UETfishes at MEDIQA 2021: Standing-on-the-Shoulders-of-Giants Model for Abstractive Multi-answer Summarization

Hoang-Quynh Le[1], **Quoc-An Nguyen**[1], **Quoc-Hung Duong**[1], **Minh-Quang Nguyen**[1]
Huy-Son Nguyen[1], **Tam Doan Thanh**[2], **Hai-Yen Thi Vuong**[1] and **Trang M. Nguyen**[1]
[1]VNU University of Engineering and Technology, Hanoi, Vietnam.
{lhquynh, 18020106, 18020021, 19020405}@vnu.edu.vn
{18021102, yenvth, trangntm}@vnu.edu.vn
[2]doanthanhtam283@gmail.com

Abstract

This paper describes a system developed to summarize multiple answers challenge in the MEDIQA 2021 shared task collocated with the BioNLP 2021 Workshop. We present an abstractive summarization model based on BART, a denoising auto-encoder for pre-training sequence-to-sequence models. As focusing on the summarization of answers to consumer health questions, we propose a query-driven filtering phase to choose useful information from the input document automatically. Our approach achieves potential results, rank no.2 (evaluated on extractive references) and no.3 (evaluated on abstractive references) in the final evaluation.

1 Introduction

In the past several decades, biomedicine and human health care have become one of the major service industries. They have been receiving increasing attention from the research community and the whole society. The rapid growth of volume and variety of biomedical scientific data make it an exemplary case of big data (Soto et al., 2019). It is an unprecedented opportunity to explore biomedical science and an enormous challenge when facing a massive amount of unstructured and semi-structured data. The development of search engines and question answering systems has assisted us in retrieving information. However, most biomedical retrieved knowledge comes from unstructured text form. Without considerable medical knowledge, the consumer is not always able to judge the correctness and relevance of the content (Savery et al., 2020). It also takes too much time and labour to process the whole content of these documents rather than extracting the useful compressed content. *Automatic summarization* is a challenging application of biomedical natural language processing. It generates a concise description that captures the salient details (called summary) from a more complex source of information (Mishra et al., 2014). Summarization can be particularly beneficial for helping people easily access electronic health information from search engine and question answering systems.

MEDIQA 2021[1] (Ben Abacha et al., 2021) tackles three summarization tasks in the medical domain. Task 2- Summarization of Multiple Answers challenge aims to promote the development of multi-answer summarization approaches that could simultaneously solve the aggregation and summarization problems posed by multiple relevant answers to a medical question.

There are two approaches to summarization: extractive and abstractive. Extractive summarization, i.e., choose important sentences from the original text, is extensively researched but have several limitations: (i) it is unable to keep the coherence of the answer, (ii) the information compressed may be incomplete because information may take many sentences to expose, and (iii) it must include non-relevant part of a relevant sentence. Recently, the research has shifted towards more promising approaches, i.e. abstractive summarization, which can overcome these problems give higher precision than extractive summaries (Gupta and Gupta, 2019). Abstractive text summarization is the task of generating a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text. Abstractive summarization helps resolve the dangling anaphora problem and thus helps generate readable, concise and cohesive summaries. In abstractive summary, we can merge several relate sentences or make them shorter, i.e., removing the redundancy part.

Our proposed model for the multi-answer summarization task follows abstractive summarization

[1]https://sites.google.com/view/mediqa2021

Proceedings of the BioNLP 2021 workshop, pages 328–335

approaches. We try to process original answers as a shorter representation while preserving information content and overall meaning. We take advantage of BART, a pre-trained model combining bidirectional and auto-regressive transformers (Lewis et al., 2020). We construct an architecture with two filtering phases to choose the more concise input for BART. Since the summary should be question-oriented, the coarse-grained filtering phase removes question-irrelevant sentences. The fine-grained filtering phase is then used to cut-off noise phases.

The remaining of this paper is organized as follows: Section 2 gives brief introduction of some state-of-the-art related work. Section 3 describes task data and our proposed model. Section 4 is the experimental results and our discussion. And finally, the Conclusion.

