JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Automatic Text Summarisation in the perspective of context preservation

[1]Abinash Tripathy, [2]Sundareswar Pullela, [3]Mounika Kola and [4]Sai Raj Kasa

[1,2,3,4]Department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam

Abstract:

The automatic text summarization system generates a summary, i.e., short-length text that includes all the important information of the document without losing any of the original contexts of the text. With the large amounts of information being generated and stored on servers and made easily accessible through the internet, text summarization is an important and potent tool to interpret all the textual information. The summary can be generated through extractive as well as abstractive methods. Abstractive methods are highly complex as they need extensive natural language processing. Therefore, the research community is focusing more on extractive summaries, trying to achieve more coherent and meaningful summaries. For a decade, several extractive approaches have been developed for an automatic summary generation that implements several machine learning and optimization techniques. Despite all the proposed methods, the generated summaries are still far away from human-generated summaries. Most research focuses on the extractive approach. It is required to focus more on the abstractive and hybrid approaches. In this paper, we address the automatic summarization task. Recent research works on extractive-summary generation employ some heuristics, but few works indicate how to select the relevant features. We will present a summarization procedure based on the application of trainable Machine Learning algorithms which employs a set of features extracted directly from the original text.

Keywords: Extractive Summarization, Abstractive Summarization, Natural Language Processing, Machine Learning

1. Introduction

Humans communicate with each other using words and text. The way that humans convey information to each other is called Natural Language. Every day humans share a large quality of information with each other in various languages such as speech or text. However, computers cannot interpret this data, which is in natural language, as they communicate in 1s and 0s. The data produced is precious and can offer valuable insights. Hence, you need computers to be able to understand, emulate and respond intelligently to human speech [1]. Natural Language Processing or NLP refers to the branch of Artificial Intelligence that allows machines to understand human language. Natural language processing helps computers communicate with humans in their own language and scales other language-related tasks. For example, NLP makes it possible for computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important. Today's machines can analyse more language-based data than humans, without fatigue and in a consistent, unbiased way. Considering the staggering amount of unstructured data that is generated every day, from medical records to social media, automation will be critical to fully analyze text and speech data efficiently. NLP is important because it helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics [2].

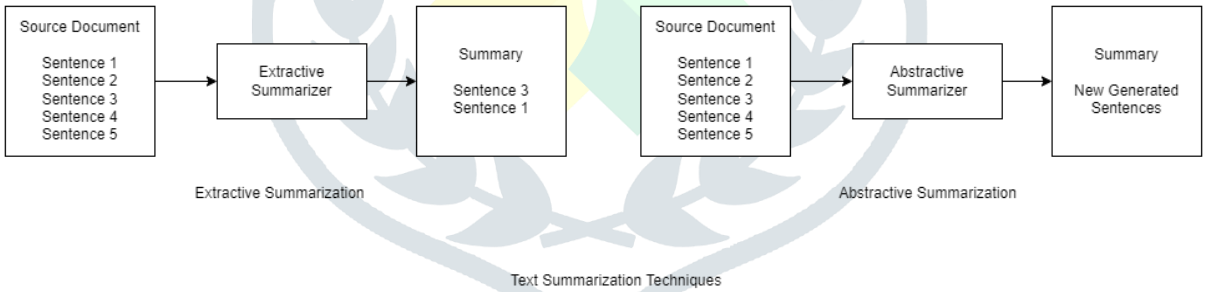
NLP is one of the ways that people have humanized machines and reduced the need for labour. It has led to the automation of speech-related tasks and human interaction. Some applications of NLP include as follow:

1. **Text Summarization**: Automatic summarization is pretty self-explanatory. It summarizes text, by extracting the most important information. Its main goal is to simplify the process of going through vast amounts of data, such as scientific papers, news content, or legal documentation. There are two ways of using natural language processing to summarize data: extraction-based summarization – which extracts key phrases and

creates a summary, without adding any extra information – and abstraction-based summarization, which creates new phrases paraphrasing the source. This second approach is more common and performs better.

