International Journal for Rapid Research in Engineering Technology & Applied Science
Vol IX Issue IV April 2023
ISSN Online :- (2455-4723)

Text Summarization

Abhishek Patidar, Aditya Sharma, Aditya Jain, Alfaiz Khan
Acropolis Institute of Technology and Research, Indore, M.P.
Abhishekpatidarcs19@acropolis.in, Adityajaincs19@acropolis.in, Adityasharmacs19@acropolis.in, alfaizkhan203d14@acropolis.in

Abstract: Text summarization is a process of extracting or collecting important information from original text and presents that information in the form of summary. Text summarization has become the necessity of many applications for example search engine, business analysis, market review. Summarization helps to gain required information in less time. This paper is an attempt to summarize and present the view of text summarization from every aspect from its beginning till date. The two major approaches i.e., extractive and abstractive summarization is discussed in detail. The technique deployed for summarization ranges from structured to linguistic. In Indian many languages also the work has being done, but presently they are in infancy state. This paper provides an abstract view of the present scenario of research work for text summarization.

Keywords: Text Summarization, Natural Language Processing, Extractive Summary, Abstractive Summary.

I Introduction

The amount of data and information on the Internet continues to increase every day in the form of web pages, articles, academic papers, and news items. In spite of the abundance, it is difficult to find information needed efficiently because most information is irrelevant to a particular user's needs at a particular time. Therefore, the need for automatic summarization and extraction of relevant information continues to be a productive research area within natural language processing. Automatic summarization helps extract useful information while discarding the irrelevant. It can also improve the readability of texts, and decrease the time that users spend in searching. Researchers have been trying to perform suitable automatic text summarization since the late 1950s. The goal is to generate summaries, combining the main points in a readable and cohesive way, without having unuseful or repeated information [1]. Text summarization methods usually extract important words, phrases or sentences from a document and use these words, phrases, or sentences to create a summary. Text summarization can be classified into single document and multi-document summarization, depending on the number of input documents. Single document text summarization only accepts one document as input [2], whereas multi-document summarization accepts more than one document, where each document is related to the main topic. Meaningful information is extracted from each document and then gathered together and organized to generate a summary [3] [4]. Extractive summarization chooses important sentences from a document and combines them to create a summary without changing the original sentences.

Abstractive summarization first converts the important sentences extracted from a document into an understandable and coherent semantic form, and then generates the summary from this internal form, thus potentially changing the original sentences. Hybrid text summarization combines both extractive and abstractive summarization. Generally, the processing architecture of all automatic text summarization systems contains three steps. The first is preprocessing to usually identify words, sentences and other structural components of the text. The second is processing, which converts the input text to a summary by using a text summarization method. The third is post-processing, which fixes problems in the created draft summary [5]. Several recent surveys have been published on automatic text summarization, and most focus on extractive summarization techniques [1] because abstractive summarization is difficult and requires comprehensive Natural Language Processing (NLP). Most state-of-the-art papers focus on a part of automatic text summarization such as focusing on one approach, or on one specific domain in automatic text summarization. Mahajani et al. [6] recommended using a hybrid system that combines extractive and abstractive summarization approaches to leverage their respective advantages. Therefore, the goal of this survey is to present various methods in text summarization to help readers understand how a good summary can be generated by combining more than one approach or method. The present review is organized into three sections: a brief introduction to text summarization, text summarization approaches, and the conclusion of the paper. The architectures, advantages, and disadvantages of the approaches are included in detail in the second part.

II TEXT SUMMARIZATION FEATURES

Text summarizers identify and extract key sentences from the source text and concatenate them to form a concise summary. A list of features as discussed below can be used for selection of key sentences in Table 1 [2, 4].

