UETrice at MEDIQA 2021: A Prosper-thy-neighbour Extractive Multi-document Summarization Model

Duy-Cat Can[1], **Quoc-An Nguyen**[1]*, **Quoc-Hung Duong**[1], **Minh-Quang Nguyen**[1]
Huy-Son Nguyen[1], **Linh Nguyen Tran Ngoc**[2], **Quang-Thuy Ha**[1] **and Mai-Vu Tran**[1]†
[1]VNU University of Engineering and Technology, Hanoi, Vietnam.
{catcd, 18020106, 18020021, 19020405}@vnu.edu.vn
{18021102, thuyhq, vutm}@vnu.edu.vn
[2]Viettel Big Data Analytics Center, Viettel Telecommunication Company, Viettel Group.
linhntn3@viettel.com.vn

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 propose an extractive summarization architecture based on several scores and state-of-the-art techniques. We also present our novel prosper-thy-neighbour (PtN) strategies to improve performance. Our model has been proven to be effective with the best ROUGE-1/ROUGE-L scores, being the shared task runner-up by ROUGE-2 $F1$ score (over 13 participated teams).

1 Introduction

Biomedical documents are available with the tremendous amount on the Internet, together with several search engines (e.g., Pubmed®[1]) and question-answering systems (e.g., CHiQA[2]) developed. However, the returned results of these systems still contain a lot of noise and duplication, making them difficult for users without medical knowledge to quickly grasp the main content and get the necessary information. Hence, generating a shorter condensed form with important information would benefit many users as it saves time and can retrieve massive useful information. This motivation leads to the growing interest among the research community in developing automatic text summarization methods. The BioNLP-MEDIQA 2021 shared task[3] (Ben Abacha et al., 2021) aims to attract further research efforts in text summarization and their applications in medical Question-Answering (QA). This shared task is motivated by a need to develop relevant methods, techniques, and gold standards for text summarization in the medical domain and their application to improve the domain-specific QA system. Task 2 - Summarization of Multiple Answers focuses on developing multi-document summarization approaches that could synthesize and compress information from answers to a medical question.

According to Radev et al. (2002) a summary is defined as *'a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that'*. Automatic text summarization is the task of condensing the document(s) and generating a compressed summary, which is shorter but preserves key information content and overall meaning. A summary can be generated through extractive or abstractive approaches (or hybrid). Typically, to produce an abstractive summarization, we need to use advanced linguistic techniques to 'understand' the text as well as re-generate the summary in natural language from useful information. Up to now, the research community is focusing more on extractive summarization. This approach tries to achieve coherent and meaningful summaries in a more simple and faster way than the abstractive approach. Extractive summarization chooses important sentences (or phrases) from the original documents (without any modification) and merges them to generate a summary.

Our proposed model for the multi-answer summarization task follows extractive summarization approaches. We try to select sentences containing the most important information in the original answers. Our novel contributions are: (i) Proposing the question-driven scores to ensure that the summary is the answer to the question, (ii) Proposing Prosper-thy-neighbour (PtN) strategies, which increase the constraint of neighbouring sentences, to take advantage of paragraph information in the answer. (iii) Combining several scores that successfully applied for summarization problem, includ-

*Contributed equally & Names are in alphabetical order
†Corresponding author
[1]https://pubmed.ncbi.nlm.nih.gov/
[2]https://chiqa.nlm.nih.gov/
[3]https://sites.google.com/view/mediqa2021

Proceedings of the BioNLP 2021 workshop, pages 311–319
June 11, 2021.

ing TF-IDF, Lexrank, and Textrank with optimized weights, (iv) Improving the maximal marginal relevance technique (MMR) for multi-document summarization with BERT-based embedding to improve the performance.

