A Survey of Automatic Text Summarization

Mak Chun Chung, Desmond
The Hong Kong Polytechnic University

Abstract

Automatic Text Summarization is a popular branch in Natural Language Processing (NLP). The purpose of automatic text summarization is to analyze the given documents and generate a summary of them. Automatic Text Summarization is constructed by different factors, such as the classification methods, summary types, the style of outputs, types of summarization task, the language of summarization task, etc. [1] *Until now, many researchers have invented various ways to handle the above tasks. The need for automatic text summarization is also why it becomes a hot and essential branch in NLP. Such as the webpage* [2]*, the E-email conversation* [3]*, and social media* [4]*, their information needs to be summarized for different usages. Therefore, this survey will introduce and explain the various methods and categories in automatic text summarization and the different types of applications implemented by the summarization techniques.*

1. Introduction

Automatic Text Summarization generates a summary of one or multiple document(s) using different approaches to read and analyze them. As early as the end of the 1950s, computer scientist H. P. Luhn has started to investigate the prototype of automatic text summarization. [5] Despite the then limitation of the hardware and lacking relevant research, his research still brought many brainstorming ideas to the future study of automatic text summarization. Since then, automatic text summarization becomes one of the NLP tasks in which the researchers are interested. During these years of research, various challenges in automatic text summarization are found. For example, generating abstraction summary [6], and the issues of multiple document summarization [1]. Thereby, the researchers are spending many times to investigated different solutions to these challenges. Those methods are constructed by the traditional algorithms as well as state-of-art techniques such as deep learning.

The implementation of automatic text summarization consists of a few essential factors: number of documents, classification methods, summaries types, output styles, types of the task, and summary lingual. Each of their category and importance in the automatic summarization task will be explained in section 2.

Many Internet applications need to apply automatic text summarization to improve their service and business, such as E-email service and social media. In section 3, the application of automatic text summarization will be introduced, and their implementation will be briefly explained as well.

As mentioned, the approaches of automatic text summarization consist of traditional methods (e.g., statistics-based) and state-of-art methods (e.g., Artificial Neural Network-based). This paper collected different well-known approaches, and they will be described in sections 4 to 8.

2. Factors of Automatic Text Summarization

Although text summarization is an easily understandable task, it is not easy to get a satisfactory result. Summarizing a summary needs to consider different factors, such as different types of documents requiring different strategies. After decades of research, the crucial factors in automatic text summarization have been indicated. They are the number of documents, summarization methods, summaries types, output styles, types of the task, and summary lingual. [1] They can also be regarded as the considerations of designing a unique automatic summarization system, which depends on what type the target reader is, what the summary is used for, where the summary is used, etc.

2.1. Number of Documents

In automatic text summarization, the number of documents is one of the essential factors. There are two categories of it, which are single-document summarization and multi-document document summarization.

Single-document summarization means that each summarization task is only given one single document to generate the result. Multiple-document summarization is much complex than single. It generates a summary from multiple provided documents, so many issues and challenges may be ignored in single-document summarization, but we can find them in multiple. There are four reasons to make multiple-document summarization more complicated than single which are high redundancy in a group of documents (e.g., repeated meaning of the sentence), temporal dimension in a group of documents (e.g., E-mail conversation, news), compression ratio (i.e., summary size with respect to the size of the group of documents), and co-reference problem. [7]

2.2. Classification Summarization Methods

The result of automatic text summarization can be generated by two summarization methods: extractive and abstractive.

An extractive summary means that the summary is generated by selecting the critical and most represented sentences and concatenating them. An abstractive summary is complicated than an extractive summary, the words in generated summary are different from the words in the given documents, but the meaning is the same. Generally speaking, the abstractive summary is generated by the machine or system after reading the given

documents. Therefore, abstractive summary generation is much complicated than extractive summary generation.

2.3. Summaries Type

There are two types of summaries produced by automatic text summarization, which are generic and query-focused.

The generic summary represents that the summary is constructed by the abstract idea or information of the given documents. The query-focused means that the summary is based on the given query or topic to find the relevant information from the givens documents. The query-focused summary is also known as a topic-focused or user-focused summary. The summary information in this type is based on the user's imputed quires or some specific topics to conclude.

2.4. Output Style

The output style describes what kind of summary the system would like to present to the reader. There are two styles in automatic text summarization, which are indicative and informative.

The indicative summaries present the summary that is about the topic of given documents. On the contrary, the informative summaries show the summary that concludes the whole information of provided documents. In general, the indicative abstract typically objectively highlights the basics of a document, whereas the informative abstract summaries all the significant components of the document in a compressed form. [8].

2.5. Types of Task

The automatic text summarization usually can be conducted either supervised or unsupervised. These two types are the main categories of summarization tasks.

