A Hybrid Multi-answer Summarization Model for the Biomedical Question-Answering System

Quoc-An Nguyen[1], **Quoc-Hung Duong**[1], **Minh-Quang Nguyen**[1], **Huy-Son Nguyen**[1]
Hoang-Quynh Le[1], **Duy-Cat Can**[1], **Tam Doan Thanh**[1,2] **and Mai-Vu Tran**[1]
[1]VNU University of Engineering and Technology, Hanoi, Vietnam.
{18020106, 18020021, 19020405, 18021102}@vnu.edu.vn
{lhquynh, catcd, vutm}@vnu.edu.vn
[2]Viettel Data Governance Department, Viettel Group.
doanthanhtam283@gmail.com

2021 13th International Conference on Knowledge and Systems Engineering (KSE) | 978-1-6654-9975-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/KSE53942.2021.9648640

***Abstract*—In natural language processing problems, text summarization is a difficult problem and always attracts attention from the research community, especially working on biomedical text data which lacks supporting tools and techniques. In this scientific research report, we propose a multi-document summarization model for the responses in the biomedical question and answer system. Our model includes components which is a combination of many advanced techniques as well as some improved methods proposed by authors. We present research methods applied to two main approaches: an extractive summarization architecture based on multi scores and state-of-the-art techniques, presenting our novel prosper-thy-neighbor strategies to improve performance; EAHS model (Extractive-Abstractive hybrid model) based on a denoising auto-encoder for pre-training sequence-to-sequence models (BART). In which we propose a question-driven filtering phase to optimize the selection of the most useful information. Our propose model has achieved positive results with the best ROUGE-1/ROUGE-L scores being the runner-up by ROUGE-2 $F1$ score by extractive summarization results (over 24 participated teams in MEDIQA2021).**

***Index Terms*—ROUGE, biomedical text data, multi-document summarization, extractive summarization, hybrid summarization model, question-driven.**

I. Introduction

Textual data is an invaluable source of information and knowledge that needs to be effectively extracted and synthesized to be useful. Especially on the biomedical data, this copious data gives us meaningful value because of its high applicability in industry, society, science and important in supporting human health care. Nowadays biomedical documents are available with a massive amount through several developed search engines and question-answering systems. However, the returned results of these systems still contain a lot of redundant content which making users hard to quickly grasp the necessary medical knowledge. Therefore, we look forward to building a system which supports people to generate a shorter condensed form with important knowledge to save time and to be able to retrieve useful information.

Text summarization are split into two types based on approaching way: extractive summarization (filtering important sentences/phrases from the raw documents without any modification) and abstractive summarization (creating a generalized summary with advanced language generation/compression techniques).

Our work concentrates on synthesizing and compressing information from related documents of a medical question. We have combined the pros and cons of both approaches to propose a hybrid model that meets users need. For extractive summarization problem: Investigating, inheriting and upgrading a series of typical methods that have been successfully applied in related studies such as TF-IDF score, similarity based on edit distance, embedding, entity identification, maximum relevant boundary, lexrank, textrank etc. More, we propose the architecture of combining the existing methods and some new techniques to increase the model results such as neighbor weight spreads. For abstractive summarization problem: taking advantage of BART, a pre-trained model combining bidirectional and auto-regressive transformers [1]. Our architecture establishes pair filtering phases to sieve the more concise input for BART. The coarse-grained filtering phase removes question-irrelevant sentences because we need a question-oriented summary. Afterward the fine-grained filtering phase is applied to cut-off noise phrases.

This paper is organized as follows: Section II gives a general viewpoint about some state-of-the-art related works in the text summarization field. Section III is the detailed description of our proposed multi-answer summarization model. Section IV provides a specific and objective viewpoint about the effectiveness of our proposed model and comparisons with some related works.

II. Related works

So far, the research community has presented more on extractive summarization because this approach often achieve logical and significant results in a more straightforward way than abstractive summarization. Extractive summarization have been extensively researched and published since the mid-20th century [2]. In general, they proposed and used prominent weighting methods to select important sentences/phrases in the document. Term Frequency-Inverse Document Frequency (TF-IDF) [3] is a technique for text vectorization based on the Bag of words (BoW) model. Maximal Marginal Relevance (MMR)

978-1-6654-9975-0/21/$31.00 ©2021 IEEE

[4] is is an important ”diversity based ranking technique”, used to maximizes the relevance and novelty in finally retrieved top-ranked items. Lexrank [5] and Textrank [6] are two graph-based methods. Both of them use the PageRank algorithm for extracting top keywords. We also have conducted experiments and research based on the advantages and disadvantages of each weighting method to propose an architecture that takes advantage of some outstanding weighting methods to synthesize the final score.

