WBI at MEDIQA 2021: Summarizing Consumer Health Questions with Generative Transformers

Mario Sänger[*,♣] **, Leon Weber**[*,♣,†]**, Ulf Leser**[♣]
[♣]Computer Science Department, Humboldt-Universität zu Berlin
[†]Max Delbruck Center for Molecular Medicine
{saengema,weberple,leser}@informatik.hu-berlin.de

Abstract

This paper describes our contribution for the MEDIQA-2021 Task 1 question summarization competition. We model the task as conditional generation problem. Our concrete pipeline performs a finetuning of the large pretrained generative transformers PEGASUS (Zhang et al., 2020a) and BART (Lewis et al., 2020). We used the resulting models as strong baselines and experimented with (i) integrating structured knowledge via entity embeddings, (ii) ensembling multiple generative models with the generator-discriminator framework and (iii) disentangling summarization and interrogative prediction to achieve further improvements. Our best performing model, a fine-tuned vanilla PEGASUS, reached the second place in the competition with an ROUGE-2-F1 score of 15.99. We observed that all of our additional measures hurt performance (up to 5.2 pp) on the official test set. In course of a post-hoc experimental analysis which uses a larger validation set results indicate slight performance improvements through the proposed extensions. However, further analysis is need to provide stronger evidence.

1 Introduction

The internet provides a wealth of information on health topics through specialised websites, forums, blogs and social networks. Increasingly, consumers are using these information sources to answer their medical and health-related questions. In the course of this development, also the consumers' expectations regarding search engine functionalities have become much more demanding. Instead of reading through a list of relevant articles returned by a classical search engine, short and precise passages are now expected to answer questions. This transformation also has an impact on the technologies used to fulfill the user's information needs. In particular, approaches for automatic questions answering as well as automatic summarization and simplification of (long) articles has received a lot of attention by researchers in recent years (Allahyari et al., 2017; Kwiatkowski et al., 2019; Narayan et al., 2018b; See et al., 2017; Weber et al., 2019). This trend is also addressed by Task 1 of the MEDIQA 2021 shared task (Ben Abacha et al., 2021) through investigating consumer health-questions asked on the (experimental) medical question answering system CHiQA[1]. As we participated only in this task, we refer to it as Shared Task (ST) in the following.

The goal of Task 1 was to foster the development of new summarization approaches, specifically designed for the challenges of long and potentially complex consumer health questions. One major challenge of CHiQA is the extraction of the user's main concern from the question text. The given questions are often lengthy and contain a lot of peripheral information, which makes automatic processing and answering (much more) difficult. Recent studies highlight that expert-based summarizations of such questions can lead to significant enhancements of the overall QA process (Ben Abacha and Demner-Fushman, 2019). Effective automatic summarization methods could therefore play a key role for improving medical question answering.

We contribute to this task by first building a baseline using the general conditional generation framework and then investigating three modifications to summarize the consumer health questions. Our baseline relies on finetuning the large pretrained generative transformers PEGASUS (Zhang et al., 2020a) and BART (Lewis et al., 2020). We explore three different strategies to improve the performance of these baseline models, i.e. (i) integrating structured knowledge via entity embeddings, (ii) ensembling multiple generative models with the generator-discriminator framework and (iii) dis-

[*] These authors contributed equally. Author order was determined by coin flip.

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

Proceedings of the BioNLP 2021 workshop, pages 86–95
June 11, 2021. ©2021 Association for Computational Linguistics

entangling summarization and question word prediction. Our best performing model, a fine-tuned vanilla PEGASUS, reached the second place in the competition. We observed that all measures hurt performance (up to 5.2 pp) on the evaluation set. However, a post-hoc experimental analysis (see Section 3), using a larger validation set, indicates slight improvements through the model extensions.

The remainder of the paper is organized as follows: the next section introduces our baseline and the three extension strategies in detail. Section 3 highlights and discusses the experiments and results we obtained in our own evaluation as well as in the official assessment. The paper concludes which a summary of the main findings.

2 Methods

2.1 Data & Baselines

The shared task provides only an official validation and test set as data. For training data, we follow the tasks' organizers suggestion to use the MeQSum corpus which consists of 1,000 consumer health questions and their summaries.

We model the summarization task as conditional generation, in which a model is prompted with the original question and then generates the summary in an autoregressive fashion. We base our implementation[2] on the huggingface transformers library (Wolf et al., 2020) and experiment with the included pretrained generative transformers *bart-base*[3], *bart-large*[4], *pegasus-large*[5] and *pegasus-xsum*[6]. *pegasus-xsum* is a version of PEGASUS that was already finetuned for summarization on the Xsum dataset (Narayan et al., 2018a). For all models, we use a learning rate of $3e-5$ and train for 10 epochs. We use beam search for decoding and tune the search parameters on the validation set. We independently evaluated $\{1, 10\}$ as the number of beams and the $\{0.7, 0.8, 0.9, 1.0\}$ for the length penalty and found 10 and 0.8 to be optimal.