2 Related work

Because of the complexity of natural language, abstractive summarization is a challenging task and has only been of interest in recent years. Gerani et al. (2014) proposed an abstractive summarization system for product reviews by taking advantage of their discourse tree structure. A important subgraph in the discourse tree were then selected by using PageRank algorithm. A natural language summary was then generated by applying a template-based NLG framework.

According to current research trends, witnessing the success of deep learning in other NLP tasks, researchers have started considering this framework as an promising solution for abstractive summarization. Nallapati et al. (2016) used an attentional encoder-decoder recurrent neural networks and several models such as key-words modeling, sentence-to-word hierarchy structure, and emitting rare words, etc. Song et al. (2019) proposed an LSTM-CNN based ATS model to construct new sentences by exploring fine-grained phrases from source sentences (of CNN and DailyMail) and combining them. Gehrmann et al. (2018) used a bottom-up attention technique to improve the deep learning model by over-determining phrases in a source document that should be part of the summary. Inspired by the successful application of deep learning methods for machine translation, abstractive text summarization is specifically framed as a sequence-to-sequence learning task. BART is a transformer-based pretrained denoising encoder-decoder model that is applicable to a very wide range of end tasks, includes summarization. It combines a bidirectional encoder and an auto-regressive decoder (Lewis et al., 2020). There are several BART-based model, example includes DistilBart[2] and Question-driven BART (Savery et al., 2020). Question-driven BART re-trained BART on objectives designed to improve its general ability to understand the content of text (including document rotation, sentence permutation, text-infilling, token masking and token deletion) and fine-tuned the model for biomedical data. Another recently published abstractive summarization framework is PEGASUS (Zhang et al., 2020), it masks important sentences and generates those gap-sentences from the rest of the document as an additional pre-training objective.

3 Materials and Methods

3.1 Shared task data

The shared task suggested to use the MEDIQA-AnS Dataset (Savery et al., 2020) as the training Data. The validation and test sets includes the original answers are generated by the medical question answering system system CHiQA[3] . In these data sets, extractive and abstractive summaries are manually created by medical experts. Table 1 gives our statistics on the given datasets (see (Ben Abacha et al., 2021) for detailed description of shared task data).

Table 1: Statistics of the datasets.

Statics aspects	Training		Validation	Test
	Article	Section		
Question	156	156	50	80
Average				
A per Q	3.54	3.54	3.85	3.80
T per A	152.35	532.83	219.44	240.22
T per SSum	70.51	70.51	-	-
T per MSum	119.04	119.04	81.18	-
Compression radio				
SSum	0.07	0.32	-	-
MSum	0.04	0.13	0.15	-

A: Answer, Q: Question, T: Token
SSum: Single-answer summary,
MSum: Multi-answer summary.

3.2 Proposed model

As a team participating in MEDIQA - Task 2, we proposed an abstractive summarization sys-