Supervised summarization requires the training data composed of the sample documents and their label (i.e., the document's summary). Depending on different requirements and purposes, choose suitable classification methods, such as Support Vector Machine (SVM), Naïve Bayes Classifier, and Recurrent Neural Network (RNN). By using the training data to train the classifier. In such case, the classification defines each sentence of documents to a two-class classification problem (i.e., whether this sentence will be selected as a part of the summary).

In contrast, unsupervised summarization does not require the labeled data to train the classifier. This kind of summarizing system can be put the target articles into it and get the summary directly. In unsupervised summarization, clustering (i.e., an unsupervised machine learning method) is widely used to achieve it.

2.6. Summarization Lingual

The lingual is a concern to the summarization task. Since the target documents may consist of different languages or the reader wants to see another language of the summary that is different from the original languages of the documents, there are three types of summarization to provide the different lingual pattern to the summaries. They are multi-lingual, mono-lingual, and cross-lingual summaries.

Multi-lingual summarization means that the summaries are generated from the documents constructed by multiple languages. The generated summaries are also constructed by multiple languages, and the languages are the same as the given documents. Mono-lingual summarization generates the summaries of which languages are the same as the given documents. The last one is cross-lingual summarization. It generates the summaries, which language different from the given documents.

3. Applications of Text Summarization

Automatic text summarization can organize an abstract by selecting the critical information from the given documents or articles. After a decay's development, the technology has been improved gradually. Therefore, various industries and applications have adopted it to improve the services and analyze their users' needs. The following will introduce three common types of automatic summarization that can be applied to enhance Internet applications and services. They are the E-email, webpage, and social media applications. Since the Internet grows extremely fast, the data on the Internet becomes vast as well. The above applications and services are harder to satisfy the users' desires than the 10 or 20 years ago. However, automatic text summarization can show the compressed important information from the text data, and it is an excellent feature to help improve the above three services.

3.1. E-mail Threads Summarization

E-mail service has become one of the common conversation tools in the world because of its practical, high efficiency, and low cost. It has been developed for about 50 years. During these years, different services have also been developed belonging to e-mail, such as detecting spam emails and email classification. It also can summarize the made past business decision to help the companies to analyze the business and the future development of the company. [1]

However, summarizing the emails is different from summarizing the articles or general documents. Because emails are used for conversations, each of them is constructed of the email threads (i.e., the dialogue consists of serval emails sent from different people).

Some researchers have proposed the corresponding methods to handle the e-mail threads challenge in summarization in these years, such as sentence extraction, an e-mail threads summarization method presented in 2004 [3]. It aims to extract the crucial sentences from the given threads of emails, then concatenating them as a final summary. This approach consists

of the traditional NLP methods, such as Term Frequency-Inverse Document Frequency (TF-IDF), cosine similarity of TF-IDF, classification of the type of sentence, etc. According to various ways to extract the important sentences from the threads of emails and summarize them.

3.2. Web-page Summarization

The search engine is one of the most common and frequently used services on the Internet, such as Google Search, Bing, DuckDuckGo, etc. They have their significant algorithm to present the searching result based on the inputted keywords. However, the Internet is growing fast, and the data on the Internet also is growing fast. It becomes more complicated to perform satisfying results to users to the search engines. In order to reduce the complexity of ranking the searching results based on the user's quires, web-page summarization can improve the accuracy of the search engine to rank the searching results. Generally speaking, it is an approach to summarize the vital information of the target website. It is an efficient way to reduce the dimension of the feature of context, so the ranking system doesn't need to read the entire information of the website.

Since the webpage is not constructed of pure text, the summarization approach should be able to handle its structure. Additionally, the webpage usually contains noise information, which can affect the result a lot. To solve web page summarization issues, four approaches were proposed in 2004: Adaption of Luhn's summarization method, Latent Semantic Analysis (LSA), Content Body Identification by Page Layout Analysis, and Supervised Summarization [2].

The adaption of Luhn's summarization method is an extraction-based method. It will create a significant word pool first, which calculates the significant factor to construct a summary of the target webpage. LSA is a singular value decomposition (SVD) based summarization method. It aims to find the sentences on the webpage by using the SVD technique, the sentences that have the highest importance value on the webpage, and then using them to form a summary. Content Body Identification by Page Layout analysis is a solution to summarize the webpage's important information without affecting the noises (e.g., navigation bar, button, footer). It will classify different web page objects and only focus on the Content Body (CB) information to summarize the web page. The above three algorithms are unsupervised summarization methods. The last method is supervised summarization. In general, it is implemented by using the training data as well as their labels to train a machine learning model. During the training, the model learns what sentences are essential for generating a summary of the web page's related topic. These are the brief introduction of that four methods, and their detailed implementation methods can be found in [2].

3.3. Social Media Summarization

Social media start being popular when Web 2.0 comes, such as Facebook and Twitter. In web 2.0, it emphasizes user-generated content, usability, and interactions. Therefore, the information on the Internet is no longer restricted, that only the website developer can create the content and information. The users can share or create information on the Internet via a different social media application. The user of social media is rapidly increased. The companies have spent a lot of time investigating different analyzing approaches to improve the service and suggest the potential advertisement that users might be interested in.