2. **Speech Recognition:** Speech recognition technology uses natural language processing to transform spoken language into a machine-readable format. Speech recognition systems are an essential part of virtual assistants, like Siri, Alexa, and Google Assistant, for example. However, there are more and more use cases of speech recognition in business. For example, by adding speech-to-text capabilities to business software, companies can automatically transcribe calls, send emails, and even translate.
3. **Chatbots & Virtual Assistants:** Chatbots and virtual assistants are used for automatic question answering, designed to understand natural language and deliver an appropriate response through natural language generation.
4. **Auto-Correct:** Natural Language Processing plays a vital role in grammar checking software and auto-correct functions. Tools like Grammarly, for example, use NLP to help you improve your writing, by detecting grammar, spelling, or sentence structure errors.
5. **Translation Tools:** Tools such as Google Translate, Amazon Translate, etc. translate sentences from one language to another using NLP.
6. **Sentiment Analysis:** Sentiment analysis is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers [3] .

Automatic Text Summarization (ATS) is becoming much more important because of the huge amount of textual content that grows exponentially on the Internet and the various archives of news articles, scientific papers, legal documents, etc. Manual text summarization consumes a lot of time, effort, and cost, and even becomes impractical with the gigantic amount of textual content. Researchers have been trying to improve ATS techniques since the 1950s. ATS approaches are either extractive, abstractive, or hybrid. The extractive approach selects the most important sentences in the input document(s) and then concatenates them to form the summary. The abstractive approach represents the input document(s) in an intermediate representation and then generates the summary with sentences that are different from the original sentences. The hybrid approach combines both the extractive and abstractive approaches.



Extractive Summarization:

Extractive summaries are formulated by extracting key text segments (sentences or passages) from the text, based on a statistical analysis of individual or mixed surface-level features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. The "most important" content is treated as the "most frequent" or the "most favourably positioned" content. Such an approach thus avoids any efforts on deep text understanding. They are conceptually simple and easy to implement.

Abstractive Summarization:

An Abstractive summarization attempts to develop an understanding of the main concepts in a document and then express those concepts in clear natural language. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document.

2. Literature Survey

2.1 What is Automatic Text Summarization

The Automatic Text Summarization (ATS) system is to produce a summary that includes the main ideas in the input document in less space and to keep repetition to a minimum. The ATS systems help the users to get the main points of the original document without the need to read the entire document The users will benefit from the automatically produced summaries and they will save a lot of time and effort. These systems can be classified as single-document or multi-document summarization systems. ATS systems are designed by applying one of the text summarizations approaches: extractive, abstractive, or hybrid. ATS is one of the most challenging tasks in Natural Language Processing (NLP) and Artificial Intelligence (AI) in general [4] .

2.2 Types of Automatic Text Summarizer

There are many techniques in the ATS (Automatic Text Summarization). Most of these techniques are usually used in the pre-processing phase of an ATS system like noise removal, sentence segmentation, word tokenization, etc. Semantic and parsing based techniques are then performed on the pre-processed sentences. There are several uses of parsing techniques in the ATS processing phase like constructing the text graph models and in the post-processing phase like sentence compression and sentence merging. There are many parsing techniques like syntactic parsing, text chunking, semantic parsing, and shallow semantics. There are many semantic-based techniques like word sense disambiguation, anaphora resolution. These techniques cover all the ATS system phases. It is mainly focused on extractive-based text summarization methods such as term frequency-inverse document frequency method, cluster-based method, text summarization using neural networks, graph-based method, text summarization with fuzzy logic, latent semantic analysis method, machine learning approach, query-based summarization [4].

2.3 History of Automatic Text Summarization

In this survey, the popular and important work was done in the field of single and multiple document summarizations. This is about the method-based approaches for text summarization. These method-based approaches include term-based frequency method, graph-based method, time-based method, separation and merging-based method, semantic dependency method, topic-based approaches, discourse-based approaches, Latent Semantic-based approaches, approaches based on lexical chain, and approaches based on fuzzy logic [5].

2.4 Unsupervised techniques of text summarization

The system of automatic text summarization using unsupervised learning was proposed by Devihosur et al 2017. They have used a technique of simplified Lesk calculation to assess the significance and importance of sentences in information, an online semantic lexicon wordnet is utilized. Here the system architecture consists of three stages that are data pre-processing, evaluation of weights, summarization, and the overall representation of automatic text summarization using Natural Language Processing consists of input document, pre-processing, Lesk algorithm, generation of summary, and Lesk algorithm is connected to WordNet [6] .