[2] https://huggingface.co/sshleifer/DistilBart-cnn-12-6

[3] https://chiqa.nlm.nih.gov

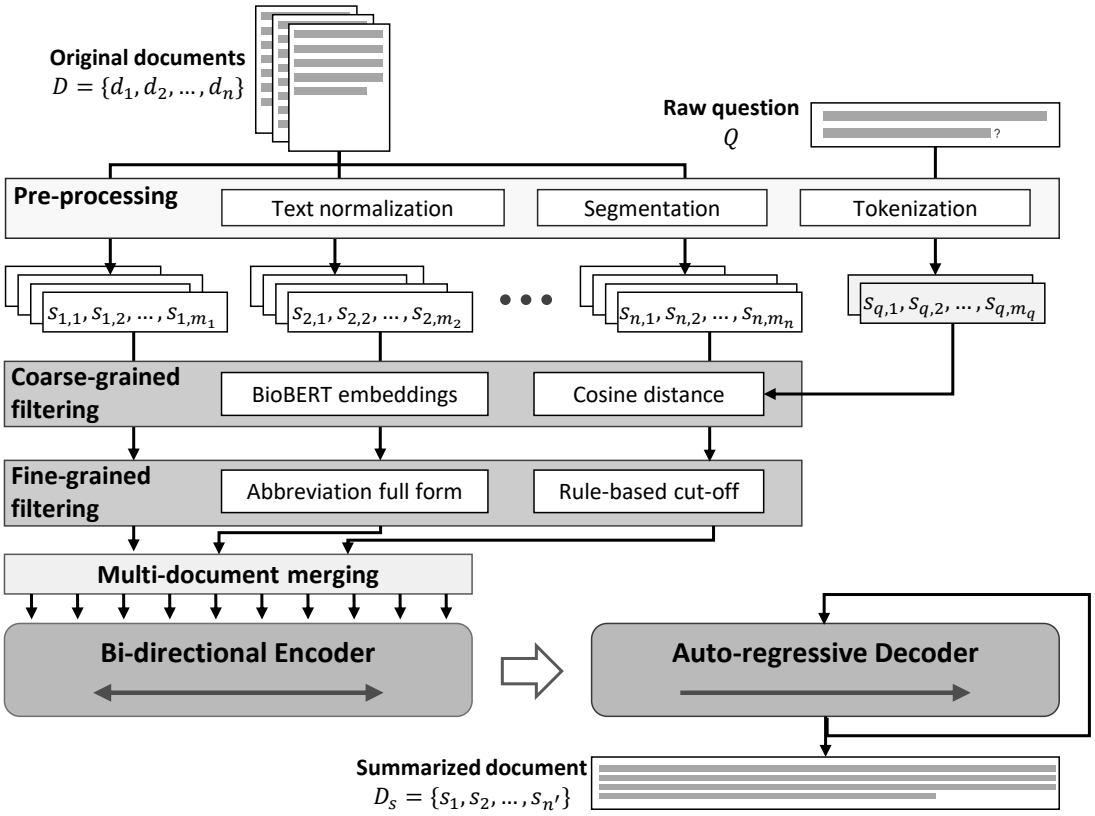

Figure 1: The proposed 'Standing-on-the-Shoulders-of-Giants' model.

tem based on BART - the denoising sequence-to-sequence model. We designate this as a 'Standing-on-the-Shoulders-of-Giants' (SSG) model because BART is the recently state-of-the-art model for abstractive summarization task. To improve the performance, we propose to apply two filtering phases to make the condensed question-driven input for BART. In addition, the BART-based model only receives a limited length document *(with 1024 tokens)*, and our original input is too large to fit. Our model requires a cut-off strategy to reduce length. The overall architecture of the system is described in Figure 1 which includes five main phrases: pre-processing, coarse-grained filtering, fine-grained filtering phase and BART-based summary generation.

3.2.1 Pre-processing

The pre-processing phase receives question Q and a set of corresponding answers (documents) $D = \{d_i\}_{i=1}^{n}$ as the input. The pre-processing phase removes html tags, non-utf-8 characters and redundant signs/spaces. scispaCy (Neumann et al., 2019), a powerful tool for biomedical natural language processing, is also used for the typical pre-processing steps (i.e. segmentation and tokenization).

3.2.2 Coarse-grained filtering

The original BART summarizes a text by generating a shorter text with the same semantic. It processes all information with the same priority and does not take the question into account. Therefore, its output may lose the function of answering the question. We orient BART to question-driven by filtering out less valuable sentences, increasing the rate of question-related sentences in the BART input. There are two strategy to choose sentences that are highly related to the questions:

(i) **Top-n query-driven sentences:** The main idea of this method is to choose sentences that most likely can answer the questions. We calculate the cosine similarity between two bioBERT embedding vectors (Lee et al., 2020) of the question and each sentence. We assume that the sentence with higher cosine similarity might be a good answer for the question. The top-n sentences of each answer with the highest scores are kept with their original orders.