Automatic text summarization is a widely used approach to social media for analyzing the users' habits and desires—for example, opinion mining and sentiment-based summaries [1]. Moreover, different social media provide a different style of information created by users. Some automatic text summarization is specially developed to specific social media, such as Twitter Summarization [9] which generates the summaries of the tweets on Twitter.

4. Traditional Approaches

Automatic text summarization has been developed for over half a century. During these years, many researchers have proposed different implementation methods. Before machine learning and deep learning are widely used in text summarization tasks, the traditional approaches are mainly used to handle the summarization tasks. Most of them are developed based on statistics and probability. In the traditional text summarization task, the sentences in documents are scored, and the highest score sentences are selected to be composed as the summary. Therefore, the features scoring can bring a lot of effects to the summarization results. In this section, the traditional approaches, as well as the scoring methods, will be introduced. They are Word Frequency and Bayesian Model.

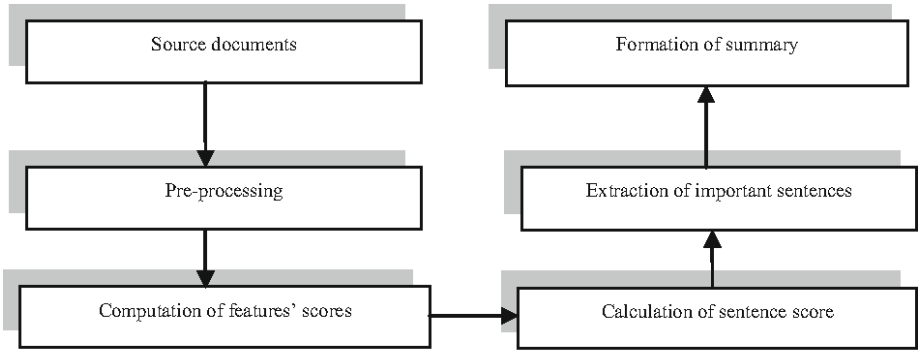

Figure 1 Block diagram of automatic extractive text summarization system by using statistical techniques [1]

4.1. Word Frequency

To determine the sentence's importance in the contents, each sentence should be evaluated whether its meaning represents the topic or abstract of the target documents or not. In traditional summarization, word frequency is an ideal feature for finding the important sentences in the documents, such as Term Frequency-Inverse Document Frequency (TF-IDF) and word probability [10].

The word probability is applied to evaluate the importance of sentences. SUMBASIC is an approach that calculates the importance of sentences in the documents based on word probability. This method aims to find the highest probability word in the best scoring sentence and then select another best sentence that covers this word to concatenate with the previous sentence. Because the highest probability word in the best scoring sentence represents that the word represents the most important topic in the document., it is a looping, and it will stop when the desired summary length is achieved.

However, the SUMBASIC replies on the stop word removal (i.e., removing too common words in the content such as "the", "a"). TF-IDF is the approach that provides outstanding performance in stop word removal. It consists of the term frequency and the inverse document frequency to evaluate the importance of the target word, apart from removing stop words for SUMBASIC. TF-IDF is also applied in different summarization methods, such as Centroid summarization.

4.2. Bayesian Model

Bayesian models [10] are similar to the word probability method, but the consisting weight of the model is different. The Bayesian model consists of a distribution of general English, a distribution of the entire cluster to be summarized, and distribution for each document in that cluster. Each of these three distributions consists of the words and their probabilities. Although Bayesian models are widely applied in topic model representation, they can also be applied in sentence scoring.

The additional requirement of applying the Bayesian model in sentence scoring is a table with word probabilities. By calculating the divergence of the word probabilities rather than the sum, multiply or average the word probabilities, the Bayesian model can use that divergence to score each sentence in the documents. Then, it generates a summary by selecting the best score of sentences and concatenating them.

5. Topic-Based Approaches

A topic can present the subject of single or multiple documents. The automatic text summarization can be developed based on the topic of documents. Actually, some traditional summarization approaches can also be applied to topic-based summarization, such as Bayesian Model and Word Frequency. However, this section only discusses two topic-based approaches, and one of these has been introduced in section 3.2 as an application case. They are Topic Signatures and Latent semantic analysis (LSA) [10].

5.1. Topic Signatures

Before the researchers propose the topic signatures approach, its predecessor was invented by the computer scientist H. P. Luhn who is a pioneer of automatic text summarization [5]. He proposed that the summaries of documents can be generated using the frequency threshold to identify descriptive words in that document. It can be seen as the generation of the document's topic. His approach removes the words that occur most frequently (e.g., Determiners, prepositions, or domain-specific words) and the least frequently based on counting the frequency of the words in the target document.

