damo_nlp at MEDIQA 2021: Knowledge-based Preprocessing and Coverage-oriented Reranking for Medical Question Summarization

Yifan He and **Mosha Chen** and **Songfang Huang**
Alibaba Group
{y.he, chenmosha.cms, songfang.hsf}@alibaba-inc.com

Abstract

Medical question summarization is an important but difficult task, where the input is often complex and erroneous while annotated data is expensive to acquire.

We report our participation in the MEDIQA 2021 question summarization task in which we are required to address these challenges. We start from pre-trained conditional generative language models, use knowledge bases to help correct input errors, and rerank single system outputs to boost coverage. Experimental results show significant improvement in string-based metrics.

1 Introduction

Question summarization for medical forum is important for medical knowledge discovery and retrieval and facilitates downstream tasks such as biomedical question answering (Jin et al., 2021). Medical questions are often complex, scattered with non-medical information, and can sometimes be erroneous because forum users are not domain experts (Ben Abacha and Demner-Fushman, 2019). In addition, annotation in the medical domain is harder to acquire than in the general domain. These challenges make medical question summarization an important and difficult task where annotation is often scarce.

The MEDIQA 2021 shared task 1 (Ben Abacha et al., 2021), medical question summarization, requires participants to build summarization systems for noisy medical forum texts with limited annotation data. The official training set of the task is the MeQSum dataset (Ben Abacha and Demner-Fushman, 2019), which is composed of 1,000 medical questions and their corresponding summaries. The validation and test sets consist of 50 and 100 questions respectively and topic words are sometimes misspelled.

Scarcity of data, noisy input, and complexity and redundancy of text all pose challenges for question summarization systems. We try to address these challenges using a combination of knowledge-based error correction, pre-trained generative language models, and output reranking.

Knowledge-based error correction leverages multiple levels of lexical resources and a high coverage knowledge base to correct errors in input. Our experiments show that knowledge-based error correction helps downstream summarization performance according to the Rouge metric.

Pre-trained generative language models are transformer-based language models trained with loss functions that facilitate sequence to sequence generation. Models such as Pegasus (Zhang et al., 2020a), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020) achieve state-of-the-art performance on various text generation tasks and are shown to perform well on few-shot generation scenarios (Goodwin et al., 2020). We finetune pre-trained language models to obtain baseline systems with limited amount of training data.

Output reranking picks the best output among multiple systems. The availability of different language models offers a diverse set of summaries to choose from. We observe difference in summarization styles between the training and the validation set and devise a simple heuristic to pick the best output based on this observation.

In the rest of the paper, we describe these components and report evaluation results on the validation and the test set.

2 Task and Architecture Overview

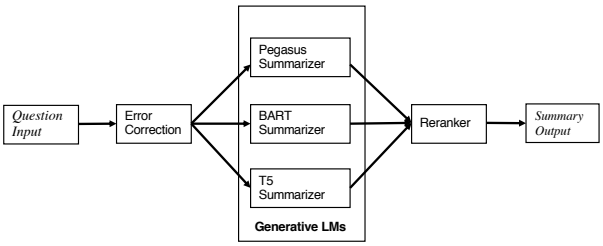
The MEDIQA question summarization task requires participants to summarize user generated medical queries into shorter, more focused questions. We present an example from the MEDIQA 2021 task 1 validation set in Figure 1 (a). We note that the name of the disease "folliculitis" is spelled

Proceedings of the BioNLP 2021 workshop, pages 112–118

Question Hi, Please can you help - I am writing from South Africa. My daughter suffers with acute follicilitus, and has been since the age of 13. She is now 20 and is in so much distress as nothing seems to alleviate the itching and soreness... I am writing to you for any help you could give me to try and assist her. Could you recommend a specialist and someone who could help us with research? Please could you point us in the right direction? I am happy to send through her lab tests - please let me know. Thanks

Summary How can we find a specialist or clinical trial for chronic follicilitis?

(a) Example from MEDIQA 2021 Task 1 Validation Set



(b) Architecture of our submission

Figure 1: Question-summarization example and system architecture

incorrectly in the input question and the question contains a lot of irrelevant information. We attempt to correct misspellings with a dedicated module in our system. As useful information is often scattered in different sentences in the input, abstractive summarization suits this task better than extractive summarization. We perform abstractive summarization with pre-trained language models.