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

2 Related works

From the early 1950s, various methods have been proposed for extractive summarization (Allahyari et al., 2017). Some of them are based on the idea of using scores to choose the most important phrases in the documents. Term Frequency-Inverse Document Frequency (TF-IDF) (Hovy et al., 1999; Christian et al., 2016) is a frequency-based score to detect important sentences by calculating the scores of its words. Lexrank (Erkan and Radev, 2004) and Textrank (Mihalcea and Tarau, 2004) are two graph-based methods that rank sentences/words using their degree centrality. Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998; Bennani-Smires et al., 2018) is one of the most well-known approaches for multi-document summarization. It is a diversity-based re-ranking method based on the document similarities and can be used to remove redundancy in the summaries. Although encouraging results have been reported, most of these scores are applied individually. Since each score type has its unique contribution, combining them may help to improve the performance. Hence, we propose an architecture to take advantage of several scores with weights and calculate a final combined score.

With the advent of machine learning techniques in NLP, many research projects tried to apply machine learning methods to extractive summarization tasks, from the Naive Bayes, Decision tree, Support vector machine (Gambhir and Gupta, 2017) to deep learning models. Most recently, Savery et al. (2020) improved the Bidirectional auto regressive transformer (BART) with a question-driven approach, but it is more well-known for abstractive summarization, which is not discussed in-depth in this paper.

3 Materials and Methods

3.1 Shared task data

The MEDIQA-AnS Dataset (Savery et al., 2020) is used as the training data set. The validation and the test sets are the summaries that were created by the experts from the original answers generated by the question-answering system namely CHiQA[4]. Table 1 gives our statistics on the given datasets (see (Ben Abacha et al., 2021) for detailed description of shared task data).

An important observation is that answers often tend to have related sentences in a passage that makes an important 'point'. Some adjacent sentences are structured in a 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 pieces of text ensures a complete summary while enhancing fluency and natural language resemblance. Our prosper-thy-neighbour strategies are proposed to take advantage of this characteristic.

Table 1: Statistics of the datasets.

Statistic aspects	Training		Validation	Test
	Article	Section		
Questions	156	156	50	80
Average				
A per Q	3.54	3.54	3.85	3.80
Sent per A	84.93	29.07	14.50	13.03
Sent per SSum	6.31	6.31	-	-
Sent per MSum	10.30	10.30	11.06	-
Compression ratio				
SSum	0.12	0.49	-	-
MSum	0.06	0.18	0.33	-

A: Answer, Q: Question, Sent: Sentence, SSum: Single-answer Summary, MSum: Multi-answer Summary

3.2 Proposed model

The overall architecture of our Prosper-thy-Neighbour (PtN) summarization model is shown in Figure 1. It comprises four main phases: pre-processing, single document summarization, multi-document summarization and post-processing phases.

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. ScispaCy (Neumann et al., 2019), which is based on SpaCy (Honnibal et al.,

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

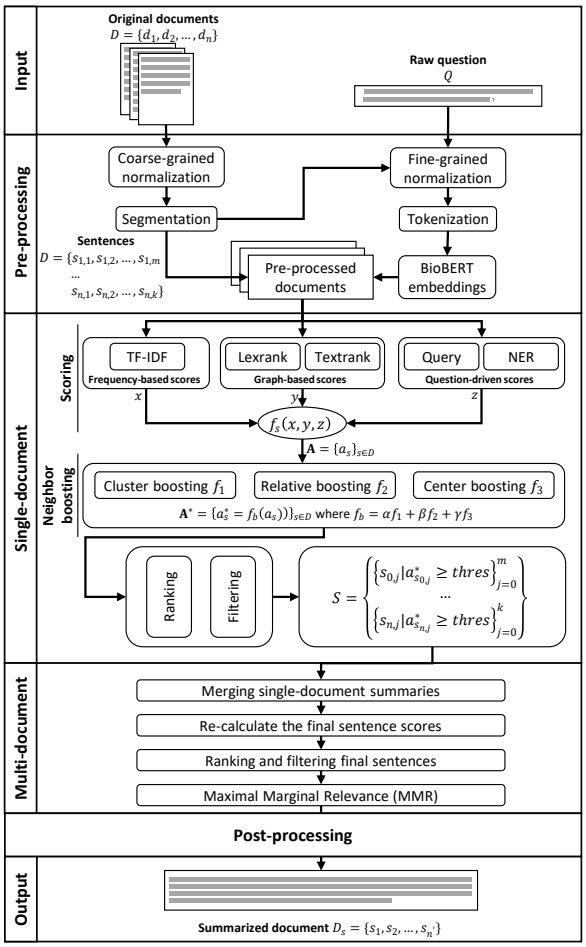

Figure 1: The proposed Prosper-thy-neighbour model.