TABLE 1 TEXT SUMMARIZATION FEATURES [2, 4]

Features	Description
Term Frequency	Salient terms provided by statistics are based on term frequency, thus salient sentences are those words that occur repeatedly [4]. The frequently occurring word increases score of sentences. The most common measure widely used to calculate the word frequency is TF IDF [2].
Location	It depends on the intuition that important sentences are located at certain position in text or in paragraph, such start or end of a paragraph [4]. First and last sentence of paragraph has greater chance to be included in summary [2].
Cue Method	Effect of positive or negativity of word on the sentence weight to indicate importance or key idea such as cues: “in summary”, “in conclusion”, “the paper describes” [2].
Title/ Headline word	Words in the title and heading of a document that occur in sentences are positively related to summarization [2]. Words that appear in the title are also indicative of the topic or subject of the document [4].
Sentence length	Keeps in view the size of summary. Generally, very long and very short sentences are also not suitable for summary [2].
Similarity	Similarity can be calculated with linguistic knowledge. It indicates similarity between the sentence and title of the document, and similarity between the sentence and remaining sentence of the document [2].
Proper noun	For document summarization sentences having proper nouns are important. Like, name of a person, place or organization [2].
Proximity	The distance between text units where entities occur is a determining factor for establishing relations between entities [2].

III TECHNIQUES USED FOR TEXT SUMMARIZATION

Conceptually, there are two approaches for text summarization, which are extractive and abstractive summarization. Within each approach, there are many methods and techniques. Every approach has some advantages and disadvantages.

A. Extractive Summarization

The architecture for extractive summarization includes three steps:
Pre-processing, Processing, and Post-processing, as shown in Figure 1. Pre-processing performs tasks such as tokenization and extraction of sentences and paragraphs. The processing step creates appropriate representation of the input text using techniques such as N-grams and graphs, or performs neural network based feature extraction and encoding [2] followed by scoring each sentence depending on input text representation [7]. After that, the approach chooses highly ranked sentences and links them together as a summary [7] [8]. Post-processing involves steps such as changing pronouns with their antecedents, and rearranging the extracted sentences [9].

Advantages and Disadvantages for Extractive Summarization

Since extractive summarization depends on directly generating the summary from the text without changing the content sentences in any way, it is faster and simpler [10].

The disadvantage of this approach is that it is not the same as how humans write the summary. The approach usually results in the reduction of semantic quality and cohesion because of wrong connections between sentences in the generated summary, making the flow stilted and unnatural [11]. The generated summary may not be accurate enough, and not cover all important content sentences in the input document [12]. However, if the output summary is long enough, the issue of missing significant sentences may not arise. But it may contain unnecessary parts that may not be needed in the summary, making it longer than necessary [9].

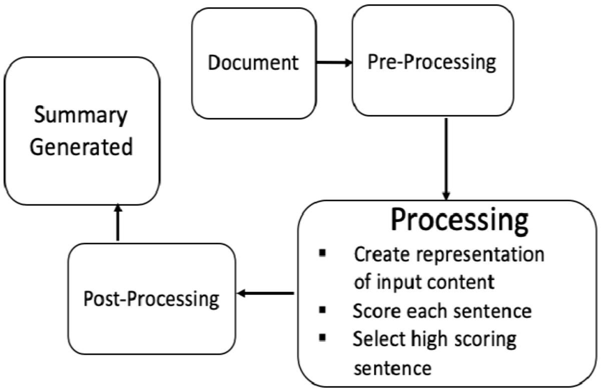

Fig. 2. Extractive Text Summarization Architecture

B. Extractive Summarization Methods

There are various extractive summarization methods for selecting and scoring sentences. These include Conceptual, Linguistic, Statistical, Machine Learning methods, Fuzzy logic, and Deep learning.

1) Concept Methods:

Such a method produces a summary of the concepts present in a document that can be found in external information repositories like WordNet [13] and Wikipedia. Depending on the concepts extracted, the important sentences are identified based on connection to external information bases instead of words. From the external information base's scores, a graph model or vector is built to produce the connection between the sentences and the concepts.

The concept methods of summarization can cover a very large number of concepts because WordNet and Wikipedia are large repositories. However, such a method depends on high quality similarity measurements to decrease redundancies in calculating concept-sentence correlations [12].

2) Linguistic Methods:

A linguistic method focuses on the relationships between words and concepts to get to the meaning to generate the summary. Abstractive summarization includes some level of semantic processing, so that, it can be thought also of as a linguistic method.

Linguistic methods are useful because they try to understand the meaning of every sentence in a document. However, this method is time-consuming requiring high effort. A linguistic method also needs a large amount of memory for saving additional linguistic repositories such as WordNet. It needs powerful processors for complicated linguistic processing [14].

3) Statistical Methods:

Such methods use statistical features of the document to identify the important pieces of the text. In a statistical method, a sentence is selected based on features like word frequency, position of the sentence, indicator phrases, title, location, and other features regardless of the meaning of the sentence. The method calculates the scores of the selected sentences and chooses a few highest scoring sentences to create the summary [15] [9].