Based on the development trend of deep learning models in other NLP problems, advanced techniques have been born and improved significant performance for abstractive summarization solution. With the application of neural network, Rush et al. (2015) [7] create each word of the summary conditioned on the input sentence by utilizing a local attention-based model. With Reinforce-Selected Sentence Rewriting, Chen et al. (2018) [8] utilize a novel sentence-level policy gradient method to connect the non-differentiable computation between these two neural networks in a hierarchical technique (still guaranteeing language fluidity). BART [1] is a transformer-based pretrained denoising encoder-decoder model which is a famous method that has been applied to solve a lot of NLP problems. It combines a bidirectional encoder and an auto-regressive decoder. There are many featured BART-based model which can be mentioned Question-driven BART [9] which is re-trained BART to enhance its ability to perceive content (including document rotation, sentence permutation, text-infilling, token masking and token deletion). A recently popular SOTA model is PEGASUS [10] which masks/removed important sentences from an input document and generates one output sequence from the extant sentences, the same to an extractive summary.

III. Materials and Methods

In this section, we will propose our hybrid model (see Figure 1) taking advatange of both extractive and abstractive approaches. It includes four phases: (i) pre-processing, (ii) single-answer extractive summarization, (iii) multi-answer extractive summarization and (iv) abstractive summarization.

A. Pre-processing

In this phase, the inputs include a question Q and a set of answers $D = \{d_i\}_{i=1}^{n}$. We use SciSpaCy [11] as the main pre-processing tool for biomedical data segmentation and tokenization. Then, we also construct two normalization modules: (i) The coarse-grained normalization applies only to answers to clean the raw text by removing non-meaning components like non-ASCII characters, HTML tags, redundant spacing, etc. (ii) The fine-grained normalization applies to both the question and answers to normalize all words by removing stop-words, transforming into lowercase, stemming, and generating full form of abbreviation [12] . Furthermore, specially designed for text processing tasks in biomedical field, BioBERT [13] is used for part-of-speech tagging (POS tagging), named entities/keywords recognizing (NER). Multiple 768-dimensional vectors for each sentence are also generated to calculate the similarity between sentences.

B. Single-answer extractive summarization

With the data retrieving from the pre-processing phase, we try to generate the summarization for each answer first. The extractive summarization phase attempts to determine which sentences are important by using multiple scoring, ranking and filtering steps.

1) Sentences scoring: Because it is difficult to identify which sentences are important from a single aspect, we decide to use multiple scoring strategies, which are Frequency-based scores, graph-based scores, and question-driven scores.

a) Frequency-based score: : Term Frequency - Inverse Document Frequency (TF-IDF) [3] is the probabilistic method that reflects the importance of words by comparable numbers. Suppose that $tfidf(w, d, D)$ is the TF-IDF score of a word w in document d of a document set D. Besides, we apply two following rules to enhance the TF-IDF: (i) boosting the TF-IDF score of words that are in the keywords list generated from the pre-processing phase and (ii) assigning the score to 0 for sentences with the TF-IDF score lower than a predefined threshold. Then, the TF-IDF score of a sentence is the accumulated score of all its contained words.

b) Graph-based scores : are used to determine the “centrality” of sentences and words in the answer. Lexrank and Textrank are two of the most familiar methods.

Lexrank [5] calculates the importance of a sentence according to the theory of centroids of eigenvectors in its graph representation. A document is considered a graph, with each node representing a sentence. A weighted edge between two nodes is computed based on the cosine similarity of their corresponding sentences (see Formula 1).

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

In which, n is the number of distinct tokens of two sentences, x and y are represented by n-dimension TF-IDF vectors X and Y respectively.

In order to calculate the centrality of a node, we analyze the weight adjacent edges and the centrality of adjacent nodes. The more the adjacent edges' weight and nodes' centrality are, the higher probability the sentence can represent other sentences (see Formula 2).

