Text Summarization

Mingzhi Hu Liuyi Cui Noah Dcruz Haoen Huang
Department of Mathematics, Northeastern University

April 25, 2024

Abstract

Text summarization is a class of sequence-to-sequence problems that is useful in our daily lives. The goal is to learn the mapping between input sequences and output sequences which could brief and cohesive summarization. We introduce two kinds of methods to do the job, the sentence ranking method and the deep learning method. We also apply them to appropriate datasets to see how methods work and perform.

1 Introduction

1.1 Definition

Text summarization is a process in natural language processing (NLP) where a computer program reduces a text document to its most essential contents or ideas, providing a concise version of the original document's main points. In the context of academic or professional papers, text summarization can be especially valuable for readers who need to quickly understand a document's content without reading the full text.

1.2 Applications

Text summariazation is widely used in different scenario.

1.2.1 News Aggregation

Automated text summarization tools are crucial for news aggregation platforms that curate content from various sources. These tools generate concise summaries of news articles, allowing users to quickly scan through headlines and key points without delving into full articles. This helps users stay informed with less time investment and enables them to cover more topics or stories. It is particularly beneficial during breaking news situations where information is rapidly updated.

An example of this application is that a platform like Google News uses summarization to present snippets of news articles from different publications, providing a digest of daily news.

1.2.2 Academic Research

Researchers use text summarization to manage the vast amount of literature available, particularly when conducting reviews of existing literature or when trying to identify research gaps and trends. Summarization reduces the time and effort needed to access relevant information across numerous documents and helps in synthesizing broad themes and findings from multiple studies.

1.2.3 Business Intelligence

Summarization algorithms can quickly sift through business reports, executive summaries, market analyses, and internal documents to extract crucial insights and data points. This enables decision-makers to quickly grasp essential information and trends without reading through extensive documents, which is key in dynamic and fast-paced business environments.

An example for this is a multinational corporation might use summarization tools to consolidate monthly performance reports from different regions into a single, concise executive overview.

1.2.4 Legal Document Review

In the legal field, summarization helps in processing large volumes of text found in case documents, legislation, contracts, and precedents. Lawyers can quickly find relevant information, reducing the hours spent on manual document review. This efficiency is critical in preparation for cases or during discovery processes where time and accuracy are paramount.

1.2.5 Customer Feedback Analysis

Companies utilize text summarization to analyze customer feedback collected through reviews, surveys, and social media channels. Summarization extracts significant sentiments and common issues raised by customers, facilitating quicker response to feedback and aiding in the prioritization of customer service and product development initiatives.

2 Related work

In the realm of text summarization, various methodologies have been developed to enhance the efficacy and accuracy of summaries generated from extensive texts. One prevalent approach is the extractive summarization method, which involves selecting key sentences or phrases from the original text without altering their content. This method primarily relies on algorithms such as TextRank or LexRank,

which utilize graph-based models to determine the relevance and importance of each sentence based on their structural and contextual relationships within the text.

Another significant approach is abstractive summarization, where the goal is to generate new sentences that encapsulate the core information of the source text, often using advanced neural network models like sequence-to-sequence (Seq2Seq) architectures. These models, particularly when enhanced with attention mechanisms, have shown remarkable ability to understand and reformulate text, producing more coherent and concise summaries.

Hybrid models combine the strengths of both extractive and abstractive approaches, employing techniques to first extract relevant information before using deep learning strategies to paraphrase and condense these extracts into a fluent summary. Such models aim to leverage the precision of extractive methods and the natural language generation capabilities of abstractive systems.

Machine learning techniques, specifically supervised learning, have also been applied to summarization tasks. These methods involve training classifiers to identify sentences that should be included in a summary based on features like sentence length, term frequency-inverse document frequency (TF-IDF), and others. The effectiveness of these techniques largely depends on the quality and size of the annotated datasets used for training.

Lastly, semantic analysis methods such as Latent Semantic Analysis (LSA) are used to uncover the underlying thematic structures of texts. By identifying patterns and relationships among terms and sentences, LSA can effectively discern and compile the most representative sentences, providing a gist of the content that reflects the text's thematic substance.

Together, these methodologies reflect a diverse toolkit for researchers and practitioners in the field of natural language processing, each contributing unique strengths to the challenges of text summarization.

3 Formulation

Our goal is to get a model that can generate the summary of an article from a dataset with articles and highlights.