Baxendale [16] focused on the position of sentences in his summarization research. He found that the best locations for the most important parts of the paragraph are the first and last sentences. He examined 200 paragraphs, and concluded that the topic sentences are included the first sentence of the paragraph in around 85% of the cases while 7% it was in the last sentence of the paragraph. Statistical methods do not take into account the meaning of sentences, and as a result, they may produce low-quality summaries. Statistical methods require low memory and processor capacity [15].

4) Machine Learning Methods:

The idea behind machine learning is to use a training set of data to train the summarization system, which is modeled as a classification problem. Sentences are classified into two groups: summary sentences and non-summary sentences [17]. The probability of choosing a sentence for a summary is estimated according to the training documents and corresponding extractive summaries [18]. The steps for ranking sentences in Machine Learning methods are extracting features from a document, and feeding those features to a machine learning algorithm that gives an output score as a value [12]. Some of the common machine learning methods used for text summarization are linear regression, na¨ıve Bayes, support vector machine, artificial neural networks, and fuzzy logic [19] [15].

A large training data set is necessary to improve the choices of sentences for the summary [12]. A simple regression model may be able to produce better output when compared with the other classifiers [15]. Every sentence in the basic text must be labeled as a summary or non-summary, demanding extensive manual work to generate extractive summaries for training [12].

5) Fuzzy Logic Based Methods:

Such text summarization methods use a multiple-valued system known as fuzzy logic. Fuzzy logic produces an efficient way to provide feature values for sentences that are between the two logical values “one” and “zero”, because these two values often do not represent the “real world” [20]. For ranking sentences, the first step is to choose a group of features for every sentence. The second step is to apply the fuzzy logic concept to get a score for every sentence based on the importance of the sentence. This means every sentence has a score value from 0 to 1, depending on the features [1].

Fuzzy logic represents uncertainties in selecting a sentence as a ‘fuzzy’concept [20]. However, one negative factor is redundancy in the selected sentences for the summary, impacting the quality of the generated summary. Therefore, a redundancy removal technique is required to enhance the quality of the generated summary [21].

6) Deep Learning Methods:

Kobayashi et al. [22] suggest a system for text summarization using document level similarity depending on embeddings. They assume that an embedding of a word represents its meaning, a sentence considered as a bag-of-words, and a document as a bag-of-sentences. They formalize their task as the problem of maximizing a submodular function which is identified by a negative summation of closest neighbors’ distance on embedding distributions. They found that the document level similarity is more complex in meaning compared with sentence-level similarity. In Chen et al. [23], they suggest automatic text summarization that used a reinforcement learning algorithm and Recurrent Neural Network (RNN) model with a single document. By using a sentence level selective encoding technique, they select the significant features, generating the summary sentences.

In deep learning methods, the network could be trained depending on the reader’s style, and the features can be changed depending on the user’s requirement. However, it is difficult to identify how the network generates a decision [12]. Recent research shows that using a combination of various methods helps produce a better summary by taking the advantage of the strengths of the individual methods [12], [24], [25], [26]. For instance, Moratanch and Chitrakala [12] used a combination of both graphs and concept based methods to generate summaries. Mao et al. [26] combine three different methods of supervised learning with unsupervised learning to create a summary for a single document. Combining different features together may also help produce better outcomes during the calculation of the weights of sentences [1].

C. Abstractive Summarization

Abstractive text summarization creates a summary of a document by extracting and understanding the concepts present in the text during processing [27] [28]. It paraphrases the text, but does not directly copy from the content of the original text [29]; instead it creates new sentences that better reflect the human way of constructing summaries. As a result, the input content needs more analysis for abstractive summarization [30].

The processing architecture for abstractive summarization is presented in Figure 3. It is composed of Pre-processing, Processing that contains two sub-steps, and Post-processing. For example, Moratanch and Chitrakala create an internal semantic representation and then use various techniques to create summaries [31].