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

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 [6] is mostly similar to Lexrank. Instead of calculating the centrality of sentences, it calculates the centrality of terms (see Formula 3). Moreover, we also assign the Textrank score to 0 if it is lower than a predefined threshold. Then, the Textrank score of a sentence is the sum of Textrank scores of its participated terms.

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

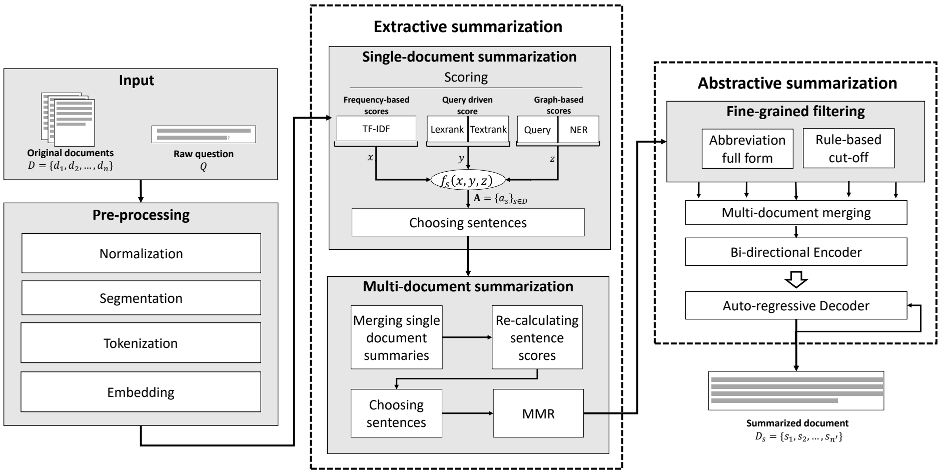

Fig. 1. The proposed hyprid multi-answer summarization model.

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

c) Question-driven scores : are used to give higher priorities to sentences that are related to the questions. These scores focus on the question-driven summarization task to ensure the predicted summary is high-related to the question.

Question-similarity score uses the BioBERT and Cosine distance (see Formula 1) to calculate the similarities between two represented vectors of the question and sentences. Formally, suppose that the question-similarity score of a sentence is $\mathrm{qb}(sentence)$, it is defined by the following formula:

$$\mathrm{qb}(sentence) = \mathrm{sim}(sentence, question) \tag{4}$$

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

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

2) Scores combining: After calculating all the above scores, we use Min-Max normalization to scale all values into the range $[0 - 1]$. The final score is computed by combining weighted scores of each score components (see Formula 6).

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

in which, w_i is the weight of each score and is fine-tuned on the training data. In our model, we use the validation set (Section IV-B).

3) Ranking and filtering sentences: The combined final scores of all sentences in an answer are sorted in decreasing order. After that, two strategies are proposed to extract the single-answer summary: getting top-n or top-p% of sentences.

With a proportion- and threshold-based approach, the number of sentences depends on the length of the document and the sentence score, unexpected bias can occur during the following multiple document overview stages. According to lots of experimental results on validation set, the number of sentences per single-answer summary is fixed to achieve the best effectiveness. The single-answer summarization is all selected sentences reordered by their original positions.

C. Multi-answer extractive summarization

Multiple extractive single-answer summaries from the previous phase are merged into a single document. In the previous step, there may be some duplicate sentences because of selecting the fixed number of sentences for all answers. We also need to re-calculate scores of sentences in the merged document (using the proposed score described in Section III-B), reranking and filtering to cuts off low-score sentences.

In order to remove duplicate sentences, we decide to use *Maximal Marginal Relevance* [4]. It removes duplicate sentences while maintaining the answer's relevance by using the combination of the relevance and diversity concepts in a controllable way. Precisely, 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 (see Formula 7).

$$\begin{aligned} \mathrm{MMR}_i = & \arg\max_{S_i \in D} [\lambda(sim(S_i, Q) \\ & - (1 - \lambda) max_{j \neq i} sim(S_i, S_j))] \end{aligned} \tag{7}$$

In which, the ratio λ decide how different the similarity to the question and the duplication with other sentences affects the MMR score. The similarity is calculated is the cosine distance between represented vectors of sentences using BioBERT.