We illustrate the architecture of our submission in Figure 1 (b): we first try to correct spell errors in the input; then summarize each question with three generative LMs: Pegasus, BART, and T5; finally, for each question, we pick the best output with a feature-based reranker and the best output is chosen as the summarization of the question.

3 Knowledge-based Error Correction

Misspellings are prevalent in medical forums, where non-expert users discuss highly specialized medical topics. Uncorrected misspellings can lead to mismatch between the source text and the summary during training and cause errors if copied verbatim during prediction. These errors are penalized heavily by string matching-based metrics like Rouge as they break n-grams.

In this shared task, we conservatively correct misspelled words in input by reusing a cascade of candidate generation modules from an entity linking system. Entity linking is the task to link entity mentions in text to entities in a knowledge base (KB). Candidate generation is an intermediate step in entity linking to generate candidate KB entities from potentially abbreviated, misspelled, or alias text mentions (see e.g. (Charton et al., 2014)). Our method is also comparable to previous work on Levenshtein distance-based (Levenshtein, 1966) medical query correction (Soualmia et al., 2012), but we augment that approach with cascaded knowledge sources and an alias table.

Error correction can be implemented easier and with possibly higher quality if search suggestions from online search engines (Zhou et al., 2015) are utilized. We use in-house error correction to keep the submission offline.

3.1 Resources

The error correction module relies on the following resources:

- **Medical word list**. We collect tokens from the English side of ~20K bilingual medical phrases collected from dictionaries and drug names.
- **Wikipedia dump**. We use a 20210101 dump of the English Wikipedia as the knowledge base and alias table.
- **High frequency word list**. We use the top 10,000 words in the Google 1T corpus [1].

We use Wikipedia instead of a medical KB because of its broad coverage. Edges (redirects, links etc.) in the Wikipedia KB can be used as an alias table to capture common misspellings and aliases.

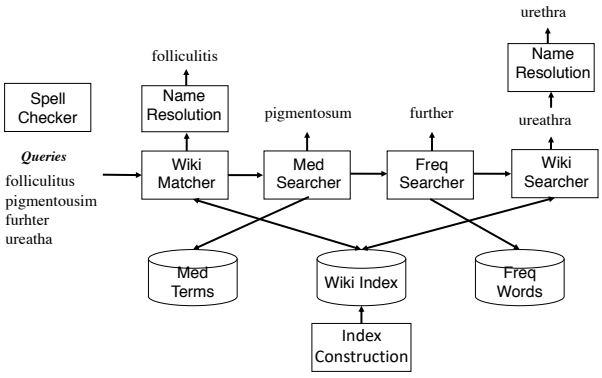


Figure 2: Example of error correction

3.2 Error correction steps

During error correction, we handle tokens composed entirely of alphabetical characters and allow at most 2 edits in similarity searches. We only consider tokens that share 3-prefix or 3-suffix with the query to limit search space.

[1] https://books.google.com/ngrams/info

Error correction consists of the following steps:

- **Index construction**. We build a token index of Wikipedia. We only index titles with no more than two tokens and tokens more than 5 characters long. We use the first token to represent the title. When a token can map to more than one titles, we map it to the title with the lowest id.
- **Spell checking**. We pass the text through a spell checker with medical terms[2] to detect potential errors. The flagged tokens are the *query* words for the error correction pipeline.
- **Wikipedia match**. If the query has an exact match in the Wikipedia token index, we link the query to the token and its corresponding Wikipedia title. Note that a title can either be an entity or an alias, which we resolve later in the name resolution step.
- **Medical word search**. We search the medical word list to find medical terms that spell similarly to the query. We choose the medical term if a result is found.
- **Frequent word search**. We search the high frequency word list to recall common English words that spell similarly to the query. We choose the word if a result is found.
- **Wikipedia search**. We search the Wikipedia token index for queries longer than 5. To further constrain search space, we only consider tokens that share 5-prefix, 5-suffix, or all consonants with the query. We choose the token with the highest sequence matching ratio[3].
- **Name resolution**. For corrected tokens retrieved from the medical word list and the Wikipedia, we search the Wikipedia dump to check if it is an alias of another entity and maps it to its canonical form.

Consider the example in Figure 2. Input queries of the error correction pipeline are the misspelled words identified by the spell checker. Wikipedia match catches the common misspelling **folliculitus* and recovers its canonical form *folliculitis*. Medical word search recovers *pigmentosum* from the medical dictionary. Frequent word search recovers misspellings of popular words, avoid sending them to the noisy Wikipedia search. Finally, Wikipedia search first map **ureatha* to its closest alias *ureathra* in Wikipedia and then maps *ureathra* to the canonical form *urethra*.