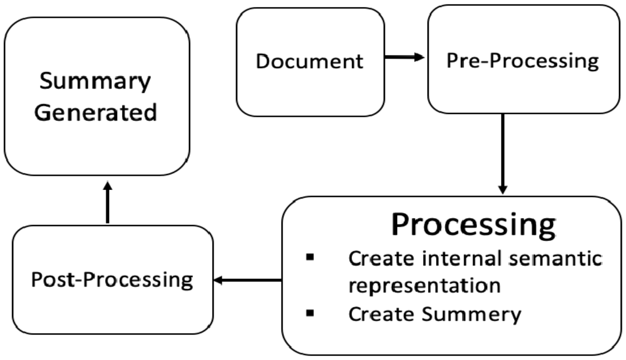

Fig. 2 Abstractive Text Summarization Architecture

Advantages and Disadvantages of Abstractive Summarization:

Some of the advantages of abstractive summarization are that the generated summary is created to be different from the original text by using more resilient expressions based on paraphrasing [11]. So, the generated summary is likely to be closer to a human summary [32]. Compared to extractive summarization, abstractive summarization can decrease the amount of generated text and produce a summary that removes any redundancy, obtaining a concise and expressive summary [33].

Some of the disadvantages of abstractive summarization are that it is difficult to perform high-quality abstractive summarization [11]. It is difficult to create a good abstractive summary because it needs to use natural language generation technology, which still needs a lot of progress [34]. Current abstractive summarization approaches seem to create repetitions in word choice. In addition, good abstractive summarization should be able to explain why it creates new sentences in the summary, which is difficult to do. The approach is also unable to handle out-of-vocabulary words properly [11].

Furthermore, the approach's ability is constrained by what underlying semantic representation it uses, because a system cannot generate a summary if its representation scheme cannot capture necessary nuances and details [9].

D. Abstractive Summarization Methods

Abstractive summarization methods can be classified into three categories, which are structure-based, semantics-based, and deep learning-based methods [35]. A structure-based approach uses pre-defined structures such as trees, graphs, templates, rules, and ontologies. Therefore, it recognizes in the input document, the most important information, and then using the previously mentioned structures, it generates the abstractive summary. The semantics based construction of the input document generates a semantic representation by using information items, semantic graphs, and predicate-argument structures. Then, using approaches in natural language generation, it generates the abstractive summary [35].

1) Structure-Based Methods:

Templates-Based Methods: Human summaries tend to use certain characteristic sentence structures in some domains. These can be identified as templates. To perform abstractive summarization, the information in the input document is used to fill slots in appropriate pre-defined templates based on the input document's style [36]. Text snippets can be extracted using rules and linguistic cues, to fill template slots [35].

Rule-based Methods: To find the important concepts in the input document and use them in the generated summary, one needs to define rules and categories. To use these methods, one needs to classify the input document based on the concepts and terms present in it, create relevant questions depending on the domain of the input document, answer the questions by detecting the concepts and terms in the document, and feed the answers into patterns to generate the summary [35].

Tree-based Methods: To perform abstractive summarization in tree-based methods, one needs to cluster similar sentences in the input that have related information, and then work with these sentence clusters for the summary [35]. Similar sentences are formulated into trees, parsers are applied to build the dependency trees, a popular tree based representation. Then, a process such as pruning linearization is used to produce trees in order to generate summary sentences from some of the sentence clusters [35].

Graph-Based Methods: The authors in [37] used a graph model which contains nodes, with each node expressing a word and positional information, that is connected to other nodes. The structure of sentences is represented by directed edges. The steps for the graph method contain constructing a textual graph representing the source document and generating abstractive summary. Such a method explores and scores many sub-paths in the graph in order to create the abstractive summary [37].

Ontology-Based Methods: Ontology-Based methods generate abstractive summarization from an input document by utilizing an ontology [38]. Many documents in specialized domains are connected to a domain specific ontology, and can be mapped to such an ontology. The mapping is traversed to generate a summary [39].

2) Semantics-Based Methods :

These methods process the input text to obtain semantic representations such as information items, semantic graphs, and predicate-argument structures. The representation is processed to provide the abstractive summarization by performing word choices, and stringing the words together using verb and noun phrases [35]. The authors in [40] perform multi-document abstractive summarization by extracting predicate-argument structures from the input text by performing semantic role labeling. By using a semantic similarity measurement, they cluster the semantically similar predicate-argument structures in the text, and then score the predicate-argument structures using feature weighting. Finally, they use language generation approaches to create sentences from predicate-argument structures.