3.1 Evaluation with ROUGE-L

We use ROUGE, short for"Recall-Oriented Understudy for Gisting Evaluation" for our evaluation method. ROUGE is a set of metrics used to evaluate the quality of automatic summaries generated by text summarization systems compared to reference summaries. ROUGE-L is a variation of ROUGE, it specially accesses the overlap of Longest Common Subsequences (LCS). The LCS is the longest sequence of words that appears in both the system-generated summary and the reference summary.

ROUGE-L computes precision, recall, and F1-score based on the LCS as follows:

$$R_{lcs} = \frac{\sum_{u}^{i=1} LCS(C, r_i)}{m} \tag{1}$$

$$P_{lcs} = \frac{\sum_{u}^{i=1} LCS(C, r_i)}{n} \tag{2}$$

$$F_{lcs} = \frac{(1+\beta^2) \cdot P_{lcs} \cdot R_{lcs}}{\beta^2 \cdot P_{lcs} + R_{lcs}} \tag{3}$$

In equation (1), $LCS(C, r_i)$ represents the LCS score between system-generated summary C and the i-th sentence in the reference summary r. u is the total number of sentences in r. m is the total number of words in r. Recall measures the proportion of LCS words in the system-generated summary compared to the total number of words in the reference summary. Higher recall means more of the reference content is covered.

In equation (2), n is the total number of words in the system-generated summary. Precision here measures the proportion of the system-generated summary that corresponds accurately to any part of the reference summary. It is normalized by the total number of words in the system-generated summary. High precision means less irrelevant information in the system-generated summary.

In equation (3), β represents a weighting factor that adjusts the importance of recall relative to precision. In summarization tasks, recall often is more heavily weighted (? 8), emphasizing the importance of not missing information from the reference. The F1-score is the harmonic mean of precision and recall, providing a single metric to gauge the quality of the summary.[7]

In our method, we calculate these three scores through ROUGE-L to evaluate the model performance, but we focus on comparing the F1-score. Due to the higher β, the score is biased towards recall, ensuring that the summary captures as much of the important information as possible even if it includes some irrelevant details. In other words, a higher F1-score indicates that a system-generated summary is more similar to a human-written summary in terms of its content and quality.

3.2 Dataset description

We use the datasets from Hugging Face called the "CNN dailymail". It contains over 300,000 news articles, each paired with multi-sentence summaries, giving us a rich training ground for our model. The dataset spans various topics, providing our model with a broad understanding of language and context, crucial for producing relevant summaries. We split the data with 280000 pairs for training,10000 for validation, and 10000 for testing.

4 Models

4.1 Sentence ranking method

4.1.1 General Idea

The sentence ranking methodology is predicated on a straightforward concept: to distill the essence of an article by extracting sentences directly from the original text. Consequently, we assess the importance of each sentence and select those with the highest rankings, arranging them sequentially to produce the final summary. The ranking of importance is based on the premise that the more frequently a word appears, the more significant it is deemed to be.

4.1.2 Detailed method

To ascertain the significance of each sentence, the initial step involves decomposing each sentence into its constituent words. A straightforward approach to accomplish this is to segment the text at each occurrence of space, a task performed using the Python package spaCy. Subsequently, it is imperative to eliminate words that do not contribute meaningfully to the text. This category includes punctuation marks such as commas and periods, as well as the '\n' character, commonly found in various file formats. Moreover, certain auxiliary verbs such as 'are' and 'is', which do not add substantive value to the sentence's meaning, must also be removed. The string package contains a collection of punctuation marks, while the spaCy package provides a list of stop-words that aid in excluding these non-essential words. Following this filtration, the remaining words typically consist of nouns, verbs, adjectives, and adverbs, which are precisely the types of words sought.

To establish a ranking based on these meaningful words, the frequency of each word is calculated, and then each sentence is ranked based on the frequency of each meaningful word it contains. With these rankings established, the final task is to select the sentences with the highest rankings and compile them into a cohesive summary.

4.1.3 Limitations

This is our baseline method comparing the following t5 model one. The baseline method, centered on sentence ranking, exhibits several constraints. Notably, it operates under the assumption that summaries are often direct extractions from the original text—an assumption that does not hold in standard text summarization practices where paraphrasing is commonplace. During the sentence ranking phase, the algorithm quantifies the significance of words based on their frequency within the text. This frequency-based representation, however, does not adequately account for semantic variations introduced through rephrasing, which is a prevalent stylistic element in article writing. As a consequence, the algorithm may erroneously devalue semantically crucial words simply because they are not frequently repeated.