(ii) **Top-n query-driven passages:** Some passages are structured in an deductive manner (e.g., several explanatory sentences follow after a stated

sentence) or inductive (e.g., the last sentence is the conclusion of previous sentences). Extracting these whole text pieces may help an important sentence have some adjacent sentences to clarify or support it, making it more coherence and informative. There are three factors to determine an important passage:

- *Central sentence:* A passage is chosen if and only if it has at least one sentence likely answering the question. Cosine similarity with BioBERT embedding vector is used to find these sentences.
- *Passage length:* A passage must not exceed k sentences.
- *Break point:* If the similarity between two adjacent sentences is lower than a pre-defined threshold, a breakpoint is addressed.
- *Passage score:* is calculated by the sum of its sentences similarity scores.

Top-n best passages are then combined with their original order.

In addition to two aforementioned strategies, we also use two other simple strategies as the baseline:

(iii) n **first sentences:** Taking n first sentences from each answers.

(iv) n **random sentences:** Taking n random sentences from each answers.

In which, the number of passages/sentences is not limited which satisfies that the whole length of final document is fit of smaller than the allowed input size of BART model. It should take as much information as possible.

3.2.3 Fine-grained filtering

The nature of BART is to convert one piece of text into another with the same semantics. If the input contains too much noise and is difficult to understand, it may negatively affect the output quality. Therefore, we try to filter out the noise phrases to get the most concise input to BART, thereby getting better results. Through the data surveying, there are two approaches to reduce noises and ambiguous information:

(i) Biomedical text uses many abbreviations, of which many do not follow a standard convention and are only used locally within the scope of authors' articles. Unfortunately, these local abbreviations might be the keywords and lead to the ambiguous to the system. We identify and generate the full form of all local abbreviation use the Ab3P tool (Sohn et al., 2008).

(ii) we apply some rules to cut redundant elements of sentences. Examples include:

- Cut-off listed text that follows *'such as'*.
- Cut-off text that follows *'for example'*.
- Cut-off text that appears in the brackets *()*.
- Cut-off text that follows a colon and is not in enumerated form.

3.2.4 BART-based summary generation

All sentences are selected and cut-off from aforementioned filtering phases are then combined into a single document. This is the input to the BART-based summary generation phase.

BART is implemented as a standard sequence-to-sequence Transformer-based model. It is a denoising autoencoder that maps a corrupted document to the original document it was derived from (Lee et al., 2020). Special power of this model is that it can map the input string and output string with different lengths. BART consists of two components: Encoder and Decoder that combines the advantages of BERT and GPT.

Encoder: BART uses a bidirectional encoder over corrupted text taken from BERT (Devlin et al., 2019). As the strength of BERT lies in capturing two-dimensional contexts, BART can encode the input string in both directions and get more context information. In the abstractive text summarization problem, the input sequence is the collection of all token in the answers. Each word is represented by x_t, where i is its ordinal. The h_t hidden states are calculated with the formula:

$$h_t = f(W^{hh} \cdot h_{t-1} + W^{hx} \cdot x_t) \qquad (1)$$

in which, the hidden states are computed by the corresponding input x_t and the previous hidden state h_{t-1}. Encoder vector is the hidden state at the end of the string, calculated by the encoder. It then acts as the first hidden state of the decoder.

Decoder: BART uses a left-to-right autoregressive decoder. Its decoder is similar to GPT (Radford et al.) with the capability of self-regression, can be used to reconstruct the input noise. A stack of subnets is the element of the RNN that predicts the output y_t at time t. Each of these words takes input as the previously hidden state and produces its own output and hidden state.

For the abstractive text summarization problem, the output sequence is the set of words of the summarized answer. Each word is represented by y_t where i is the word order. The hidden state is calculated by the preceding state. So, the h_i hidden states are calculated by the formula:

$$h_t = f(W^{hh} \cdot h_{t-1}) \quad (2)$$

We compute the output using the corresponding latency at the present time and multiply it by the corresponding weight W^S. Softmax is used to create a probability vector that helps us to determine the final output. The output y_t are calculated by the formula:

$$y_t = softmax(W^S \cdot h_t) \quad (3)$$

BART uses Beam Search algorithm for decoding.