Based on Luhn's approaches, some researchers have invented a statical version [10]. The approach uses the log-likelihood ratio test to identify the highly descriptive word, and it is also known as "topic signatures". The words in the topic signature can be determined by the chi-square distribution (χ^2). However, the computation of the topic signature requires the additional collection of documents. The words in the topic signature should frequently occur in the input document but infrequently occur in the other collection of documents. Compared to the traditional approach setting the threshold with an arbitrary value proposed by Luhn, the statical version provides an approach to set the threshold rather than the arbitrary threshold. It divides all words in the input into either descriptive or not by applying the log-likelihood ratio test.

The sentence's importance is the consideration factors to evaluate whether that sentence is selected to be a part of the summary. There are two methods to calculate the importance of a sentence based on the topic representation: the computation relying on the number of topic signatures or relying on the proportion of topic signatures in the sentence. The former is always used in longer sentences scoring because they contain more words. The latter evaluates the importance depending on the density of topic words.

5.2. LSA

Latent semantic analysis (LSA) is an unsupervised technique to derive an implicit representation of the text semantics based on observed co-occurrence words [10]. As mentioned in section 3.2, it has been applied in the web-page summarization.

LSA is an approach based on Singular Value Decomposition (SVD), a decomposition method in linear algebra. The LSA is focusing on the topic-based summarization task. It can identify the important topics in documents without using lexical resources, so it is also an unsupervised approach. The words of the input documents and sentences of the input documents are used to construct a matrix. SVD is capable of reducing the dimensionality of the features. In LSA, removing the noise of documents (i.e., Low weight topics) relies on the dimension reduction of SVD. Additionally, the SVD computation can indicate to what extent the sentence conveys the topic by combining the topic weights and the sentence representation.

By using SVD, the important information of documents is distilled. After that, only a certain number of topics will be retained, and the number is the same as the number of sentences that the summary has. The sentences with the highest weight for each retained topic are selected to be a part of the summary.

6. Graph-Based Approaches

Graph-based summarization is constructing a graph by connecting the nodes and edges according to the relevant extent of the text elements in documents and then select the important sentences to generate the summary [1]. The nodes in the graph represent the sentences in documents, and the edges between nodes represent the weight of how similar between the sentences. The cosine similarity is a method in natural language processing that is used to compute the similarity of words by using the TF-IDF weights of the words. It is also a graph-based method.

Besides assigning weights to the edges in a graph, the graph can be constructed by setting a threshold. Only the sentences' similarity exceeds this predefined threshold which can be connected. In the graph, the sentences that meaning can represent the other sentences usually are the center of the graph and recommended to be a part of the summary.

Apart from using weighting to evaluate if a sentence can be selected in summary, the probability also can be used in graph-based summarization. The weight of edges can be normalized to form a probability distribution. The sum of the weight of the outgoing edges from a given node is one. The graph becomes a Markov Chain that each node in the graph represents a state, and the edges represent the probability that one state goes to the next state. By computing a stationary distribution in the graph and then using the distribution to find the probability from a given node which is a sentence. The high probability sentences can be selected to generate the summary.

There are some strengths that graph-based summarization can provide [10]. First, it is robust to the language issues in text summarization because it does not require language-specific linguistic processing. So that it can be applied to other languages. Also, it can handle the single document and multiple documents both well, which is a common issue in text summarization and mentioned in section 2.1. Moreover, it provides better performance than TF-IDF-based cosine similarity by appending the syntactic and semantic role information in graph construction.

7. Machine Learning Approaches

The traditional approaches or topic-based approaches rely on the statistics method or counting the word frequency to evaluate each sentence's importance in the given documents. However, their provided sentence importance is evaluated by the superficial features. For example, the approach of SUMBASIC mentioned in section 4.1 scoring the sentence by using word probability. In fact, many features help look for the representative sentence in input to form a summary, such as the position of the sentence in the document, position in the paragraph, sentence length, the similarity of the sentence with the document title, or weights of the words in a sentence, etc. [10] Machine learning are capable of achieving the classification by multiple features. It consists of two categories of models: generative (e.g., Naïve Bayes and Hidden Markov Models) and discriminative (e.g., Support Vector Machine and Logistic Regression). This section introduces three machine learning methods in text summarization: Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning.

7.1. Supervised Learning

In supervised learning, the classifier is trained by the labeled training data. The summarization task in supervised learning is classifying whether the sentence is either a summary or not. The training data for training the summarization classifier consists of the sentences that should be a part of the summary labeled by humans and the relevant features of the sentences. The classification result is used to determine whether it is included in the summary. It could be computing the likelihood of a sentence belonging to the summary class or the score given by the classifier to indicate that how possible the sentence is a part of the summary—then choosing the best score or likelihood sentences to form a summary.

However, there are some limitations in text summarization by using supervised learning. As mentioned, supervised learning requires the labeled sentences to train the classifiers. It is a complicated task because collecting a lot of labeled data is not easy. Especially in text summarization, it requires collecting the sentences, which is worth being a part of the summary. The simplest way is to ask professional people to select the summary-related sentences manually, but it is time-consuming, and the result is easily affected by different people’s subjective judgment [10]. Besides, abstractive summaries are more popular than

extractive summaries. So that fewer people are interested in solving the training data problem in extractive summary tasks, the differences between abstractive and extractive summaries can read the section 2.2.