2.5 Domain based and multiple document summarization

The system that investigates the problem of building the domain-based single and multiple document Amharic text summarization is proposed by Mekuria et al 2017. In this paper they have suggested that multi-document summarization targets to compress the most important information from a set of documents to produce a short summary and also suggested that text summarization can be performed based on input, purpose, and output. They have proposed a system that solves the existing problem by developing the combinations of extractive and abstractive-based approach on single and also multiple document input from the user. In this text summarization they discussed about the importance of page rank algorithm as it is used in finding out sentence score and weights of sentence in document and also concluded that the proposed model summarizes only text documents but in future a system would be proposed to develop text summarization for all types of documents including images, graphs, pictures, and videos [7] .

2.6. Text summarization

The text summarization as it solves the problems of presenting the information needed by a user in compact form. Here it is explained different latent semantic analysis-based summarization algorithms and two algorithms were proposed by the authors. They have used ROGUE scores to evaluate the performance of the algorithms. They mainly discussed the latent semantic analysis approach and sentence selection approach for text summarization. One of the

algorithms proposed by authors produced best scores and both the algorithms performed equally well on summarizing English and Turkish document texts [8].

3. Methodology

In this section, we explain and elaborate the working mechanism of out text summarization model in detail. After defining the summarization task at hand, we specify the pre-processing steps involved in cleaning and preparing the input data to get it ready for the model to work on and finally we give a view of the Neural Network structure of our proposed seq2seq model.

3.1 Problem description

The dataset consists of N number of data entries, that consist of the source text and represented as X and the target summary of the article, professionally written by human summarizers as our target output Y. Both the source document and target summaries consist of words and sequence of sentences. The deep learning task at hand is to generate legible summary of the source article without losing the key statements and essence of the source material.

3.2. Pre-processing

The pre-processing involves cleaning and preparing the source article text data for the proposed model to train on. The steps involved in pre-processing are

i. Merging the paragraphs of the article into a single sentence to ensure that the complete article is being loaded into the memory without any loss of information for the algorithm to have complete context without leaving behind any key points and keywords.
ii. Splitting the article into distinct sentences for them to be evaluated separately for the value of information embedded in them.
iii. Removing symbols and other special characters from individual sentences
iv. Splitting each sentence into distinct word in the order of occurrence and storing them sentence-wise in its own data-structure.
v. Removing stop words from the input sentences to ensure that the model is not affected by their frequency and skew its accuracy.

3. 3. Model configuration parameters

The model on initialization is also supplied with a few parameters to such as a sorted list of most frequent words, their frequency, and their index position of occurrence for both the source article and the summary test to assist in tuning the hyperparameters.

3.4. Seq2Seq Model

The neural network layer architecture on both encoder and decoder and they will calculate the intermediate states through the input elements. We represent the output of lth layer as $\mathbf{z}^l = (z^l_1, \ldots, z^l_m)$ for encoder and $\mathbf{h}^l = (h^l_1, \ldots, h^l_n)$ for decoder. Each layer consists of a one dimensional convolution and a non-linearity. If a decoder has one layer with kernel width being k, then its output h1i will compress the information of k input elements. To enlarge the length of input elements, we stack over each other, for example, stacking 6 blocks with k = 5 could represent 25 input elements. If needed, non-linearities enable our model to deal with the entire input sequence or only a few elements. The computational advantage of our model is that it conducts a parallel computation which is much more efficient than the traditional RNN model computed one element by one element. As mentioned in Section 1, to represent a sequence with n words, CNNs only need O(nk) operations, while RNNs need O(n) operations. However, our hierarchical CNN model is not that efficient than traditional CNN model as ours has to stack CNN layers to represent a sequence more expressively, while a traditional CNN model only needs one layer to explore a whole sequence. In each convolution kernel, parameters are $W \in \mathbb{R}^{2d\,X\,od}$, $b_w \in \mathbb{R}^{2d}$. The input is represented as $X \in \mathbb{R}^{o\,X\,d}$, which is a matrix having o input elements with the dimension being d. Then the input is mapped by the layer to get the output being a single element $Y \in \mathbb{R}^{2d}$ with its dimension twice of that of the input. Then the o outputs elements will be fed to the subsequent layers. We leverage the gated linear units (GLU) [9] as non-linearity, which is applied on the output represented as $Y = [A\ B] \in \mathbb{R}^{2d}$

4. Experiments and Results

We introduce the datasets used in out experiment in detail, along with the experiment setup and the evaluation metric. Then we demonstrate the efficiency of our seq2seq model.