2020) models, is used for the typical pre-processing techniques (i.e. segmentation and tokenization) in terms of biomedical, scientific and clinical text. We also construct two normalization modules. (i) The coarse-grained normalization is applied to the answer only. It removes noise from the raw text (non-ASCII characters, HTML tags, duplicate spacing, etc.) (ii) The fine-grained normalization includes stop-words removing, lower-casing, stemming, and full form generation (Schwartz and Hearst, 2002) for biomedical abbreviations. Finally, BioBERT (Lee et al., 2020), which is designed for multiple biomedical text mining tasks, is used for part-of-speech tagging, named entities/keywords recognizing and embedding generating. BioBERT-based embeddings are $768-$ dimensional vectors used for calculating the similarity of words and sentences.

3.2.2 Single-answer extractive summarization

Using information from the pre-processing phase, the single-document extractive summarization phase generates the summary for every single answer. Our extractive summarization model tries to determine which sentences are important to the document by sentence scoring.

Sentences scoring: Since it is difficult to identify the importance of sentences from a single point of view, hence, we use three different types of scores: Frequency-based scores, graph-based scores and question-driven scores.

Frequency-based score: *Term Frequency - Inverse Document Frequency (TF-IDF)* (Salton and McGill, 1986) is the probabilistic method that reflects the importance of words in a set of documents by a float number. The TF-IDF score of a word w contained in document d of document set D is defined as $tfidf(w, d, D)$. We apply two rules to improve TF-IDF: (i) Boosting the TF-IDF score of keywords, and (ii) Assigning TF-IDF score to 0 if it is lower than a pre-selected threshold. The TF-IDF score of a sentence is the cumulative TF-IDF scores of its component words.

Graph-based scores are used to determine which sentences and words seem to be the core of a document. Lexrank and Textrank are two of the most well-known methods of this approach.

Lexrank (Erkan and Radev, 2004) computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. A document is considered as a graph, each node represents a sentence. Two nodes have a weighted edge depending on the similarity of their corresponding sentences. Cosine similarity is used to calculate the similarity between two sentences x and y (see Formula 1). In which, x and y are represented by TF-IDF vectors of n dimensions, i.e., X and Y respectively (n is the number of distinguished tokens in two sentences).

$$sim(x, y) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \quad (1)$$

To calculate the centrality of a node, we analyze the weight of its connected edges and the centrality of adjacent nodes (Formula 2). If a sentence is similar to many other sentences, it has higher centrality and conceived having a certain ability to represent other sentences.

$$p(u) = \frac{d}{n} + (1-d) \sum_{v \in adj_u} \frac{sim(u, v)}{\sum_{z \in adj_v} sim(z, v)} p(v) \quad (2)$$

where adj_u is the set of nodes that adjacent to u, n is the number of nodes and d is the damping factor.

Textrank (Mihalcea and Tarau, 2004) is mostly similar to Lexrank. It calculates the centrality of terms instead of the centrality of sentences as in Formula 3. In the PtN model, if the Textrank score is lower than a predefined threshold, we assign it to 0. The Textrank score of a sentence is the sum of Textrank scores of its participated terms.

$$sim(X, Y) = \frac{|w|w \in X \text{ and } w \in Y|}{log(|X|) + log(|Y|)} \quad (3)$$

in which w is the token and X and Y are two terms.

Question-driven scores are used to give higher priorities to sentences that are related to the questions. These scores are proposed to focus on the answer summarization task, ensuring that the summary is a suitable answer to the question.