After re-ranking and filtering, MMR scores support to discard duplicated and question-irrelevant sentences. Finally, the multi-answer extractive summarization is created by reordered by the original positions of the remaining sentences.

D. Abstractive summarization

After producing the single- and multi-answer extractive summarization, we continue to process that outputs to t-e abstractive summarization phase which is based on BART [14] - the recently state-of-the-art denoising sequence-to-sequence model. Depending on some special characteristics of the abstractive summarization as well as the limited length input of BART (maximum 1024 tokens are allowed), we propose two additional filtering phases to make the more condensed question-driven input for BART: (i) coarse-grained filtering and (ii) fine-grained filtering.

a) Coarse-grained filtering: The original BART model can generate more concise text, but there is no priority for important information, like a question. Therefore, we help BART to be question-driven by giving the question-driven input, in a phase we call filtering. There are two strategies to choose such sentences as the BART input:

- (i) Top-n query-driven sentences: choosing sentences that are similar to the questions by calculating the cosine similarity on two BioBERT embedding vectors. The higher cosine similarity is assumed to be more related to the question.
- (ii) Reuse the extractive summarization: this is the preferred way to use the extract summary output from the previous phase. Because of the high extraction summary recall results, we will not lose much important information when using it as the input.

b) Fine-grained filtering: In this step, we need to (i) filter out noise phrases, using rules to cut off redundant elements of sentences, and (ii) generate the full-form of all local biomedical abbreviations containing in the answers by using the Ab3P tool [15].

All selected sentences from two filtering phases are then combined into a single document which is the input of the BART model.

c) BART-based summary generation: BART [14] is implemented as a standard sequence-to-sequence Transformer-based model to translate string to string with variant lengths. It consists of two components: Encoder and Decoder that take advantage of both BERT and GPT.

Encoder: BART uses a bidirectional encoder over corrupted text taken from BERT [16]. As the strength of BERT lies in capturing two-dimensional contexts, BART can encode the input string (the sequence of tokens) in both directions and get more context information. Suppose that the i-th word is represented by the word vector $\mathbf{x}_t$, the $\mathbf{h}_t$ hidden states are calculated with the formula:

$$\mathbf{h}_t = f(\mathbf{W}^{hh} \cdot \mathbf{h}_{t-1} + \mathbf{W}^{hx} \cdot \mathbf{x}_t) \tag{8}$$

in which, the hidden states are computed by the input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$; $\mathbf{W}^{hx}$ and $\mathbf{W}^{hh}$ are weights' matrices used to for $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$, respectively. The output encoder vector is used as an input for the next decoder step.

Decoder: BART uses a left-to-right auto-regressive decoder which is similar to GPT2[1], with additional autoregressive capabilities that can be used to reconstruct input noise. In each step, we take the previous hidden state and the element in the output word vector from the encoder step to produce the new hidden state. The output sequence is the set of words of the summarized answer, in which i-th word is represented by y_i. The hidden state $\mathbf{h}_i$ is calculated from the previous state by the following formula:

$$\mathbf{h}_t = f(\mathbf{W}^{hh} \cdot \mathbf{h}_{t-1}) \tag{9}$$

We compute the output using the latency at present and multiply it by the corresponding weight $\mathbf{W}^S$. The $\mathrm{softmax}$ function create a probability vector that helps us to determine the final output and Beam Search algorithm is used to find the output. The final output y_t are calculated by the formula:

$$\mathbf{y}_t = \mathrm{softmax}(\mathbf{W}^S \cdot \mathbf{h}_t) \tag{10}$$

IV. Experiences and Comparisons

A. Evaluation metrics

ROUGE score [17], is the most well-known metric to measure the effectiveness of the summary by comparing it with the ideal one. ROUGE-n, including Recall (R), Precision (P) and $F1$, between predicted summary and ideal summary are calculated as in Formular 11, 12 and 15, respectively. the more correct sentences are selected, the higher ROUGE-n R and P increase.

$$\text{ROUGE-}n\ P = \frac{|\text{Matched N-grams}|}{|\text{Predict summary N-grams}|} \tag{11}$$

$$\text{ROUGE-}n\ R = \frac{|\text{Matched N-grams}|}{|\text{Reference summary N-grams}|} \tag{12}$$