On the validation set, the process is unable to recover the word **preagnet* (pregnant). We are able to recover most other errors on the validation set. Impact of error correction is evaluated in Section 6.2.1.

4 Summarization with Pre-trained Conditional Generative Language Models

Pre-trained conditional generative language models have become the dominating paradigm for text generation and especially summarization, with recent models such as Pegasus (Zhang et al., 2020a), BART (Lewis et al., 2020), T5 (Raffel et al., 2020), and PALM (Bi et al., 2020) achieving state-of-the-art results on standard benchmarks CNN-Dailymail (See et al., 2017) and XSUM (Narayan et al., 2018). Recent work has also shown that these models achieve good performance in few-shot medical summarization settings (Goodwin et al., 2020).

Following (Goodwin et al., 2020), we use Pegasus, BART, and T5 single systems as our baselines.

- **Pegasus** (Zhang et al., 2020a) is a conditional language model designed specifically for abstractive summarization and is pre-trained with a self-supervised gap-sentence-generation objective, where the model is pre-trained to predict entire masked sentences from the document.
- **BART** (Lewis et al., 2020) is a model combining bi-directional and auto-regressive transformers, trained to both denoise and reconstruct corrupted texts.
- **T5** (Raffel et al., 2020) is pre-trained on multiple objectives, including masking, translation, classification, machine reading comprehension (MRC) and summarization, all formulated as conditional generation tasks.

We use `Pegasus-large`, `BART-large`, and `T5-base` respectively in our experiments.

[2] https://github.com/glutanimate/hunspell-en-med-glut

[3] https://docs.python.org/3/library/difflib.html

5 Output Reranking

Following previous work on reranking generative LM outputs (Mi et al., 2021), we pick the best summary for each question using the following linear model from outputs of three heterogeneous generative LMs,

$$T^* = \underset{T'}{\operatorname{argmax}} \sum_i \psi_i(\mathbf{T}, T', S) w_i \quad (1)$$

where T' is output of a single system, $\mathbf{T}$ is the set of outputs of all single systems, and S is the input text. T^* is the ensemble output, which is picked from single system outputs by highest score.

The feature function $\psi(\mathbf{T}, T', S)$ is a function to estimate the quality of T' using information from $\mathbf{T}$ and S. w_i is a weight of $\psi(\mathbf{T}, T', S)$. In sequence generation tasks such as machine translation (Kumar and Byrne, 2004), ψ is usually a combination of consensus and linguistic features and w_i can be tuned by optimization algorithms such as MERT (Och, 2003) towards an automatic evaluation metric.

Our approach. We use a simple and coverage-oriented approach for reranking, based on the size and characteristics of the validation data. We notice that the writing style of the validation set is different from the MeQSum data set which we use for training: in MeQSum 18.5% sentences start with "*What are the treatments for*", 14.6% start with "*Where can I find*", and 2.5% start with "*What are the causes of*". A model trained on MeQSum tends to generate these topic-based boilerplates that are not mentioned in the source text. But in the validation set, summaries do not have these boilerplate texts and resemble the content of the source text more closely, which inspires us to pick the output with high coverage of the source.

We consider the validation set (50 sentences) too small for automatic tuning, so we design a minimal set of features and set the weights w_i manually.

Features. We use fidelity, length, consensus and wellformedness features:

- **Fidelity** (w_f). We calculate the Rouge-2 score between the input and the prediction. A higher score indicate a high-coverage summary.
- **Length** (w_l). The length ratio between the prediction and the input.

	Rouge-2	Rouge-L
Pegasus	0.187	0.333
Pegasus EC	0.206	0.344
BART	0.220	0.342
BART EC	0.227	0.342
T5	0.213	0.353
T5 EC	0.208	0.354

Table 1: Single system results on validation set. EC: Input error correction

	Rouge-2	Rouge-L
Best Single	0.220	0.342
Reranked	0.217	0.361
Best Single EC	0.227	0.342
Reranked EC	0.230	0.364

Table 2: Reranking results on validation set. EC: Input error correction

- **Consensus** (w_c). 1 if T' shares any bigram with $\mathbf{T} - T'$, 0 otherwise.
- **Wellformedness** (w_w). 1 if T' has less than three subsentences and starts with one question marker, 0 otherwise.