Therefore, researchers have proposed three methods [10] to collect the labeled training data. The first method is achieving an automatic alignment of the human abstracts and the input documents. It can isolate the summary sentences and non-summary sentences from the documents to provide the labeled data for training. The second method is using the information in the manually created summaries to generate the useful training data. The last method calculates the similarity between human abstracts and the input to find the sentences that are most similar to the summary, and it doesn't require full alignment.

7.2. Unsupervised Learning

On the contrary, unsupervised learning does not require the labeled data. It is robust to the first-time meet data. Clustering and Hidden Markov Model (HMM) are famous examples in unsupervised learning, and they both can handle different issues in automatic text summarization.

Clustering [10] is an unsupervised machine learning approach (e.g., K-means clustering) to group the approximate data to achieve the classification purpose. Also, it does not require the labeled data because it does not need to be trained. In automatic text summarization, the information frequently occurs in the input documents, which means that it deserved to be a part of the summary. Using clustering as the summarization, converting the content of documents to the features needs to be done first. The sentence similarity is mainly used to represent the features of the sentences in text summarization. Therefore, each sentence of documents is converted to similarity and uses the similarity to achieve the clustering. For the summarization, only select the cluster that contains the most sentences. Because the more sentence in a cluster means the cluster is more important and representative, the sentences in that cluster can form a summary to represent the documents.

Hidden Markov Model (HMM) [10] is another famous unsupervised learning model. It is widely used in natural language processing except in text summarization, such as part-of-speech tagging. It has the ability to capture what topics, as well as the flows of them, are discussed in the content of documents. Since some features of a story or content are presented in a few documents continuously, connecting them is a big help in summarization. In HMM, a summary sentence is selected depending on its probability, and only the higher is selected.

7.3. Semi-Supervised Learning

As mentioned, the labeled data is a bottleneck of supervised learning because it requires a large amount of labeled training data. For solving this problem in text summarization, some researchers proposed using semi-supervised learning rather than supervised learning to train the summarizer [10].

Semi-supervised learning is one of the machine learning methods. They are similar, but the key difference is that its learning producer consists of a small amount of labeled data and a large amount of unlabeled data. An approach mentioned in [10], requires two classifiers and one classifier to train the labeled data. After that, the trained classifier is used to predict the unlabeled data. Then selecting the most confident results are added into the labeled data and using the updated labeled data to train another classifier. This repeating training is stopped when the result is satisfied.

8. Deep Learning Approaches

In recent years, deep learning is proliferating and is widely used in different NLP tasks because of its outstanding performance. The deep learning models are various, such as recurrent neural network (RNN) and transformer learning based RNN, which both models are famous in NLP. In automatic text summarization, the deep learning approaches are among the popular topics in the relevant research. This section will introduce two deep learning-based text summarization approaches invented by the researchers: the Sequence-to-sequence RNNs and transformer-based approaches.

8.1. Sequence-to-sequence RNNs

Most of the summarization approaches mentioned in previous sections are used to generate extractive summaries. The critical differences between extractive and abstractive summary have been explained in section 2.2. The abstractive summary is generated by the machine or system, which consists of unseen words in the input documents. The traditional approaches or machine learning models might not provide outstanding performance in abstractive summarization. Therefore, researchers have referred to a famous Machine Translation approach to invent the abstractive summarization approach using sequence-to-sequence RNNs [11].

This approach consists of the attention mechanism, encoder-decoder, and Recurrent Neural Networks (RNNs). In addition, the researchers designed a few specific features for the model to solve the particular problems in the summarization. They will be briefly explained in the following.

RNNs is a type of neural network widely used in NLP tasks because it can extract the features sequentially. The text contents are presented sequentially, so using RNNs can bring better performance. In this approach, the encoder and decoder are constructed by the RNNs. Encoder-Decoder is a framework, and it is the basis of the sequence-to-sequence model. Since

the traditional RNNs cannot handle the task like translation and summarization, their input and output length are usually different. The Sequence-to-Sequence (seq2seq) model can solve this issue because it is constructed by the encoder-decoder framework, which converts the fixed-length input to the context vector and generates a fixed-length output. However, the weakness of Encoder-Decoder is the sentence length. If the sentence length is too long, the model will be unreliable. Therefore, research proposed combining the attention mechanism into the Encoder-Decoder. In general speaking, the attention mechanism can compute a context vector for each decoder. It is recommended to read the original paper [12] to know the detail of the design.