Moreover, this approach faces challenges in dealing with information redundancy. In instances where pivotal information is reiterated within the source text, the model's selection criteria might lead to the redundant extraction of sentences. Such repetition dilutes the brevity and efficacy of the resulting summary, indicating a need for an improved mechanism that can discern and condense repeated core messages without sacrificing the essence of the text. The current baseline method's limitations underscore the necessity for a more sophisticated model that can navigate the nuances of linguistic expression and content repetition with greater finesse.

4.2 T5 model

4.2.1 Intro to Transformers

Transformers are a type of deep-learning model that caused a revolution in NLP. Introduced by Vaswani et al. in their 2017 paper "Attention is All You Need". The advantage of transformers is not using the recurrent units that make transformers have less training time than recurrent neural architectures, like CNN, and LSTM, transformers rely entirely on self-attention mechanisms to weigh the importance of different words in a sequence when processing information.[3]

Similar to early seq2seq models, the transformer model utilizes an encoder-decoder architecture. The encoder consists of encoder layers that iteratively process the input, while the decoder comprises decoder layers that perform the same operations on the encoder's output. Each encoder layer's function is to determine which parts of the input data are related to each other. It passes its encoding as input to the next encoder layer. The function of each decoder layer is the opposite, reading the encoded information and using the integrated context information to generate the output sequence.[4] In the transformer model, each encoder and decoder layer uses an attention mechanism.

The innovation of transformers is the self-attention mechanism, this mechanism allows the model to extract state information from any previous point in the sequence. The attention layer can access all previous states and weight them according to the learned relevance metric, providing information about tokens that are far apart. In the Transformer architecture, self-attention is not computed once, but rather multiple times in parallel and independently.[5]

4.2.2 Inrto to T5 Model

The T5 model, short for "Text-To-Text Transfer Transformer," is a powerful neural network architecture based on the Transformer model. As its name, T5 frames all NLP tasks as a text-to-text transformation problem. This means that regardless of the task, both the input and output are treated as sequences of text. This unified approach allows T5 to handle various tasks, including classification, regression, language modeling, translation, summarization, and more, by simply adjusting the input and output formats.[6]

The one of key features of T5 model is pre-training and fine-tuning, T5 is pre-trained on large text corpora using unsupervised learning objectives. After pre-training, it can be

steps	training loss	validation loss
512	1.7432	1.724787
1024	1.7311	1.722753
1536	1.7333	1.709278
2048	1.8558	1.680838
2560	1.8427	1.661814
3072	1.827	1.664096
3584	1.8285	1.653124
4096	1.8193	1.656115
4608	1.8059	1.660652
5120	1.778	1.644493
5632	1.7708	1.640512
6144	1.7671	1.636624
6656	1.7722	1.638855
7168	1.7666	1.632809
7680	1.7655	1.636061
8192	1.7706	1.62685
8704	1.7627	1.629631
9216	1.7528	1.626875
9728	1.7405	1.627758
10240	1.7336	1.624775
10752	1.7378	1.626855

Table 1: Logs of training with learning rate 3e-4

used for our purpose easily with high performance.

4.2.3 Training Model

In the beginning, we use the training dataset and test dataset during the training step and use validation data for model comparison.

For the model, we use the transformers package, and the 'google-t5-small' pre-trained model as start point. We choose 3e-4 as the learning rate, which is recommended in the huggingface T5 model introduction page [5]. We set the batch size to be 8, and after 3 epochs, the training loss and validation loss are shown in Table 1.

The validation loss is lower than the training loss along the whole period of training, which is abnormal. To eliminate the influence of the dataset's variability between training data and test data, we retrain the model on the new datasets, which are generated by randomly splitting the original training data into two new data sets. The training loss and validation loss are shown in Table2. After 2560 steps, we can see the validation loss still is less than the training loss, which indicates that the variability of the training and validation dataset is not the priority reason.

Another reason that could lead to this phenomenon is the presence of dropout layers. When we are training the model, dropout layers present in the model, which deactivate some weights for the forward pass. It helps avoid overfitting but reduces the performance of the model. However during evaluation steps, dropout layers will not work, the complete network may perform better on the same model.

