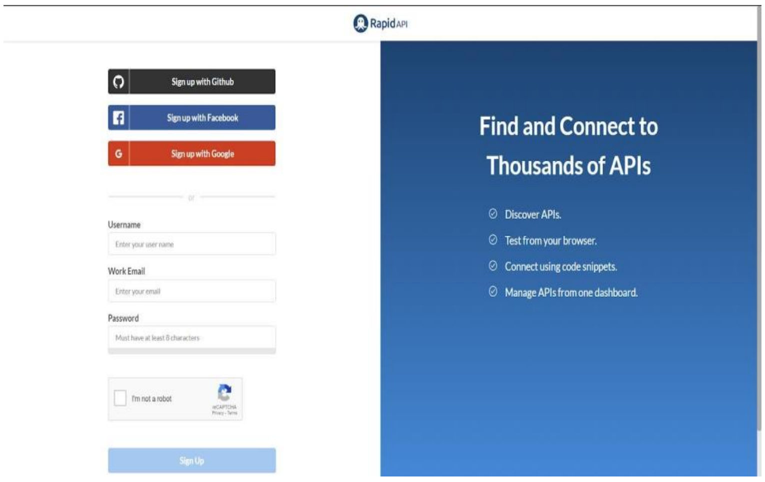


Fig.1 Objective block

2. Literature Review

Abstractive methods select words based on semantic understanding, even those words did not appear in the source documents. It aims at producing important material in a new way. They interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text.
It can be correlated to the way human reads a text article or blog post and then summarizes in their own word.
Input document → understand context → semantics → create own summary.
One approach to summarization is to extract parts of the document that are deemed interesting by some metric (for example, inverse-document frequency) and join them to form a summary. Algorithms of this flavor are called extractive summarization.
Original Text: Alice and Bob took the train to visit the zoo. They saw a baby giraffe, a lion, and a flock of colorful tropical birds.

Extractive Summary: Alice and Bob visit the zoo. saw a flock of birds.
Above we extract the words bolded in the original text and concatenate them to form a
Extractive methods attempt to summarize articles by selecting a subset of words that retain the most important points.
This approach weights the important part of sentences and uses the same to form the summary. Different algorithm and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.Another approach is to simply summarize as humans do, which is to not impose the extractive constraint and allow for rephrasing. This is called abstractive summarization.
Abstractive summary: Alice and Bob visited the zoo and saw animals and birds
The limited study is available for abstractive summarization as it requires a deeper understanding of the text as compared to the extractive approach.
Purely extractive summaries often times give better results compared to automatic abstractive summaries. This is because of the fact that abstractive summarization methods cope with problems such as semantic representation, inference and natural language generation which is relatively harder than data-driven approaches such as sentence extraction.
There are many techniques available to generate extractive summarization. To keep it simple, I will be using an unsupervised learning approach to find the sentences similarity and rank them. One benefit of this will be, you don‘t need to train and build a model prior start using it for your project.
It‘s good to understand Cosine similarity to make the best use of code you are going to see. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Since we will be representing our sentences as the bunch of vectors, we can use it to find the similarity among sentences. Its measures cosine of the angle between vectors. Angle will be 0 if sentences are similar.
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



Find and Connect to Thousands of APIs

- Discover APIs.
- Test from your browser.
- Connect using code snippets.
- Manage APIs from one dashboard.

Fig.2 API

3. Proposed System

For text summarization, recent deep learning techniques have demonstrated promising results. The problem of text summarization has been framed as a sequence-to-sequence learning problem, which has sparked the development of strategies based on the application of deep learning techniques to automatic machine translation.

The task of creating a headline or brief summary of a few sentences that summarizes the most important ideas in an article or passage is known as abstractive text summarization. Mapping an input sequence of words from a source document to a target sequence of words known as a summary is another natural way to frame this task.

Using Sequence-to-Sequence RNNs and Beyond for Abstractive Text Summarization, 2016

These deep learning approaches to automatic text summarization can be thought of as abstract methods because they learn a language generation model specific to the source documents and generate a completely new description. Abstractive summarization is now possible thanks to the recent success of sequence-to-sequence models, in which recurrent neural networks (RNNs) read and freely generate text. Get To The Point: 2017: Synthesis Using Pointer-Generator Networks

When compared to extractive methods, the results of deep learning methods are not yet considered state-of-the-art, but impressive results have been achieved on limited problems, such as creating headlines for news articles, that are comparable to or better than those produced by abstractive methods.

The approach promises that the models can be trained from beginning to end without specialized data preparation or submodels, and that the models are completely data-driven without specialized vocabulary or expertly pre-processed source documents. A data-driven approach to abstractive sentence summarization is what we propose. The structure of the model is simple; It scales to a large amount of training data and can be easily trained from beginning to end. 4.1 Text Summarization Using Deep Learning A sequence to sequence model has three main components:

Encoder is a bi-directional LSTM layer that takes data from the original text and applies it. The above is highlighted in red. Since it is a bi-directional LSTM, the bi-directional LSTM updates its hidden state based on the current word as well as the words it has read in the past. It reads one word at a time.

Decoder is a unidirectional LSTM layer that produces word-by-word summaries. Once the signal indicates that the entire source text has been read, the LSTM decoder can begin working. To generate the probability, it makes use of the encoder's information in addition to what it has already written distribution over the next word. The Decoder is shown in yellow above with the probability distribution in green.