3) Deep Learning-Based Methods:

Recent research in generating abstractive summarization has used deep sequence to-sequence learning [11]. In many different NLP tasks such as machine translation, sequence-to-sequence learning has led to good results [34]. RNNs with attention models have accomplished promising results in text summarization. Deep learning-based methods are being actively explored, and researchers are trying to solve many deep learning issues. Some of the issues are the inability to handle out-of-vocabulary words, and generation repeated phrases or words [11].

Abstractive summarization has recently concentrated on utilizing deep learning methods, particularly for short text summarization [41]. It is a recommendation by some to use more than one method to produce a better abstractive summary by taking advantage of each method. Using different text summarization algorithms on the same input document will produce different summaries. To generate a better summary, it is necessary to combine outputs of various text summarization algorithms rather than using single algorithms [42].

Usually, structure-based methods are used as extractive techniques for generating hybrid summaries while semanticsbased or deep learning-based methods are used to generate abstractive summaries [35]. For instance, one of these methods can be used in the pre-processing step to select the important phrases, and the other method to create the abstractive summarization [35]. The authors in [41] suggest a combination of semantics-based data transformation, followed by a encoderdecoder deep learning models for abstractive summarization

IV Conclusions

Text summarization is growing as sub – branch of NLP as the demand for compressive, meaningful, abstract of topic due to large amount of information available on net. Precise information helps to search more effectively and efficiently. Thus text summarization is need and used by business analyst, marketing executive, development, researchers, government organizations, students and teachers also. It is seen that executive requires summarization so that in a limited time required information can be processed. This paper takes into all about the details of both the extractive and abstractive approaches along with the techniques used, its performance achieved, along with advantages and disadvantages of each approach. Text summarization has its importance in both commercial as well as research community. As abstractive summarization requires more learning and reasoning, it is bit complex then extractive approach but, abstractive summarization provides more meaningful and appropriate summary compare to extractive. Through the study it is also observed that very less work is done using abstractive methods on Indian languages, there is a lot of scope for exploring such methods for more appropriate summarization.

References

[1] N. Nazari and M. Mahdavi, “A survey on automatic text summarization,” Journal of AI and Data Mining, vol. 7, no. 1, pp. 121–135, 2019.

[2] M. Joshi, H. Wang, and S. McClean, “Dense semantic graph and its application in single document summarisation,” in Emerging Ideas on Information Filtering and Retrieval. Springer, 2018, pp. 55–67.

[3] S. Modi and R. Oza, “Review on abstractive text summarization techniques (atst) for single and multi documents,” in 2018 International Conference on 5 Computing, Power and Communication Technologies (GUCON). IEEE, 2018, pp. 1173–1176.

[4] S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, “A comprehensive survey on text summarization systems,” in 2009 2nd International Conference on Computer Science and its Applications. IEEE, 2009, pp. 1–6.

[5] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A comprehensive survey,” Expert Systems with Applications, p. 113679, 2020.

[6] A. Mahajani, V. Pandya, I. Maria, and D. Sharma, “A comprehensive survey on extractive and abstractive techniques for text summarization,” in Ambient Communications and Computer Systems. Springer, 2019, pp. 339–351.

[7] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in Mining text data. Springer, 2012, pp. 43–76.

[8] J. Zhu, L. Zhou, H. Li, J. Zhang, Y. Zhou, and C. Zong, “Augmenting neural sentence summarization through extractive summarization,” in National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017, pp. 16–28.

[9] V. Gupta and G. S. Lehal, “A survey of text summarization extractive techniques,” Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 3, pp. 258–268, 2010.

[10] A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, “Multidocument text summarization-a survey,” in 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE, 2016, pp. 331–334.

[11] L. Hou, P. Hu, and C. Bei, “Abstractive document summarization via neural model with joint attention,” in National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017, pp. 329–338.

[12] N. Moratanch and S. Chitrakala, “A survey on extractive text summarization,” in 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP). IEEE, 2017, pp. 1–6.

[13] G. A. Miller, “Wordnet: a lexical database for english,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.

[14] Y. Ko and J. Seo, “An effective sentence-extraction technique using contextual information and statistical approaches for text summarization,” Pattern Recognition Letters, vol. 29, no. 9, pp. 1366–1371, 2008.

[15] M. Gambhir and V. Gupta, “Recent automatic text summarization techniques: a survey,” Artificial Intelligence Review, vol. 47, no. 1, pp. 1–66, 2017.