Question-similarity score uses the BioBERT and Cosine distance (Formula 1) to calculate the similarities between the question and sentences in all of its answers. Formally, qb(sentence), the question-similarity score of a sentence is defined as:

$$qb(\text{sentence}) = sim(\text{sentence}, \text{question}) \quad (4)$$

Keyword-based score is determined by the percentage of question keywords that appear in a sentence. Let K is the set of question keywords, kw(sentence) is the keyword-based score of a sentence, it is defined by the following formula:

$$kw(\text{sentence}) = \frac{|\{k : k \in K\}|}{|K|} \quad (5)$$

Scores combination: All scores are normalized in the range $[0 - 1]$ by using `Min-Max` normalization. We then combine them into a final sentence score by using optimized weights (see Formula 6.

$$\begin{aligned} score =& w_1 \times tfidf \\ &+ w_2 \times lexrank + w_3 \times textrank \\ &+ w_4 \times querybase + w_5 \times keywords \end{aligned} \quad (6)$$

in which, w_i is the weight of each score. They are fine-tuned on the validation set.

Prosper-thy-neighbour strategies:
As described in Section 3.1, an important sentence may need some adjacent sentences to clarify or support it. Hence, answers often tend to have continuous segments of sentences that make important 'points'. Since the aforementioned scores do not consider the neighbours of a sentence, our prosper-thy-neighbour strategies are proposed to take advantage of this characteristic. There are three different prosper-thy-neighbour strategies: cluster-boosting, relative-boosting and centre-boosting.

Cluster-boosting: We calculate the averaged scores of n continuous sentences ($n = 3, 4, 5$) as cluster scores. We then select top-k clusters with the highest average scores. The sentence score is set equal to its highest cluster score. Sentences that are not selected in any clusters are assigned the score of 0.

Relative-boosting is performed by three steps:

- Step 1: Find top-n highest-score sentences with their original orders.
- Step 2: For consecutive selected sentences, let L is the position of the preceding sentence, R is the position of the following sentence. If $R - L + 1 \leq k$ (k is predefined), step 3 is executed.
- Step 3: Let $score_i$ be the score of the i-th sentence. The final scores $final_i$ of all sentences having the position between L and R are updated by the following formula:

$$final_i = max_{j=L}^{R}(score_j) \quad (7)$$

Centre-boosting: Let $score_i$ be the score of i-th sentences. The final score $final_i$ of sentence i-th is updated by the following formula:

$$final_i = max_{j=max(i-L+1,1)}^{min(i+R-1,n)} score_j \quad (8)$$

in which, n is the number of sentences, L and R is the number of sentences that impact the current sentence i in two directions: left and right. With centre-boosting, the important sentence strongly affects its adjacent sentences.

However, with these prosper-thy-neighbour strategies, the selected neighbour sentences can bring redundant information, i.e., we may keep too many sentences to the left/right of an important sentence. Those redundancies can be cut off in the post-processing phase (Section 3.2.4).

Ranking and and Filtering Sentences We use the final score boosted by the prosper-thy-neighbour strategy to rank the sentences. There are several ways to choose sentences for the single-document extractive summary: getting top-n or top-$p\%$ of sentences, using the threshold to filter unimportant sentences. In the proportion- and

threshold-based approach, the number of sentences depends on the document length and sentence scoring. It might probably cause an unexpected bias in the next multi-document summarization phase. Based on the experimental results on the validation set, we fix the number of selected sentences in each document.

3.2.3 Multi-answer extractive summarization

Multiple extractive single-answer summaries from the previous phase are merged into a single document. Since the previous phase chooses an equal number of sentences for all answers, there might be some redundant sentences. Since the current sentence scores are based on separate documents, we re-calculate them as in the merged document by using the proposed score described in Section 3.2.2. The filtering step then removes some lowest-score sentences.

Maximal Marginal Relevance (MMR): (Carbonell and Goldstein, 1998) is also used to reduce redundancy while maintaining query relevance. MMR works in the selected appropriate sentence in merged documents. It is the combination of the relevance and diversity concepts, in a controllable way. Let S_i is the i-th sentence, its MMR score is calculated based on the similarities between S_i, the answer D and the question Q (Formula 9). The similarity to the question and the duplication with other sentences affects the MMR score through the ratio λ. In which, BioBERT is used to represent sentences and question and Cosine distance is used to calculate the similarities. We use the MMR score to discard duplicated and question-irrelevant sentences, i.e., remove m sentences having the lowest MMR score.

$$\text{MMR}_i = \arg\max_{S_i \in D} [\lambda(sim(S_i, Q) - (1-\lambda) max_{j \neq i} sim(S_i, S_j))] \quad (9)$$

3.2.4 Post-processing

For each segment of continuously selected sentences, we find the position of the most important sentence which has the highest combined score. Then, for other sentences in the segment, if the distance from their position to the important sentence exceeds a predefined k parameter, those should be eliminated in the final multi-document extractive summary.

