Survey, Implement and Evaluate Automatic Text Summarization Techniques

Dhyey Pandya
Department of Computer Science (MCS)
University of Illinois Urbana Champaign
dhyeyhp2@illinois.edu

Introduction

Automatic text summarization is the process of generating a summary of a large text that conveys the main points of the original content. There are mainly two approaches to text summarization: extractive and abstractive. In extractive text summarization, the sentences of the text are scored and the top k sentences are selected for the summary. In abstractive text summarization, new sentences are generated based on the content of the text, presenting the main points of the original text in the summary using different words than those found in the original text.

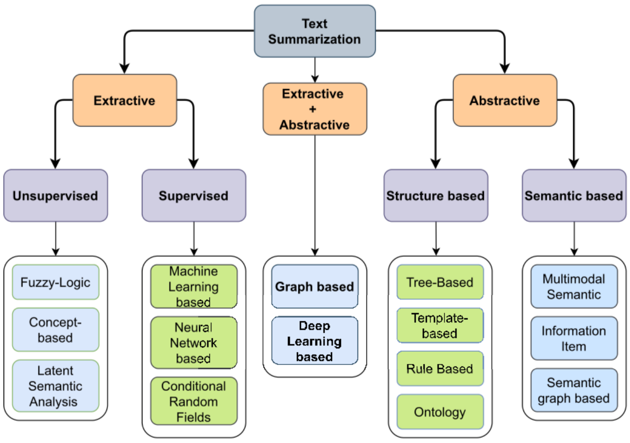

Fig 1. Text Summarization Techniques

There are numerous techniques, Fig 1 [1], that can help achieve abstractive or extractive summary of text, but the following deep learning-based techniques have gained popularity in recent times. Moreover, the abstractive methods are generally associated with better performance as compared to extractive ones. Three of such abstractive text summarization techniques are:

1) PEGASUS: PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization)[2] is

a transformer based pre-trained neural network which combines both encoder and decoder for text generation. PEGASUS-large provides the best results for the XSum dataset.

2) GSum: GSum [3] helps perform automatic text summarization using guidance signals. These signals can be keywords or phrases entered manually or selected via an algorithm or even summaries obtained via the extractive method. GSum, in particular, provides the best results for CNN / DailyMail and Reddit TIFU Long datasets.

3) BART (Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension) [4] is a neural network designed for text generation. It is based on a Transformer type architecture, comprising both a bidirectional encoder and an auto-regressive decoder. BART also seems to provide consistent and satisfactory results on news-based datasets.

Implementation

TextRank [5]: TextRank is a a graph-based ranking model for text processing which can be utilized in extractive text summarization task. This algorithm was inspired from the original PageRank algorithm for Web. After representing the whole text as a Graph where each sentence is a node, and the sentences are scored/ranked using TextRank algorithm. The top scored sentences can then be used in summarizing the whole text. TextRank is completely unsupervised, and unlike other supervised systems, it relies exclusively on information drawn from the text itself, which makes it easily portable to other text collections, domains, and languages.

Pre-Processing

Here, I tested the algorithm with and without pre-processing step. Pre-processing included stemming of word tokens, removed stop words, and filtered word tokens based on their part of speech(pos), which is to keep only Noun and Adjective phrases. Pre-processing was done for both, the model summaries and the references while computing the ROUGE score for model summaries. This was done to observe the impact of pre-processing in TextRank algorithm.

Results and Analysis

In my implementation of TextRank, DUC2004 dataset[1] is used which consists of 1000 news's articles in English and their reference summaries for evaluation, and ROUGE-1[6] evaluation metric is used for model performance comparison. While evaluation it was observed that TextRank performed better with pre-processing filters as compared to including all words (no filters). Moreover, we see TextRank performed better on DUC2004 dataset with pre-

[1] *The original paper used DUC2002 dataset, but since it required formal request to gain access of DUC2002 which could have taken unknown time to gain access to, So I went ahead with DUC2004 dataset available publicly and since it is also widely used for training and evaluating Text Summarization tasks.*

processing filters applied. Lastly, we see some difference in performance on DUC2002 dataset as compared to DUC2004, and one reason for this is that evaluation on DUC2004 was done based on only a subset of texts and their reference summaries and not the entire dataset.

System	Dataset	Rouge-1 score **with** filters (stemmed, no stop words, POS filtering)	Rouge-1 score **without** filters (no stemming, with stop words, and no POS filtering
Original TextRank	DUC2002	0.4299	0.4708
TextRank	**DUC2004**	**0.3982**	**0.3360**

Table 1 Comparison of ROUGE-1 Score on DUC2002 and DUC2004 Datasets

Conclusion

In this study, we briefly discussed text summarization and its approaches. We then implemented one technique for extractive text summarization called TextRank on a different dataset and analysed the results. We also developed a Chrome extension to demonstrate the practical application of TextRank for summarizing content in web pages such as news articles and Medium articles. Some observations on the summaries produced by the TextRank-based text summarization extension are as follows:

1. The summaries produced effectively highlighted the major points in the text and had the flexibility to be short or lengthy based on the user's requirements.
2. However, the summary sometimes included long sentences that made the summary lengthy and failed to cover all important points. To improve the quality of the summary, abstractive text summarization techniques could be used to shorten the long sentences and rephrase the summary in a more concise and meaningful way.
3. Extracting relevant content from web pages with varying structures was a challenge. This issue could potentially be addressed by using complex text filtering to remove unnecessary text from the web page, such as by extracting text from specific tags like headings and paragraphs.
4. One advantage of using TextRank was that the algorithm does not rely on linguistic features of the text, making the extension general enough to work on web pages in any language.

In the future, the advancement of text summarization using large language models like GPT-3 could enable the development of various NLP-based applications in fields such as media, healthcare, and technology. One potential application that does not currently exist is a **smart text-to-infographics generator** that takes multiple sources of text, summarizes the content, and presents it in a visually rich infographic.[7] This could be useful for users who want to quickly grasp general concepts on a topic and could even help them prepare presentations.

In general, many applications could be built using text summarization and visualization to present text in a more user-friendly and easy-to-digest manner.[8]

References

[1] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan and M. M. Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges," in IEEE Access, vol. 9, pp. 156043-156070, 2021, doi: 10.1109/ACCESS.2021.3129786.

[2] https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html

[3] Dou, Zi-Yi, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang and Graham Neubig. "GSum: A General Framework for Guided Neural Abstractive Summarization." ArXiv abs/2010.08014 (2020): n. pag.

[4] Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Annual Meeting of the Association for Computational Linguistics (2019).

[5] TextRank: Bringing Order into Text (Mihalcea & Tarau, EMNLP 2004).

[6] Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries." Annual Meeting of the Association for Computational Linguistics (2004).

[7] https://venngage.com/blog/how-to-summarize/

[8] Cui, Weiwei & Zhang, Xiaoyu & Wang, Yun & Huang, He & Chen, Bei & Fang, Lei & Zhang, Haidong & Lou, Jian-Guang & Zhang, Dongmei. (2019). Text-to-Viz: Automatic Generation of Infographics from Proportion-Related Natural Language Statements. IEEE Transactions on Visualization and Computer Graphics. PP. 1-1. 10.1109/TVCG.2019.2934785.