4 Experimental results

4.1 Evaluation metrics

We adopt the official task evaluations with ROUGE scores (Lin and Och, 2004) including ROUGE-1, ROUGE-2 and ROUGE-L. ROUGE-n Recall (R), Precision (P) and $F1$ between predicted summary and referenced summary are calculated as in Formular 4, 5 and 8, respectively. Choosing correct sentences help to increase ROUGE-n R and P.

$$\text{ROUGE-}n\ P = \frac{|\text{Matched N-grams}|}{|\text{Predict summary N-grams}|} \quad (4)$$

$$\text{ROUGE-}n\ R = \frac{|\text{Matched N-grams}|}{|\text{Reference summary N-grams}|} \quad (5)$$

$$\text{ROUGE-L}\ P = \frac{\text{Length of the LCS}}{|\text{Predict summary tokens}|} \quad (6)$$

$$\text{ROUGE-L}\ R = \frac{\text{Length of the LCS}}{|\text{Reference summary tokens}|} \quad (7)$$

ROUGE-L recall (R), precision (P) and $F1$ are calculated as in Formular 6, 7 and 8, respectively. ROUGE-L uses the Longest Common Subsequence (LCS) between predicted summary and referenced summary and normalized by the tokens in summary.

$$F1 = 2 \times \frac{\text{R} \times \text{R}}{\text{P} + \text{R}} \quad (8)$$

4.2 Comparative models

We use the official results of the MEDIQA shared task as a comparison to other participated teams on the multi-answer summarization task. For a further comparison, we also make the comparisons with three state-of-the-art abstractive summarization models:

- The orginal BART (Lewis et al., 2020).
- DistilBart[4]: A very effective model for text generation task release by HuggingFace.
- PEGASUS (Zhang et al., 2020) is state-of-the-art abstractive summarization model provided by Google AI.

4.3 Task final results and comparison

Based on the experimental results on the validation set, we choose top-n query-driven passages as a coarse-grained filter to run our official output. In our model, Beam Search uses $beamwidth = 5$ and uses sampling instead of greedy decoding. Beam Search is stopped when at least 5 sentences finished per batch. After two filtering phases, the input often have 10-15 sentences and less than 1024 tokens. On average, the total token in a summary is equal to $\sim$75% of the number of tokens in the BART input.

4.3.1 Official results of the multi-answer abstractive summarization

Table 2 show the shared task official results of accepted competitors. ROUGE-2 $F1$ is used as the main metric to rank the participating teams. We also show several other evaluation metrics for further comparison: ROUGE-1 $F1$, ROUGE-L $F1$, HOMLS $F1$ and BERT-based $F1$. The organizers offer two rankings, one on the extractive references, the other on the abstractive references. Evaluated on extractive references, our team is the runner-up. On the evaluation using abstractive references, we ranked third.

4.3.2 Comparison with other state-of-the-art models

Table 3 shows the comparison between our proposed model and two other state-of-the-art text generation models, i.e., DistilBart and Pegasus. Our SSG model yields much better results than DistilBart and PEGASUS in this data. Since both models

[4] https://huggingface.co/sshleifer/distilbart-cnn-12-6

Table 2: Official results of the MEDIQA 2021: Task 2 - Multi-Answer Summarization

Team	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1	HOLMS	BERTscore F1
Evaluated on extractive references					
paht_nlp	0.585	**0.508**	**0.436**	**0.554**	**0.653**
UETfishes	0.572	0.470	0.400	0.520	0.646
UCSD-Adobe	**0.592**	0.460	0.417	0.493	0.632
yamr	0.516	0.445	0.384	0.536	0.636
I_have_no_flash	0.523	0.422	0.360	0.542	0.615
Evaluated on abstractive references					
paht_nlp	**0.386**	**0.162**	**0.232**	**0.554**	**0.653**
UCSD-Adobe	0.384	0.160	0.212	0.494	0.632
UETfishes	0.381	0.147	0.202	0.520	0.647
I_have_no_flash	0.384	0.133	0.222	0.478	0.615
yamr	0.271	0.131	0.160	0.388	0.636

Only show results of top-5 participated teams for each type of evaluation.
The highest results in each column are highlighted in bold.

