CIS5300 Project MS1

Wanqing Ding(dwanqing@seas.upenn.edu), Eileen Feng(yazhefen@seas.upenn.edu),
Crescent Xiong (zihanx3@seas.upenn.edu), Leila Zhu(leilazhu@seas.upenn.edu)

Project GitHub: https://github.com/EileenFeng/Text-Summarization-NLP

Dataset Description: https://github.com/EileenFeng/Text-Summarization-NLP/blob/main/data.md

Literature review

Text Summarization with Pretrained Encoders[1]

Liu et al. introduced a general framework for both extractive and abstractive summarization that showcased how pre-trained BERT can be applied as a document-level encoder, named BERTSUM, that can express the semantics of documents, capture sentences representation, and enable multi-sentence manipulation. An interval segment embedding is introduced to allow document representations to be learned hierarchically. Length of position embeddings are also 'extended' by adding randomly initialized embeddings that are later fine-tuned with other parameters in the encoder.

Extractive summarization is defined as a binary classification task, where labels indicate whether a text span is included in the summarization or not. The extractive classifier compose of several inter-sentence transformer layers stacked on top of the BERTSUM encoder, with sigmoid as the output layer. Sentences are scored, sorted, with top three chosen as the summary. The inter-sentence Transformer layers are jointly fine-tuned with BERTSUM. **The abstractive model** follows the standard encoder-decoder framework, with BERTSUM as the encoder, and a randomly initialized 6-layer Transformer as the decoder. A new fine-tuning schedule is introduced to separate the optimizers of both to mitigate that the encoder is pre-trained yet the decoder is not. The encoder is first fine-tuned on the extractive task and then on the abstractive task, taking advantage of the information shared between the two tasks to boost performance[2]. All linear layers are trained with dropout of 0.1, and label smoothing factor of 0.1. Beam-5 search and fine-tuned length penalty are applied for decoding. Summary is generated until the end-of-sentence is emitted. Experiments ran over different datasets with both lengthy and one-sentence summaries. Models are trained with gradient accumulation, and checkpointed periodically on validation set for evaluation. Metrics include ROUGE-1 (unigram overlap), ROUGE-2 (bigram) and ROUGE-L (longest common subsequence). A non-pretrained Transformer with the same architecture as BERTSUM yet fewer parameters is used as the baseline. Results show that BERTSUM-based models outperform baseline and a few previous works for all datasets. Further analysis shows the BERTSUM model tends to select sentences located later in the document compared to the baseline model, indicating that it is able to learn deeper document representations.

Extractive Summarization as Text Matching [3]

This paper introduces a novel approach by framing extractive summarization tasks as a semantic text matching problem, departing from the traditional sentence-level extraction methods. One of the primary criticisms of sentence-level extractors lies in their failure to capture the overarching semantics of an entire summary. This shortcoming is especially apparent in the inability of sentence-level extractors to distinguish crucial components, such as pearl-summaries[4], within the broader context. To address these limitations,

---

[1] https://arxiv.org/abs/1908.08345
[2] Sebastian Gehrmann, YuntianDeng, andAlexander Rush.2018. Bottom-upabstractivesummarization. InProceedingsof the2018ConferenceonEmpiricalMethodsinNaturalLanguageProcessing,pages 4098–4109,Brussels,Belgium.
[3] https://aclanthology.org/2020.acl-main.552.pdf
[4] A summary that has a lower sentence-level score but a higher summary-level score.

the study undertakes a comprehensive comparative analysis across six benchmark datasets—Reddit, XSum, CNN/DM, WikiHow, PubMed, and Multi-news. The findings reveal that the efficacy of different extractors is contingent upon the dataset in question. Interestingly, summary-level extractors consistently outperform their sentence-level counterparts across most datasets, underscoring the importance of a holistic approach to summarization tasks.

This paper proposed MatchSum, a summary-level extractor designed around the Siamese-BERT architecture. MatchSum's key innovation lies in its ability to compute the semantic similarity between the source document and a candidate summary, addressing the limitations identified in traditional methods. Notably, the study evaluates the performance of MatchSum against several prominent competitors, including LEAD, ORACLE, MATCH-ORACLE, and BertExt. Employing the ROUGE-1 score as the metric of comparison, MatchSum emerges as the frontrunner, consistently outperforming its counterparts across all six benchmark datasets. This achievement signifies a notable leap forward in the realm of extractive summarization, shedding light on the potential of summary-level extraction models to discern and capture crucial information within a document.

Text Summarization Techniques: A Brief Survey[5]

This paper presents an overview of different extractive methods until 2017 for tackling single and multi-document summarization tasks. The author indicates that extractive summaries often provide better results than automatic abstractive summaries, considering by then there’s no pure abstractive summarization system but rather abstractive summarizers rely on some extractive preprocessing components.

The approaches generally consisted of three stages: (1) constructing intermediate representations for every sentence, (2) scoring these sentences based on representations, (3) selecting a summary from these sentences. There are two types of approaches for Stage 1: topic representation and indicator representation. The former transforms the input text in a way such that the resulting representations reveal the topics uncovered in the sentences, while the latter utilizes a set of features to indicate the importance of sentences, including sentence length, sentence positions, etc. For topic representation, frequency-driven approaches such as word probability, TF-IDF, centroid-based summarization, are illustrated. Besides, there are topic words and latent semantic analysis (LSA) approaches. All the previously mentioned approaches have the drawback of assuming independence among sentences and the disregard of topics that are embedded within the documents. Moreover, these approaches rely heavily on heuristics and an explicit probabilistic interpretation is missing. Bayesian topic models, however, compensate for these two drawbacks. Among these models, Latent Dirichlet Allocation (LDA) has been working very well, especially for summarizing multiple documents. Adding knowledge bases like semantics or ontologies also improve the performance of summarizers. For indicator representation, graph-based methods inspired by PageRank and machine learning methods are widely used in summarizing text. At the end, besides human evaluation, Recall Oriented Understudy for Gisting Evaluation (ROUGE) is the most widely used metric for automatic evaluation. Some popular variants of ROUGE being used include ROUGE-n, ROUGE-L, and ROUGE-SU.

[5] Mehdi Allahyari, Seyedamin Pouriyeh, Medhi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text Summarization techniques: A brief survey, 2017.