For our experiments on the validation set and Rouge-2 experiments on the test set, we set $w_f = 1$, $w_l = 0.01$, $w_c = 10$, $w_w = 10$. The idea is to select the summary that has highest coverage of the source that is a one sentence question, with at least one bi-gram in common with other summaries.

The choice to favor high coverage summary is based on this particular pair of training and validation data, rather than general ensemble principles for text generation. We switch the weights for w_f and w_l for length reranking experiments on the test set. Impact of reranking is evaluated in Section 6.2.2.

6 Experiments

6.1 Experimental settings

Our systems are based on the Transformers (Wolf et al., 2020) package. We finetune baseline models on the MeQSum (Ben Abacha and Demner-Fushman, 2019) dataset for 50 epochs, with batch size 8 and learning rate 2e-5 with the AdamW optimizer on Nvidia P100 GPUs. Finetuning is indispensable for this task: without finetuning, `BART-large` scores 0.06 Rouge-2 and 0.15

	ID	R1	R2 P	R2 R	R2 F1	R-L	HOLMS	BERTScore
Single Systems								
1	T5	0.296	0.122	0.109	0.107	0.267	0.541	0.673
2	BART	0.286	0.120	0.090	0.098	0.258	0.550	0.667
3	Pegasus	0.312	0.130	0.123	0.118	0.281	0.547	0.684
Length rerank								
4	3 Sys	0.342	0.149	0.166	0.148	0.299	0.561	0.689
5	3 Sys EC	0.351	0.157	0.175	0.155	0.307	0.566	0.688
6	4 Sys EC	0.358	0.160	0.181	0.159	0.310	0.565	0.689
Coverage rerank								
7	3' Sys EC	0.350	0.177	0.169	0.161	0.313	0.571	0.691
8	4 Sys EC	**0.351**	0.173	**0.173**	**0.161**	0.313	0.568	0.689
-	Best team	0.351	0.185	0.173	0.161	0.315	0.579	0.703

Table 3: Results on the test set. EC: Input error correction; R1/2/L: Rouge-1/2/L; P: Precision, R: Recall; Best team: Best score among all teams; Scores in bold when our system achieves the best score.

Rouge-L on the validation set in preliminary experiments.

For experiments on the test set, models for ensemble are further finetuned for 50 epochs on the validation set. Models for error-corrected input are finetuned on an automatically corrected version of the validation set.

6.2 Validation set experiments

We report single and reranking system performance in Tables 1 and 2 respectively. Results are evaluated by Rouge (Lin, 2004), which is based on n-gram or longest common sequence (LCS) matching of strings.

6.2.1 Single systems and error correction

Among the pre-trained LMs in Table 1, BART performs the best on the validation set. Comparing error-corrected (Pegasus/BART/T5 EC) and original (Pegasus/BART/T5) inputs, we note that error-corrected input significantly boosts the performance of Pegasus. In addition to corrected entity names, the fixed input also leads Pegasus to generate 5% longer output and results in a much higher Rouge-2 score in this small dataset. This trend is less significant on BART and T5, but adding error correction has a positive impact in general.

6.2.2 Reranking

We compare the reranked systems against baselines, with or without error-corrected input in Table 2. In both cases, reranking does not have significant effect on Rouge-2, but helps Rouge-L significantly. We suspect that reranking does improve word and style choice, but the room for increasing 2-gram matches is small on the validation set.

6.3 Test set experiments

We run three sets of experiments on the test set and report results in Table 3: single systems are the same systems tested on the validation set and ensembles are reranked outputs from systems further finetuned on the validation set.

In addition to string-based Rouge (Lin, 2004), test set results are also evaluated by pre-trained language model-based BERTScore (Zhang et al., 2020b) and HOLMS (Mrabet and Demner-Fushman, 2020) metrics:

- **BERTScore** (Zhang et al., 2020b) leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity, where matching is performed greedily for each word by choosing the most similar word in the other sentence.
- **HOLMS** (Mrabet and Demner-Fushman, 2020) combines soft matching of contextual embeddings derived from pre-trained LMs and a string-based metric (Rouge-1 recall in practice).

String-based and pre-trained language model-based metrics rank summaries differently. We discuss the impact of the choice of metrics in Section 6.4.

We run two other experiments validating post-processing and the UniLM language model (Dong

et al., 2019), they perform inferior to their respective baselines and are not reported in Table 3.

We notice in single system experiments that the characteristics of the test set is still different from the validation set: all systems suffer from low recall, which leads us to perform more aggressive length-based reranking.