Table 3: Comparison with other state-of-the-art models.

Model	ROUGE-2		
	P	R	F1
DistilBART	0.0825	0.1031	0.0874
Pegasus	0.0401	0.0597	0.0450
Our SSG	0.0977	0.1274	0.1062

All results are reported on the validation data set.

are very strong competitors, our higher outcome may because they are not suitable with the characteristics of the data (biomedical domain, question-driven answers).

4.4 Contribution of model components

We study the contribution of each model component to the system performance by ablating each of them in turn from the model and afterwards evaluating the model on the validation set. We compare these experimental results with the full system results and then illustrate the changes of ROUGE-2 $F1$ in Figure 2. The changes of ROUGE-2 $F1$ show that all model components help the system to boost its performance (in terms of the increments in ROUGE-2 $F1$). The contribution, however, varies among components. The coarse-grained filtering phase has the biggest contribution, while abbreviation processing and cut-off rules of the fine-grained phase bring very small effectiveness. We also investigate the effectiveness of components/configures in the BART-based summary generation. Components that have a pronounced effect on the result are shown in Figure 2 : Preventing 3-gram repeater, sampling, early stopping and beam search. Preventing 3-gram repeater and using sampling also improves results.

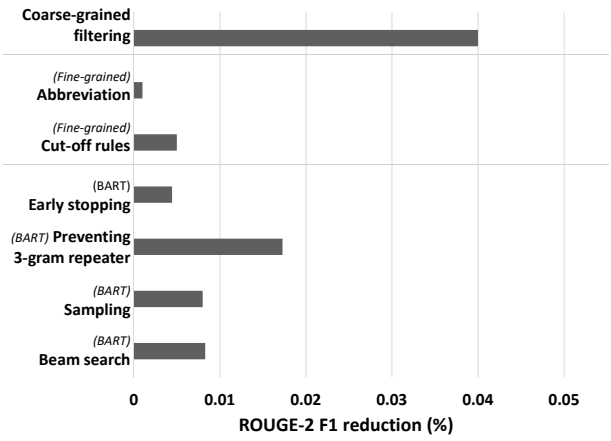

Figure 2: Ablation test results for model components.

Considering the results of three different approaches in the *coarse-grained filtering phase* (Figure 3), top-n question-driven passage seems the most promised way. Other approaches do not take advantages of the semantic relation between adjacent sentences, which leads to losing important information.

4.5 Error analysis

In order to improve the proposed model, we have analyzed the output on the validation set to find out problems that need to be taken into account. All the evidence points to five biggest problems, including content generalization, synonyms and antonyms, paraphrasing, cosine similarity problem, and aggressive cut-out strategy.

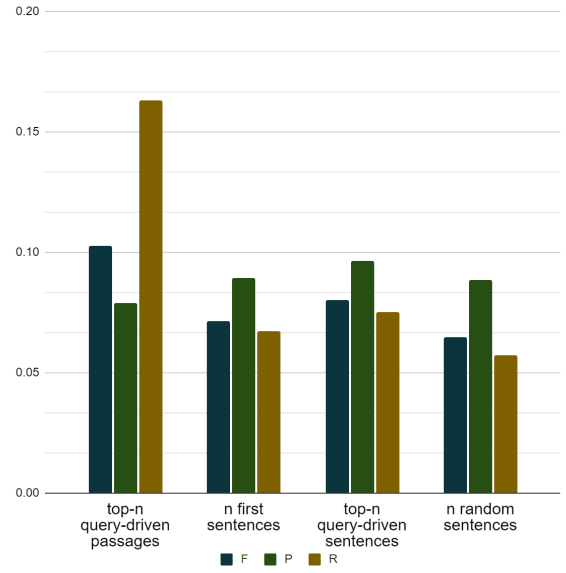

Figure 3: Comparison of different coarse-grained filtering strategies based on ROUGE-2 scores.