4.1 Training Dataset

The training dataset consists of 2226 data points containing articles and their professionally made summaries in the genres of business, technology, politics, entertainment and sports. This dataset is generated from the articles and summaries found on Kaggle at https://www.kaggle.com/datasets/pariza/bbc-news-summary. 80% of the dataset is used for training set.

4.2 Evaluation Techniques

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation [10]. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans. This paper introduces four different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S included in the ROUGE summarization evaluation package and their evaluations. Three of them have been used in the Document Understanding Conference (DUC) 2004, a large-scale summarization evaluation sponsored by NIST.

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}$ ($gram_n$) is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. ROUGE-N is a recall-related measure because the denominator of the equation is the total sum of the number of n-grams occurring at the reference summary side. Note that the number of n-grams in the denominator of the ROUGE-N formula increases as we add more references. This is intuitive and reasonable because there might exist multiple good summaries.

4.3 Experimental Results

The results of our implementation of seq2seq algorithm are as follows, using ROGUE-1, ROGUE-2, ROGUE-L evaluation parameters.

Model	ROGUE-1	ROGUE-2	ROGUE-L
Seq2seq	38.26	18.72	35.11

The ROGUE evaluation metrics show that the single word occurrence i.e., ROGUE-1 is 38% meaning there are about 38% of the words matching between the actual summary and the generated summary. The ROGUE-2 is the 2-word pair occurrence in the generated summary as compared to the actual summary which is 18.72% and the ROUGE-L metric that looks for the longest subsequence between the actual and generated summary is 35.11%. These values are a slight improvement over the current seq2seq models that exist for automatic text summarization.

5. Conclusion

We can conclude that the seq2seq model for text summarization is a valid and perfectly capable model for generating short summaries that can come close in context preservation and accuracy to human made summaries of about 10-12 words in length. This can be further improved through adding deeper neural network techniques to our framework such as CNN and RNN.

References

[1] S. Liu, "Automatic Text Summarisation," [Online]. Available: https://towardsdatascience.com/automatic-text-summarisation-ccc98d2b323f.

[2] SAS Institute Inc., "Natural Language Processing (NLP): What it is and why it matters," [Online]. Available: https://www.sas.com/en_in/insights/analytics/what-is-natural-language-processing-nlp.html#:~:text=NLP%20is%20important%20because%20it,speech%20recognition%20or%20text%20analytics.

[3] R. Wolff, "NLP Applications & Examples in Business," [Online]. Available: https://monkeylearn.com/blog/natural-language-processing-applications/.

[4] W. &. S. C. &. R. A. &. M. H. El-Kassas, "Automatic Text Summarization: A Comprehensive Survey," *Expert Systems with Applications*, 2020.

[5] W. E. Z. M. G. H. W. Q. Z. S. Congbo Ma, "Multi-document Summarization via Deep Learning Techniques: A Survey," 2021.

[6] N. R. Pratibha Devihosur, "Automatic Text Summarization Using Natural Language Processing," 2017.

[7] G. &. J. A. Mekuria, "Automatic Amharic Text Summarization using NLP Parser," *International Journal of Engineering Trends and Technology.*, 2017.

[8] M. &. A. F. &. C. I. Ozsoy, "Text summarization using Latent Semantic Analysis," 2011.

[9] Y. Dauphin, A. Fan, M. Auli and Grangier, "Language Modeling with Gated Convolutional Networks," *In Proceedings of the 34th International Conference on Machine Learning*, 2017.

[10] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Proceedings of the ACL Workshop: Text Summarization Braches Out*, 2004.