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 Formulas 10, 11 and 14, respectively. Choosing correct sentences help to increase ROUGE-n R and P.

$$\text{ROUGE-}n\ P = \frac{|\text{Matched N-grams}|}{|\text{Predict summary N-grams}|} \quad (10)$$

$$\text{ROUGE-}n\ R = \frac{|\text{Matched N-grams}|}{|\text{Reference summary N-grams}|} \quad (11)$$

$$\text{ROUGE-L}\ P = \frac{\text{Length of the LCS}}{|\text{Predict summary tokens}|} \quad (12)$$

$$\text{ROUGE-L}\ R = \frac{\text{Length of the LCS}}{|\text{Reference summary tokens}|} \quad (13)$$

ROUGE-L recall (R), precision (P) and $F1$ are calculated as in Formula 12, 13 and 14, respectively. ROUGE-L uses the Longest Common Subsequence (LCS) between predicted summary and referenced summary and they are normalized by the tokens in the summary.

$$F1 = 2 \times \frac{R \times R}{P + R} \quad (14)$$

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 detailed evaluation of the effectiveness of the single-answer summarization phase, we also make some comparisons with related works:

- Lead-3: First three sentences of an article were taken as a summary.
- k-random sentences: k random sentences were selected as a summary.
- k-best ROUGE: k sentences with the highest ROUGE-L score relative to the question were selected.

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

Team	ROUGE-1			ROUGE-2			ROUGE-L	HOLMS	BERTscore
	P	R	F1	P	R	F1	F1		F1
paht_nlp	0.471	**0.878**	0.585	0.407	**0.767**	**0.508**	0.435	0.706	**0.804**
UETrice	**0.528**	0.814	**0.611**	**0.432**	0.680	0.504	**0.441**	0.738	0.796
XIaoHouZi	0.464	0.864	0.577	0.395	0.748	0.495	0.431	0.699	0.797
ChicHealth	0.474	0.842	0.578	0.398	0.718	0.489	0.426	0.703	0.792
I_have_no_flash	0.472	0.843	0.573	0.397	0.719	0.488	0.425	**0.745**	0.791

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

- Bidirectional long short-term memory (BiLSTM) network (Hochreiter and Schmidhuber, 1997): The most relevant sentences in an article were selected by a BiLSTM.
- Pointer-generator network (See et al., 2017): A hybrid sequence-to-sequence attention model which creates summaries with two approaches: copying text and create new text from the source documents.
- Bidirectional auto-regressive transformer (BART) (Savery et al., 2020): A transformer-based encoder-decoder model improved with a question-driven approach.

The results of these comparative models are taken from experimental results reported in Savery et al. (2020).

4.3 Task final results and comparison

Based on the validation set experiments, the number of selected sentences in single-answer summarization is 7 per answer. In the multi-answer summarization phase, the score-based filter selects top-20 sentences in the merged document, then MMR removes 5 lowest-score sentences. Therefore, our multi-answer document summaries have 15 sentences (or less, based on the length of the original answers). Post-processing with distance value $k = 3$ often removes 2-4 sentences. The final outputs often have $\sim$13 sentences. Since both cluster-boosting and relative-boosting show their drawbacks with the lower F1-score performance on the validation set, we use the centre-boosting strategy in our optimal model.

4.3.1 Official results of the multi-answer extractive summarization

Table 2 shows the shared task official results of top-5 competitors. ROUGE-2 $F1$ is used as the main metric to rank the participating teams. We also show several other evaluation metrics for detailed results: ROUGE-1 $F1$, ROUGE-L $F1$, HOMLS $F1$ and BERT-based $F1$. We are the runner-up in the leader board, with ROUGE-2 $F1$ at 0.504 (0.004 less than the rank No.1 team). However, our ROUGE-1 $F1$ and ROUGE-L $F1$ are the highest of all participating teams.