ROUGE-L recall (R), precision (P) and $F1$ are calculated as in Formular 13, 14 and 15, respectively. ROUGE-L is calculated by using the Longest Common Subsequence (LCS) between predicted summary and ideal summary.

$$\text{ROUGE-L}\ P = \frac{\text{Length of the LCS}}{|\text{Predict summary tokens}|} \tag{13}$$

$$\text{ROUGE-L}\ R = \frac{\text{Length of the LCS}}{|\text{Reference summary tokens}|} \tag{14}$$

$$F1 = 2 \times \frac{\mathrm{R} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{15}$$

B. Datasets

We use the MEDIQA-AnS Dataset (training set) [9], the validation and test set of the MEDIQA shared task as the official datasets for fine-tuning and evaluating our model. The MEDIQA-AnS Dataset data includes 156 questions with related documents as the answers for each. Each answer also has an extractive and an abstractive single-answer summaries, an extractive and an abstractive multi-doc summary for each

[1] https://openai.com/blog/tags/gpt-2/

TABLE I
THE COMPARATIVE RESULTS OF EXTRACTIVE 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
Our SaExS model	**0.30**	**0.22**	**0.25**

question. The 50-question validation set shares the same format with the training set without singe-doc summaries. The public part of test set contains 80 questions and these related documents. All summaries are created manually by medical experts.

C. Results

1) Contest results: In the MEDIQA shared task, our multi-doc summaries generated at our Multi-answer extractive summarization (MaExS, Section III-C) phase achieved the runner-up [18] in the leader boards with ROUGE-2 $F1$ at 0.504 (0.004 less than the rank No.1 team) in extract summarization -evaluation. In abstract summarization-evaluation, our single-abstractive summarization model (SAbS model) [19], which uses SaExS & MaExS (extractive summarization) as Coarse-grained filtering phase, holds a high rank of both rankings which evaluate the abstractive summaries on the extractive and abstractive references.

2) Comparative models: As our Single-answer extractive summarization (**SaExS**, Section III-B) phase can be seen as an independent model, we also make some comparisons with some reported related works on single-doc summarizing ability [9] (Table I).

- Lead-3: A summary contains the first three sentences.
- k-random sentences: A summary contains k sentences selected randomly.
- k-best ROUGE: A summary contains k sentences with the highest ROUGE-L score relative to the question.
- Bidirectional long short-term memory (BiLSTM) network: A summary contains k relevant sentences selected by the biLSTM model.
- Pointer-generator network: A hybrid sequence-to-sequence attention model was used to created summaries.
- Bidirectional auto-regressive transformer (BART): A transformer-based encoder-decoder model improved with a question-driven approach.

Table II shows the comparison between our proposed **SAbS model** and two other state-of-the-art text generation models, i.e., DistilBart and Pegasus, on the validation dataset of the MEDIQA contest. Our model using yields much better results than DistilBart and PEGASUS in this data.

TABLE II
COMPARISON WITH OTHER STATE-OF-THE-ART ABSTRACTIVE SUMMARIZATION MODELS.

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

3) Contribution of model components: We figure out how important every component is in each model by removing one in turn from the model architecture and then evaluating predicted multi-doc summaries. We compare these experimental results with the full system results by the changes of ROUGE-2 $F1$ of extractive summarization phase in Figure 2 and SAbS in Figure 3. These figures show the strong decrease 0.04% of $ROUGE\!-\!2F1$ score while removing Coarse-grained filtering phase using extractive summarization. Therefore, Extractive summarization play an important role in improving the effectiveness of SAbS model.

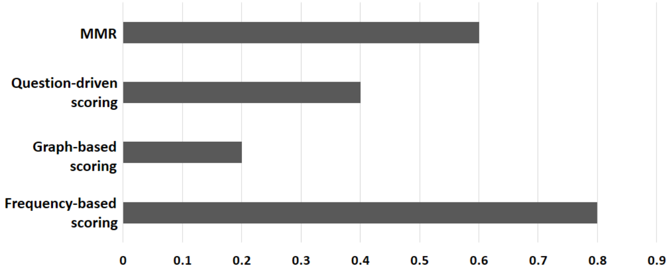

Fig. 2. Ablation test results for SAbS model components.