The biggest problem with our proposal model and other text summary models is the generalization of the input content. In particular, for the answer summary system, this issue is emphasized more and more. The responses may contain a variety of content related to the directional question. However, the summary should draw conclusions to answer that question. For example, in Question #22, to answer the question *'Is it safe to have ultrasound with a defibrillator?'*, our model performed well that carried out the summary *'Most of the time, ultrasound procedures do not cause discomfort. The conducting gel may feel a little cold and wet. Current ultrasound techniques appear to be safe.'* However, the expected outcome was *'There are no known risks or contraindications for ultrasound tests.'* For which, our model gets a 0.0 ROUGE-2 F1 score for this example.

Another problem is that golden data depends on the style and language usage of the abstractor. The writer may use different expressions, synonyms, antonyms to paraphrase and summarise, leading to the inconsistency of ground truth data. Take Question #8 for example, the sentence *'This treatment leads to remission in 80% to 90% of patients'* is paraphrased into *'Remission is possible in up to 90% of the patients.'*

The analysis process also raises some imperfections of the proposed model in sentence selection and sentence cutting strategies. Cosine similarity metric does not really perform well with documents containing many sentences. In particular, many sentences contain important content but do not have high similarity to the question. Besides, fine-grained filtering strategies also filter some important information in the sentence. We leave these problems to be addressed in future work.

5 Conclusion

This paper presents a systematic study of our abstractive approach to question-driven summarization problem, specifically depending on MEDIQA 2021 - Task 2: Multi-answer summarization. We present a model improved and optimized based on BART - a state-of-the-art method for abstractive summarization called SSG (Standing on the shoulders of giants). The proposed model has a potential performance, being the runner-up of the shared task. Our best performance achieved a ROUGE-2 $F1$ is 0.470 evaluated on extractive summarization references and 0.147 evaluated on abstractive summarization references .

Experiments were also carried out to verify the rationality and impact of model components and the compressed ratio. The results demonstrated the contribution and robustness of all techniques and hyper-parameters. Besides, the error analysis was made to analyze the sources of the errors. The evidence pointed out some imperfection of the sentence selecting strategy, the ranking score combination, and the question analyzer. In further works, there could be several ways: applying machine learning model, deeply question-analyzing, sentence clustering, etc. applied to extend the ability of the model.

Our source code will be released publicly to support the reproducibility of our work and facilitate other related studies.

Acknowledgements

We would like to thank the organizing committee of MEDIQA NAACL-BioNLP 2021 shared task. We also thank the anonymous reviewers for thorough and helpful comments.

References

Asma Ben Abacha, Yassine Mrabet, Yuhao Zhang, Chaitanya Shivade, Curtis Langlotz, and Dina Demner-Fushman. 2021. Overview of the mediqa 2021 shared task on summarization in the medical domain. In *Proceedings of the 20th SIGBioMed Workshop on Biomedical Language Processing, NAACL-BioNLP 2021*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1602–1613.

Som Gupta and SK Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin and FJ Och. 2004. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir Workshop*.

Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R Weir, Siddhartha Jonnalagadda, Javed Mostafa, and Guilherme Del Fiol. 2014. Text summarization in the biomedical domain: a systematic review of recent research. *Journal of biomedical informatics*, 52:457–467.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Max Savery, Asma Ben Abacha, Soumya Gayen, and Dina Demner-Fushman. 2020. Question-driven summarization of answers to consumer health questions. *Scientific Data*, 7(1):1–9.

Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. 2008. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9(1):1–10.

Shengli Song, Haitao Huang, and Tongxiao Ruan. 2019. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78(1):857–875.

Axel J Soto, Piotr Przybyła, and Sophia Ananiadou. 2019. Thalia: semantic search engine for biomedical abstracts. *Bioinformatics*, 35(10):1799–1801.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.