Length reranking. We experiment with a baseline approach that explicitly picks the longest output sentence by switching the weight of length and fidelity features in (1). The 3 systems in runs 4 and 5 are Pegasus and T5 finetuned on the validation set and the Pegasus system in run 3. Run 6 adds BART finetuned on the validation set.

We observe that this simple heuristic, together with further finetuning on the validation set, leads to significantly higher Rouge scores between runs 3 and 4 in Table 3. This change also improves HOLMS and BERTScore, suggesting that recall / coverage-based sentence selection does correlate to summarization quality in this scenario. Rouge is further improved by adding BART to the combination between runs 5 and 6.

Correcting input errors between runs 4 and 5 also helps Rouge significantly. BERTScore, which is based on word matching and utilizes BERT embeddings, is much less sensitive to small spelling errors and changes negatively. HOLMS changes positively as it has a Rouge component.

The negative change of BERTScore also suggests that we should be more cautious applying input error correction to summarization: mistakes in error correction might not hurt string-based metrics (the word is often misspelled already), but they can change the meaning of the sentence and degrade summarization quality.

Coverage reranking. In runs 7 and 8, we experiment with the the same setting as in Table 2. 3 systems are Pegasus, BART, and T5 finetuned on the validation set. These runs achieve balanced Rouge precision and recall, and the highest Rouge-2 score across all runs. There are small improvement on all metrics, which is expected, as Rouge-2 is a better indicator of summarization coverage than length.

According to BERT-based metrics, coverage-based reranking also leads to more steady improvement than length-based reranking. The overall improvement in all metrics suggests that coverage-based reranking does improve summarization quality in this task.

6.4 Lessons learned

In this shared task, we experimented with knowledge-based input error correction and coverage-oriented system reranking. These methods are effective in boosting string matching between the prediction and the reference summaries. According to Rouge metrics, our submissiong ranks first according to Rouge-1/2 metrics and ranks second according to the Rouge-L metric.

According to BERT-based metrics, however, reranking has a smaller impact on summarization quality and error correction has little to no effect: we are about 1 point below the best submission according to BERTScore and HOLMS, which are shown to often have higher correlation with human judgement (Zhang et al., 2020b; Mrabet and Demner-Fushman, 2020).

The discrepancy between the string-based and LM-based metrics makes the real improvement of summarization quality hard to measure. It is arguable that by focusing on misspellings and using coverage as surrogate for summarization quality, we might be optimizing more for the writing style and spelling, rather than the content of the summary. This shows the need of an efficient, optimizable summarization evaluation metric with high correlation with human judgement that our field agrees upon. We plan to look more into the choice of metric and optimization objectives for summarzation tasks in future work.

7 Conclusion

We reported our experiments in MEDIQA 2021 shared task 1. We used knowledge-based error correction and coverage-oriented reranking improve summarization. Our system performed well on string-based Rouge metrics, but less so on BERT-based semantic metrics. We plan to investigate methods that improve summarization according to human judgement.

References

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the summarization of consumer health questions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2228–2234, Florence, Italy.

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*.

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. PALM: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691.

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon. 2014. Improving entity linking using surface form refinement. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4609–4615, Reykjavik, Iceland.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Flight of the PEGASUS? comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5640–5646, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2021. Biomedical question answering: A comprehensive review. *arXiv preprint arXiv:2102.05281*.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of HLT-NAACL*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

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, Online.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.

Haitao Mi, Qiyu Ren, Yinpei Dai, Yifan He, Jian Sun, Yongbin Li, Jing Zheng, and Peng Xu. 2021. Towards generalized models for beyond domain api task-oriented dialogue. In *Proceedings of the 9th Dialog System Technology Challenge*.

Yassine Mrabet and Dina Demner-Fushman. 2020. HOLMS: Alternative summary evaluation with large language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5679–5688, Barcelona, Spain (Online).

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

Lina F Soualmia, Elise Prieur-Gaston, Zied Moalla, Thierry Lecroq, and Stéfan J Darmoni. 2012. Matching health information seekers' queries to medical terms. *BMC bioinformatics*, 13(14):1–15.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

X. Zhou, An Zheng, Jiaheng Yin, R. Chen, Xianyang Zhao, Wei Xu, Wenqing Cheng, T. Xia, and S. Lin. 2015. Context-sensitive spelling correction of consumer-generated content on health care. *JMIR Medical Informatics*.