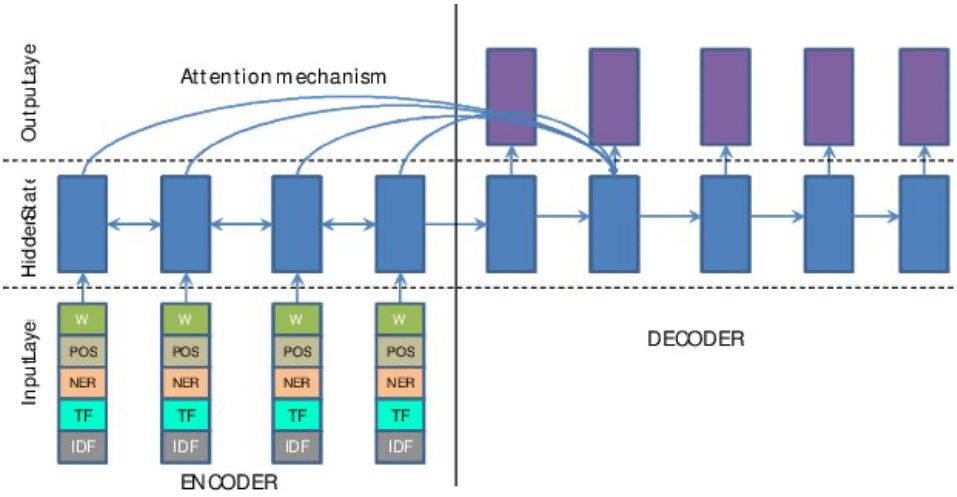

Figure 2.1 Capturing keywords using Feature-rich Encoder [11]

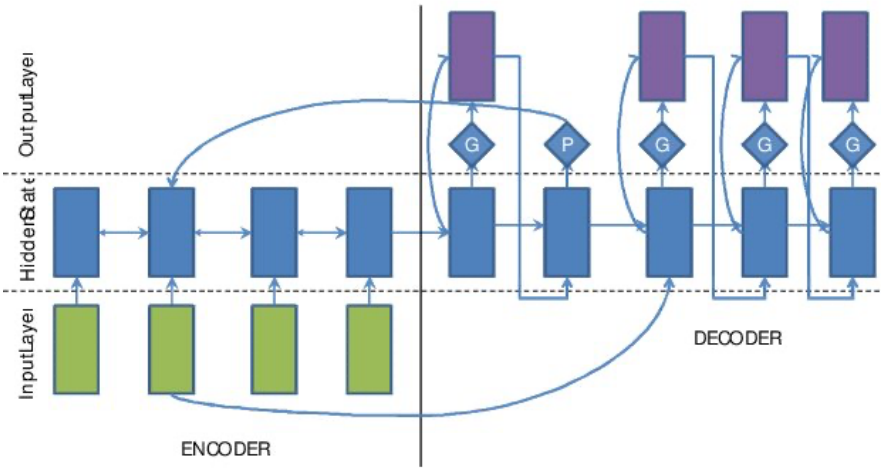

Figure 2.2 Modeling Rare/ Unseen Words using Switching Generator-Pointer [11]

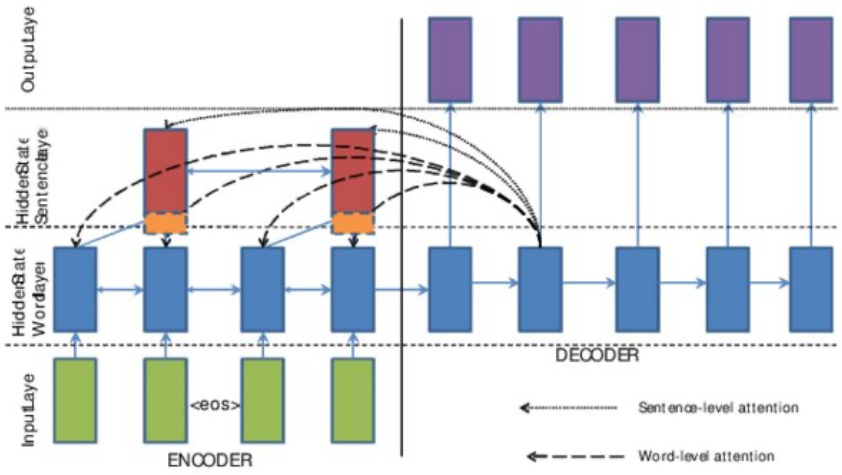

Figure 2.3 Capturing Hierarchical Document Structure with Hierarchical Attention [11]

The seq2seq RNNs summarization provides three strengths: capturing keywords, handling unseen/rare words, and capturing hierarchical documents. Figure 2.1 shows the model using a Feature-rich encoder to capture the keywords. For the abstractive summary, the key concepts or stories of the documents are essential. So that the feature-rich encoder requires the input that is not only the words of sentences, different features are also included in the input, such as part-of-speech tags, named-entity, TF-IDF, etc. Another issue is common in different NLP tasks, that is, handling the unseen/ rare words. Figure 2.2 shows the decoder using the switching generator-pointer to decide how to handle these words. The switcher is opened according to the probability. If it is opened, the decoder gives the word by the generator. Otherwise, the decoder generates a pointer to one of the word positions in the input documents by evaluating the probability and then copy the word as the output. As mentioned, the purpose of summarization is to select the key sentences from the input documents. Figure 2.3 shows the third strength of the seq2seq RNN model. It applies the hierarchical attention mechanism to capture the essential features at the word level and sentence level. Therefore, the meaning of the abstractive output summary can be highly approximate to the input documents.