Attention Mechanism — Encoder and Decoder are the building blocks here but historically encoder decoder architecture in itself without attention wasn‘t very successful. Without attention, the input to decoder is the final hidden state from encoder which can be a 256 or 512 dimension vector and if we imagine this small vector can‘t possibly have all the information in it so it became a information bottleneck. Through attention mechanism, the decoder can access the intermediate hidden states in the encoder and use all that information to decide which word is next. Attention is shown in blue above. Attention is a pretty tricky concept so please don‘t sweat if my brief description here was confusing. You can read more about attention through my blog here.

As Pointer Generator paper shows that the above architecture is good enough to get started but the summaries created by it has two problems:

The summaries sometimes reproduce factual details inaccurately (e.g. Germany beat Argentina 3– 2). This is especially common for rare or out-of-vocabulary words such as 2–0.

The summaries often repeat themselves. (e.g. Germany beat Germany beat Germany beat)

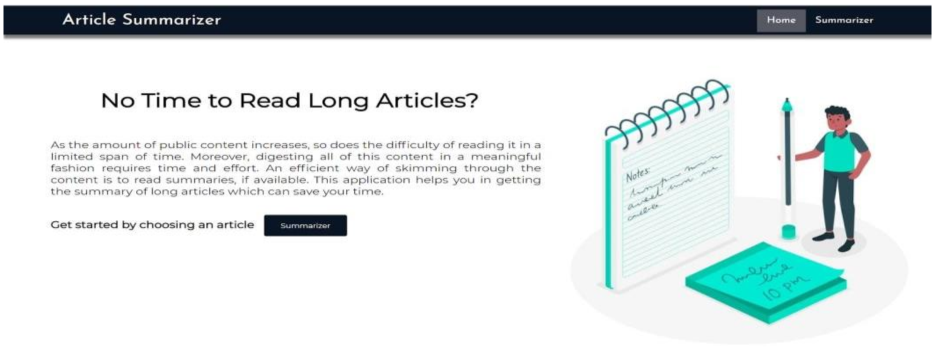
The pointer generator model solves these issues by creating a pointer mechanism that allows it to switch between generating text vs copying as is from source. Think of the pointer as a probability scalar between 0 and 1. If it is 1 then the model does abstractive generation of a word and if 0 it copies the word attractively.

Compared to the sequence-to-sequence-with-attention system, the pointer-generator network has several advantages:

The pointer-generator network makes it easy to copy words from the source text.

The pointer-generator model is even able to copy out-of-vocabulary words from the source text. This is a major bonus, enabling us to handle unseen words while also allowing us to use a smaller vocabulary (which requires less computation and storage space).

The pointer-generator model is faster to train, requiring fewer training iterations to achieve the same performance as the sequence-to-sequence attention system



No Time to Read Long Articles?

As the amount of public content increases, so does the difficulty of reading it in a limited span of time. Moreover, digesting all of this content in a meaningful fashion requires time and effort. An efficient way of skimming through the content is to read summaries, if available. This application helps you in getting the summary of long articles which can save your time.

Get started by choosing an article

Fig.3 Proposed Method

4. Conclusion

We‘ve observed that due to the nature of news headlines, the model can generate good headlines from reading just a few sentences from the beginning of the article. Although this task serves as a nice proof-of-concept, we started looking at more difficult datasets where reading the entire document is necessary to produce good summaries. In those tasks training from scratch with this model architecture does not do as well as some other techniques we‘re researching, but it serves as a baseline. We hope this release can also serve as a baseline for others in their summarization research.

References

1. S.L. Marie-Sainte, N. Alalyani, Firefly algorithm-based feature selection for Arabic text classification. J. King Saud Univ.-Comput. Inf. Sci. (2018)Google Scholar
2. L.M. Abualigah, A.T. Khader, M.A. Al-Betar, Multi-objectives-based text clustering technique using K-mean algorithm, in 2016 7th International Conference on Computer Science and Information Technology (CSIT) (IEEE, 2016), pp. 1–6Google Scholar
3. A.M. Azmi, N.I. Altmami, An abstractive Arabic text summarizer with user- controlled granularity. Inf. Process. Manage. 54(6), 903–921 (2018)CrossRefGoogle Scholar
4. D.R. Radev, E. Hovy, K. McKeown, Introduction to the special issue on summarization. Comput. Linguist. 28(4), 399–408 (2002)CrossRefGoogle Scholar
5. A. Qaroush, I.A. Farha, W. Ghanem, M. Washaha, E. Maali, An efficient single document Arabic text summarization using a combination of statistical and semantic features. J. King Saud Univ.-Comput. Inf. Sci. (2019)Google Scholar
6. J. Xu, G. Durrett, Neural extractive text summarization withsyntactic compression. arXiv preprint arXiv:1902.00863 (2019)
7. L.M.Q. Abualigah, Feature selection and enhanced krill herd algorithm for text document clustering. studies in computational intelligence (2019)CrossRefGoogle Scholar
8. R. Belkebir, A. Guessoum, TALAA-ATSF: a global operation-based arabic text summarization framework, Intelligent Natural Language Processing: Trends and Applications (Springer, Cham, 2018), pp. 435–459CrossRefGoogle Scholar