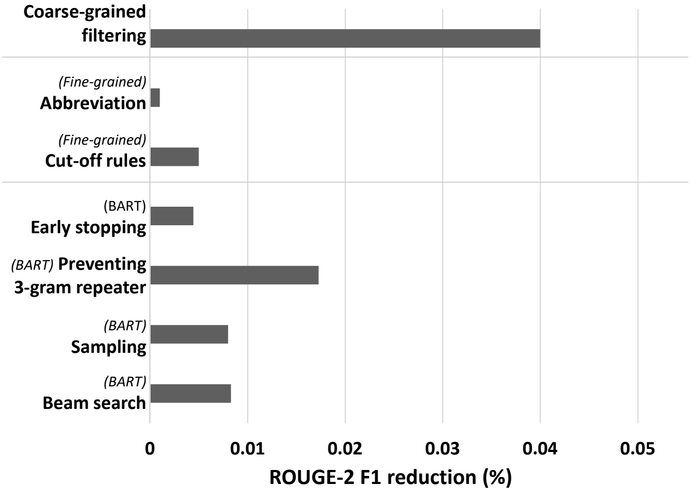

Fig. 3. Ablation test results for abstractive summarization phase components.

Conclusions

This paper reports the research of a hybrid model and a pipeline architecture to summary multi-answer in the medical question and answer system. A hybrid model was improved and optimized based on one state-of-the-art method - BART. Especially, taking advantage of multiple-scoring-based extractive summaries as the input for BART in EAHS model has proved significant performance. The best performance achieved a ROUGE-2 $F1$ is 0.147 evaluated on abstractive summarization references, being the runner-up of the shared task.

The power of contribution and robustness of all techniques and hyper-parameters was tested by experiments. The evidence shows that the performance of EACH is higher than the SAbS model and the summarized texts are more coherent and logical.

Our proposed system is extensible in several approaches such as machine learning, deeply question-analyzing, sentence clustering, etc. We will public source code to support the reproducibility of our work and support other related researches.

Acknowledgement

We wish to express our thanks to the organizing committee and reviewers of KSE 2021, NAACL-BioNLP 2021 shared tasks.

References

[1] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, 2020, pp. 7871–7880.

[2] N. Moratanch and C. Gopalan, "A survey on extractive text summarization," 01 2017, pp. 1–6.

[3] H. Christian, M. P. Agus, and D. Suhartono, "Single document automatic text summarization using term frequency-inverse document frequency (tf-idf)," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 7, no. 4, pp. 285–294, 2016.

[4] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi, "Simple unsupervised keyphrase extraction using sentence embeddings," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 2018, pp. 221–229.

[5] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[6] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[7] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[8] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," *arXiv preprint arXiv:1805.11080*, 2018.

[9] M. Savery, A. B. Abacha, S. Gayen, and D. Demner-Fushman, "Question-driven summarization of answers to consumer health questions," *Scientific Data*, vol. 7, no. 1, pp. 1–9, 2020.

[10] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.

[11] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.

[12] A. S. Schwartz and M. A. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical text," in *Biocomputing 2003*. World Scientific, 2002, pp. 451–462.

[13] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[14] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, 2019.

[15] S. Sohn, D. C. Comeau, W. Kim, and W. J. Wilbur, "Abbreviation definition identification based on automatic precision estimates," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–10, 2008.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[17] C.-Y. Lin and F. Och, "Looking for a few good metrics: Rouge and its evaluation," in *Ntcir Workshop*, 2004.

[18] D.-C. Can, Q.-A. Nguyen, Q.-H. Duong, M.-Q. Nguyen, H.-S. Nguyen, L. N. T. Ngoc, Q. T. Ha, and M.-V. Tran, "Uetrice at mediqa 2021: A prosper-thy-neighbour extractive multi-document summarization model," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021, pp. 311–319.

[19] H.-Q. Le, Q.-A. Nguyen, Q.-H. Duong, M.-Q. Nguyen, H.-S. Nguyen, T. D. Thanh, H.-Y. T. Vuong, and T. M. Nguyen, "Uetfishes at mediqa 2021: Standing-on-the-shoulders-of-giants model for abstractive multi-answer summarization," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021.