Table 3: The comparative results of single-document summarization models.

Model	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1
Lead-3	0.23	0.11	0.08
3-random sentences	0.20	0.08	0.06
3-best ROUGE	0.16	0.08	0.06
BiLSTM	0.22	0.10	0.08
Pointer-generator	0.21	0.09	0.07
BART	0.24	0.10	0.07
BART + Query-based	0.29	0.15	0.12
PtN model w/o post-processing	0.26	0.22	0.24
PtN model	**0.30**	**0.22**	**0.25**

All results are reported on the training data set.
The highest results in each column are highlighted in bold.

4.3.2 Result of the single-answer extractive summarization

Table 3 shows the performances of our model and comparative models at the single-answer level. Because the results of the comparative models are reported in the training dataset, all results are reported on the training dataset. To ensure the comparisons are fair, we report both model results with and without the post-processing phase. The results show that our model outperforms all comparative models. To ensure the comparisons are fair, we report both model results with and without the post-processing phase. The results show that our model outperforms all comparative models.

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 afterward evaluating the model on the validation set. Validation data are used for evaluation because we use validation data to optimize the model's hyperparameters. 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, TF-IDF and MMR have the biggest contribution while Lexrank/Textrank brings the smallest contribution. The prosper-thy-neighbour strategy also demonstrates its effectiveness to improve the ROUGE-2 $F1$. Centre-boosting seems to be the most suitable strategy for this task since the results increase dramatically if we replace it with cluster-boosting or relative-boosting.

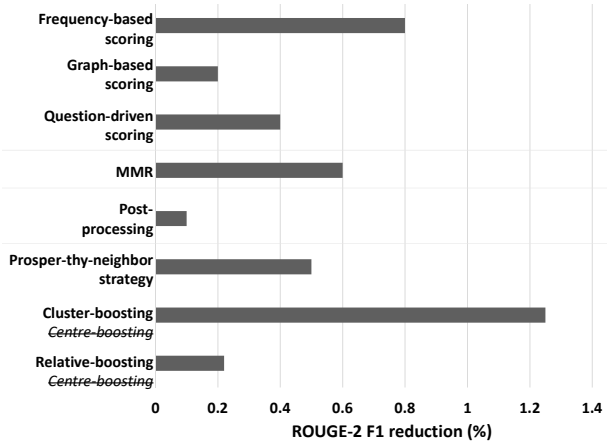

Figure 2: Ablation test results on validation data set for various components and Prosper-thy-neighbour strategies. Cluster-boosting and relative-boosting: Replace centre-boosting by another strategy.

We also investigate the change of results at different compression ratios. Figure 3 shows the change of ROUGE-2 P, R and $F1$ on the validation set when taking 2-20 sentences to the summary (excluding the post-processing step). We observed that P and F have trade-off results while increasing the number of sentences. $F1$ got the best results at 15 sentences, due to the balance between P and F. Therefore, we choose this configuration for our official runs on the test set.

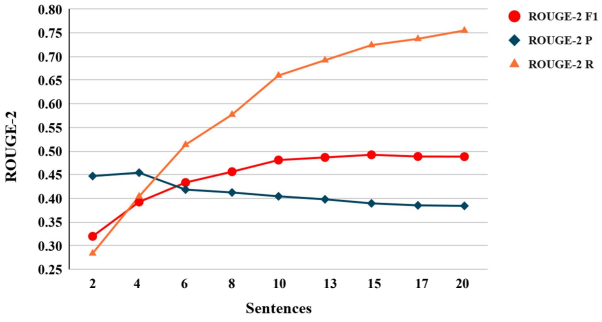

Figure 3: System performance with different compressed ratios.

4.5 Errors analysis

To further evaluate the performance of the proposed system, we have analyzed the results of the best model on the validation set. Table 4 provides some examples of the model problems and their effects.

Firstly, because of using a fixed statistical-based maximum number of output sentences, we ran into problems with too long or too short documents. Question #56 is an example of the redundancy in the output summary that there are only 5 important sentences but our model keeps fixed 13 sentences. On the contrary, in Question #91, the answer to '*How can I stop being allergic to caffeine?*' are summarized in 23 sentences. However, many relevant sentences have been filtered out to ensure a fixed size of the output.