8.2. Transfer Learning

The deep learning model performs well in automatic text summarization. Still, it is time-consuming to develop the entire model and generally requires a lot of training time to learn from the training data. It is because the deep learning model typically consists of many parameters and is a complex structure. Thereby, some researchers have proposed using transfer learning to improve the performance of the traditional deep learning model.

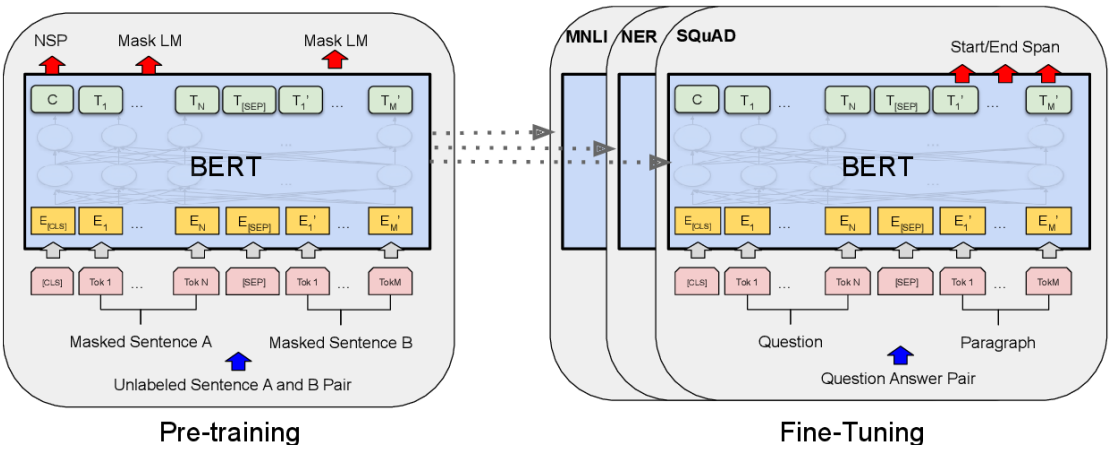

Figure 3.1 Pre-training and Fine-tuning in BERT

Transfer learning is one of the machine learning approaches. It aims to train a model for a new task by applying a pre-trained model, as long as the pre-trained and new tasks are approximate. Although it was developed for machine learning, it has been applied to different categories of deep learning, such as Computer Vision and NLP. Google researchers have proposed a famous model implemented by transfer learning that produces remarkable results in different NLP tasks called Bidirectional Encoder Representations from Transformers (BERT) [13]. It is a pre-trained encoder for different NLP tasks, such as question answering, sentence classification, sentiments analysis, etc. BERT is constructed by model training using many parameters (e.g., BERT-BASE: 110M parameters, BERT-LARGE: 340M parameters). The entire task of using BERT is shown in Figure 3.1. It consists of pre-training and fine-tuning. Fine-tuning aims to develop a final model based on the pre-trained encoder, and the final model can achieve any different task, as long as it also is a language-related task. For the fine-tuning, the model needs to be trained by the task-related dataset. For example, Figure 3.1 is fine-tuning the model for achieving a question answering system, so the dataset is related to question answering and reading comprehension.

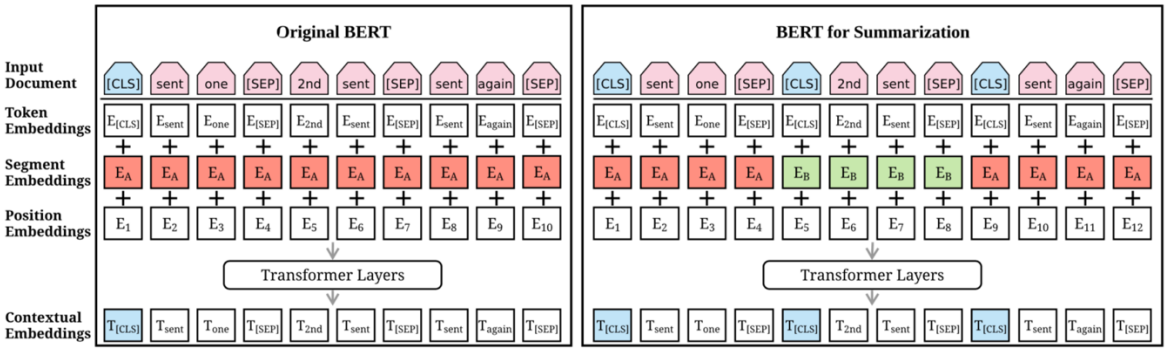

Figure 3.2 BERT and BERT for Summarization models [14]