After figuring out that there is no huge variability between the training and test datasets, we go back to the previous datasets and checkpoints of the model and retrain it. To keep the loss decreasing, we choose 1e-5, as the learning rate. After 512 steps training, the loss remains stable and then we change a smaller learning rate 1e-6. When the loss stays stable again, we finally stop the long-time training and save it as our fine-tuned model for summarization. Logs of training are shown in Table 3.

steps	training loss	validation loss
512	1.9641	1.699828
1024	1.88	1.688986
1536	1.864	1.678948
2048	1.857	1.67848
2560	1.8545	1.675138

Table 2: Logs of training on modified datasets

steps	training loss	validation loss
512	1.7080	1.62304
1024	1.6963	1.619595
1536	1.6966	1.620052

Table 3: Logs of retraining with learning rate 1e-5 and 1e-6

5 Results

The results include the performance of sentence ranking method and the T5 model.

For the sentence ranking method, we use the spaCy package and 'en_core_web_sm' model to do the job, all parameters are fixed. For the T5 model, we use our fine-tuned model and original pre-trained model to do the comparison work. And we compute the ROUGE-L F1-scores given by the rouge package.

The validation datasets contain 11490 news articles, which takes a pretty long time to complete evaluation work. We randomly select 10% of them as validation datasets.

Results are shown in Table 4, which contains the average Recall, Precision and F1-scores between predictions and labels of two different models. Besides it, we also compute the average ROUGE-L of original pre-trained model to see the changes after training.

Notice that the F1-score of the T5 model looks still not good, however during training the model and doing the evaluation on test datasets, the F1-score is over 0.69. The phenomenon may also be caused by the presentence of dropout layers. Anyway, the result of the T5 model still be better than the sentence ranking method, and the outputs are more

Methods	Recall	Precision	F1-score
Sentence ranking	0.2276	0.3475	0.2657
pre-trained T5	0.3485	0.2438	0.2786
fine-tuned T5	0.4069	0.3151	0.3470

Table 4: Rouge-l results between two methods

abstractive and readable. Besides, the performance is also better after training, which indicates our tuning works for the T5 model.

6 Discussions and future works

This paper presents a dual-method approach to text summarization, employing both sentence ranking and transformer-based deep learning techniques. The comparative analysis revealed that while the sentence ranking method excels in rapidly identifying key phrases and maintaining the factual accuracy of the original text, the transformer model demonstrates superior capabilities in understanding context and generating cohesive, abstractive summaries. Notably, the transformer model, with its inherent complexity and deeper contextual comprehension, offers summaries that are not only concise but also exhibit a higher degree of readability and fluidity, closely resembling human-generated abstracts. However, this sophistication comes at the cost of increased computational resources and processing time. Our findings suggest a complementary relationship between the two methods, where the strengths of one address the limitations of the other. The integration of these approaches within a singular framework could potentially leverage the speed and accuracy of sentence ranking with the nuanced language modeling of transformer networks, leading to an optimal summarization tool.

Looking ahead, there is ample scope for expanding upon the current text summarization models. Future research could explore the integration of additional methodologies such as latent semantic analysis, reinforcement learning, or hybrid models that blend extractive and abstractive techniques to enhance summary quality further. Moreover, the application of newer, more compact transformer architectures, such as those utilizing knowledge distillation, presents an exciting avenue to maintain high performance while reducing the computational load. Experimentation with domain-specific models, especially in fields with specialized lexicons such as legal or medical, could also yield improvements in summary precision. In the broader perspective, investigating unsupervised learning approaches could eliminate the dependency on large annotated datasets, making the model more adaptable and less resource-intensive. Finally, cross-lingual summarization remains a largely untapped frontier that, if addressed, could significantly broaden the applicability and accessibility of text summarization technology across different languages and cultures.

References

[1] Y. Kumar, K. Kaur, and S. Kaur. Study of automatic text summarization approaches in different languages. *Artificial Intelligence Review, 54, 5897–5929, 2021.*

[2] Mahak Gambhir, Vishal Gupta. Recent automatic text summarization techniques: a survey

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems.* 2017.

[4] Sequence Modeling with Neural Networks (Part 2): Attention Models *Indico.* 2016.

[5] Prateek Joshi. How do Transformers Work in NLP? A Guide to the Latest State-of-the-Art Models. *Analytics Vidhya.* 2023.

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv, cs.LG, 1910.10683.* 2023.

[7] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74-81, Barcelona, Spain. Association for Computational Linguistics.

[8] Google–t5-small: https://huggingface.co/docs/transformers /v4.14.1/model_doc/t5