Although we have combined many different ranking methods for tokens and sentences, some final scores did not meet our expectation. The frequency-based scores (TF-IDF) are failed in Question 82, in which the token *'Hirschsprung'* is over-weighted due to repeated occurrence. In addition, the popular keywords like *'treatment'*, *'medicine'* have too low weight. As a result, in Question #19 about *'the cure for pulsatile tinnitus'*, all of the sentences related to treatment and medicine were filtered out.

Some other issues related to the driven question are illustrated in Question #22 and Question #36. In the first example, the question analyzer failed to extract the keyword *'safe'*. For this reason, the summary phase went in the wrong direction – the content is only related to *'defibrillator'*. In the second one, the proposed model did not focus on the driven question so that the summary does not contain the desired information.

Besides the problems related to the model components, we also noticed some problems related to

Table 4: Examples of some errors in validation set.

#	Question	Problems	Effect
56	How can we improve fertility in Klinefelter syndrome karyotype 47 XXY?	Fixed number of output sentences	Redundant output sentences **(low precision)**
91	How can I stop being allergic to caffeine?	Fixed number of output sentences	Missing output sentences **(low recall)**
82	Where can i find information for adults with Hirschsprung's disease?	Imperfect ranking scores	Ranking of irrelevant sentences are too high **(low precision)**
19	Is there a cure for pulsatile tinnitus?	Imperfect ranking scores	Ranking of important sentences are low **(low recall)**
22	Is it safe to have ultrasound with a defibrillator?	Missing keywords and NER	Summary is on the wrong direction **(poor precision and recall)**
53	Is there a way to improve kidneys in a person on twice-weekly dialysis?	Not focus on driven-question	Summary is not contain the desired information **(poor precision and recall)**
36	Are there herbal medicines for rheumatoid arthritis?	Problem in chiQA answers	Not enough information to summarize
78	Can spinal surgery cause hydrocephalus and blindness in adults?	Problem in neighbour boosting	Adding some irrelevant sentence **(decreasing precision)**
28	Can you help me find a clinic that specializes in treatment for atopic eczema?	Problem in post-processing	Removal of important sentence **(decreasing recall)**

the input data for which Question #36 is an example. The question is about *'herbal medicines for rheumatoid arthritis'* while the chiQA answers do not mention this topic. Therefore, our model as well as other machine learning models do not have enough linguistic information to summarize these documents.

Some other errors seem attributable to our model's limitations (Example #28 and #78). We listed here some highlight problems to prioritize future researches: (i) The neighbour boosting method needs to be improved to only increase the weight of related sentences instead of all neighbouring sentences; (ii) Post-processing rules need to be stricter to avoid eliminating important sentences.

5 Conclusions

This paper presents a systematic study of our extractive approach to the MEDIQA 2021 - Task 2: Multi-answer summarization. We combined and optimized several scoring criteria such as TF-IDF, Lexrank, Textrank, query-based, keywords-based and MMR scores. We also developed a strategy called Prosper-thy-neighbour to take advantage of adjacent sentences in the answers. The proposed model has a potential performance, being the runner-up of the shared task. Our best performance achieved a ROUGE-2 $F1$ is 0.504, comparable to that of the highest-ranked system with 0.507.

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. The error analysis was made to analyze the sources of the errors. The evidence pointed to some imperfection of the sentence selecting strategy, the ranking score combination and the question analyzer. Our proposed system is extensible in several ways: applying machine learning model, deeply question-analyzing, sentences clustering, etc. We will release our source code on the public repository to support the re-producibility of our work and facilitate other related studies.

Acknowledgement

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

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *International Journal of Ad-*

vanced Computer Science and Applications (ijacsa), 8(10).

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.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294.

G¨unes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Sepp Hochreiter and J¨urgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Eduard Hovy, Chin-Yew Lin, et al. 1999. Automated text summarization in summarist. *Advances in automatic text summarization*, 14:81–94.

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.

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

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*.

Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

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.

Ariel S Schwartz and Marti A Hearst. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pages 451–462. World Scientific.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.