The BERT is powerful in various language process tasks, so some researchers propose using BERT to achieve automatic text summarization [14]. This approach constructs a summarization model based on BERT called BERTSUM, and it is capable of achieving both extractive and abstractive summarization tasks. Figure 3.2 shows the comparison between the original BERT model and the BERTSUM model. Since the original BERT is not designed to process the multiple-sentences input, its output vector is tokens instead of sentences. In original BERT, it only inserts once [CLS] token in the start position of each input. In the case of extractive summarization, each input consists of multiple sentences. Therefore, researchers proposed that [CLS] tokens should be inserted in the start position of each sentence in the input to let the BERT model collect the features for the sentence preceding the [CLS] token. The difference is shown in Figure 3.2. In BERTSUM, the "sent one" and "sent again" are the same in Segment Embeddings layers, and "2nd sent" is different from them. In Original BERT, all input words are the same in Segment Embeddings layers. Because the input in Original BERT is defined as one class, the input in BERTSUM is defined as a few classes according to the sentences in the input.

For the extractive summarization task, BERTSUM will define it as a classification task. It is using probability to determine whether the sentence is a part of the summary of input documents. It is more superficial than abstractive summarization using BERTSUM because the binary classification task is not difficult for the BERT model. In contrast, abstractive summarization is a type of text generation task, and it is challenging to the BERT model. Therefore, the researchers raise an approach using the BERTSUM as the pre-trained encoder and then apply a six-layers transformer as the fine-tuning model to achieve abstractive summarization. This approach can highly improve the performance because the researchers propose that using two-stage fine-tuning. It means that the pre-trained encoder and the abstractive summarization model will both be fine-tuned. Fine-tuning the pre-trained encoder can let it classify what sentences in the source documents can be used to construct an extractive summary. The second fine-tuning for the abstractive summarization task is to achieve the model that can generate an abstractive summary according to the input documents and the sentences selected by the pre-trained encoder.

9. Conclusion

Automatic text summarization has already been developed for over half a century. From the beginning of using pure mathematic and statistical to generate a summary to nowadays applying deep learning and machine learning, many researchers have participated in this field of research. This survey paper introduced the background about automatic text summarization and the essential factors needed in the summarization tasks. These factors can bring the different extent of results and complexity to the summarization task. Moreover, various using text summarization applications are also introduced. It reflects that automatic text summarization techniques are widely applied in different areas in the real world. Further, different types of summarization approaches, as well as their methods, are briefly explained in this paper. In the future, we will keep looking for state-of-the-art approaches and relevant researches.

Reference:

[1] Gambhir, M., Gupta, V., "Recent automatic text summarization techniques: a survey," *Artif Intell Rev 4,* pp. 1-66, 2017.

[2] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, and Wei-Ying Ma., "Web-page classification through summarization," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04)*, Association for Computing Machinery, New York, NY, USA, 242–249., 2004.

[3] Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen, "Summarizing email threads," in *Proceedings of HLT-NAACL 2004: Short Papers (HLT-NAACL-Short '04)*, Association for Computational Linguistics, USA, 105–108, 2004.

[4] Chua, F. C, S. Asur. , "Automatic Summarization of Events from Social Media," in *ICWSM*, 2013.

[5] H. P. Luhn, "The Automatic Creation of Literature Abstracts," in *IBM Journal of Research and Development, vol. 2, no. 2*, Apr. 1958, pp. 159-165.

[6] Hahn, U., Mani, I., "The Challenges of Automatic Summarization," *Computer,* vol. 33, pp. 29-36, 2000.

[7] Goldstein, Jade and Mittal, Vibhu and Carbonell, Jaime and Kantrowitz, Mark, "Multi-Document Summarization by Sentence Extraction," in *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization - Volume 4 (NAACL-ANLP-AutoSum '00)*, Association for Computational Linguistics, USA, 40–48, 2000.

[8] N. W. E. contributors, Bibliographic details for Abstract (summary), New World Encyclopedia,, 8 April 2021 23:38 UTC.

[9] Ruifang He and Yang Liu and Guangchuan Yu and Jiliang Tang and Q. Hu and J. Dang, "Twitter summarization with social-temporal context," *World Wide Web,* vol. 20, pp. 267-290, 2016.

[10] A. Nenkova, K. McKeown, "A Survey of Text Summarization Techniques," in *Mining Text Data*, 2012.

[11] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, Bing Xiang , "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," in *Proceedings of The 20th {SIGNLL} Conference on Computational Natural Language Learning*, Berlin, Germany, Association for Computational Linguistics, 2016, pp. 280--290.

[12] Dzmitry Bahdanau and Kyunghyun Cho and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR,* vol. abs/1409.0473, 2015.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[14] Liu, Yang and Lapata, Mirella, "Text Summarization with Pretrained Encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Association for Computational Linguistics, 2019, pp. 3730--3740.