2.2 Integration of structured knowledge via entity embeddings

In initial analyses, we noticed that most question summaries revolve around a few central entities such as specific diseases or medications which are almost always mentioned in the source text. Furthermore, all of the generative transformers that we used were trained on texts from the general domain, in which such entities presumably are rare. We conjectured that it could be beneficial to explicitly provide entity information to the model. We approach this by first applying a domain-specific NER model to the source text and then enriching the input embeddings of the transformer with the found entities. Formally, we extend the computation of the i'th input embedding in the transformer to:

$$e_i = w_i + p_i + s_i + n_i, \qquad (1)$$

where w_i, p_i, s_i are the standard subword, position and sequence type embeddings which are initialized with the weights of the pretrained transformer. n_i is a randomly initialized embedding, which represents the type of the named entity to which the token i belongs (including *None*) and has the same dimensionality as the other transformer embeddings. Note, that s_i is set to zero for transformers which do not use sequence type embeddings such as BART.

We experiment with two different NER models: (i) *HunFlair* (Weber et al., 2021), a state-of-the-art BioNER tagger and (ii) a custom *Flair* (Akbik et al., 2019) model trained on the CHQA corpus (Kilicoglu et al., 2018) consisting of manual annotations for the central entities of consumer health questions. Specifically, we use the *Disease* and *Chemical* models of *HunFlair* and the PC-harmonization of the CHQA corpus.

2.3 Ensembling multiple generative transformers

In preliminary experiments, we found that ensembling generative transformers by simply averaging the logits of different models hurt performance. Thus, we investigate a different strategy for ensembling generative models. We first use each model m of the ensemble to generate n summaries $\{s_{m1}, \ldots, s_{mn}\}$ conditioned on the original question q and then use a discriminative model to select the question-summary pair with the highest probability. The n different summaries are generated by simply taking the final generations of the top-n

[2] Our code is publicly available under `https://github.com/leonweber/bionlp21_summarize`

[3] `https://huggingface.co/facebook/bart-base`

[4] `https://huggingface.co/facebook/bart-large`

[5] `https://huggingface.co/google/pegasus-large`

[6] `https://huggingface.co/google/pegasus-xsum`

scoring beams. We implement the discriminator as a BERT (Devlin et al., 2019) model that receives both the original question q and a question summary s produced by one of the ensembled models and predicts the ROUGE-L-F1 score between both *ROUGE-L-F1*(s, q) using a tanh output layer. The model is trained via an L2-loss. More formally,

$$\mathbf{h} = \textit{BERT}_{\texttt{[CLS]}}(s, q) \tag{2}$$

$$o = 0.5 \cdot \textit{tanh}(\mathbf{W} \cdot \mathbf{h} + \mathbf{b}) \tag{3}$$

$$\mathcal{L} = \|\textit{ROUGE-L-F1}(s, q) - o\|_2, \tag{4}$$

where $\textit{BERT}_{\texttt{[CLS]}}$ is the BERT-embedding of the special [CLS] token, $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters and $\mathcal{L}$ is the loss value.

For training the discriminator, we require generated summaries that are close to the generated summaries on the test data. We cannot simply use the training data of the generators to create the training data for the discriminator, because we expect the distributions of the generated summaries for seen and unseen data to be significantly different. Thus, we split MeQSum training data in a 75% / 25% fashion and use the first chunk for training the generators and the combination of both to train the discriminators. The full training process is illustrated in Figure 1a.

2.4 Disentangling summarization and interrogative prediction

We observed that the consumer questions cover different categories of health-related issues in the ST data, e.g. possible side-effects of certain drugs, suitable treatments for specific diseases or food-related questions. We conjectured that providing the putative category of the question to the summarization model could guide the generator towards a better summary. Moreover, we recognized that the different categories are aligned to some extent with the interrogative of the target questions summaries. Based on these two observations, we designed a third modification by creating a separate model to predict the putative interrogative, which acts as a surrogate for the different question categories.

To this end, we implement a BERT-based classification model which gets the original user question as input and predicts the interrogative of the target question summary. We combine the classification model with the output of our baseline method using a three-step approach: (i) we generate m question summaries using a generative transformer, (ii) we predict the interrogative given the original user question based on the trained classification model and (iii) selected the highest ranked candidate questions which starts with the predicted interrogative as target summary. The process is illustrated in Figure 1b. To train the classification models we use the data from the MeQSum corpus but just take the first word of the summaries as goldstandard interrogative. Because in this model there is no dependency between generative and classification models (as opposed to our generator-discriminator framework), the classification model can be trained on the complete training data.

3 Results

3.1 Evaluation setting

We evaluate our models in two different settings.

Setting 1 For our ten submissions to the shared task, we typically use some combination of MeQSum and the validation data for training. For model selection and evaluation of our modifications, we use the official validation set of the shared task. Finally, we report scores of our models on the shared tasks' hidden test set.

Setting 2 While preparing our runs, we noticed that the variance of the results on the validation and test set is rather high, which probably has to do with the small amount of validation and test data (50 and 100 questions respectively). To evaluate the performance impact of our modifications in a more stable manner, we devised a second evaluation setting after the ST submissions were closed. For this, we combine the MeQSum data and the shared task validation data in a single dataset and then split it into a train and validation set, reserving 200 questions for validation, which leaves 850 questions for training. We ensure that for each split the ratio of original MeQSum and validation data is equal. For each result, we compute three different runs with different random seeds and report the average and standard deviation.

Table 1 highlights the used splits of the two different data settings and provides basic statistics for them. The results for both settings differ significantly and thus, we report results for both settings in the following sections. In the official evaluation of the shared task, the approaches were ranked according to the achieved ROUGE-2-F1 score.

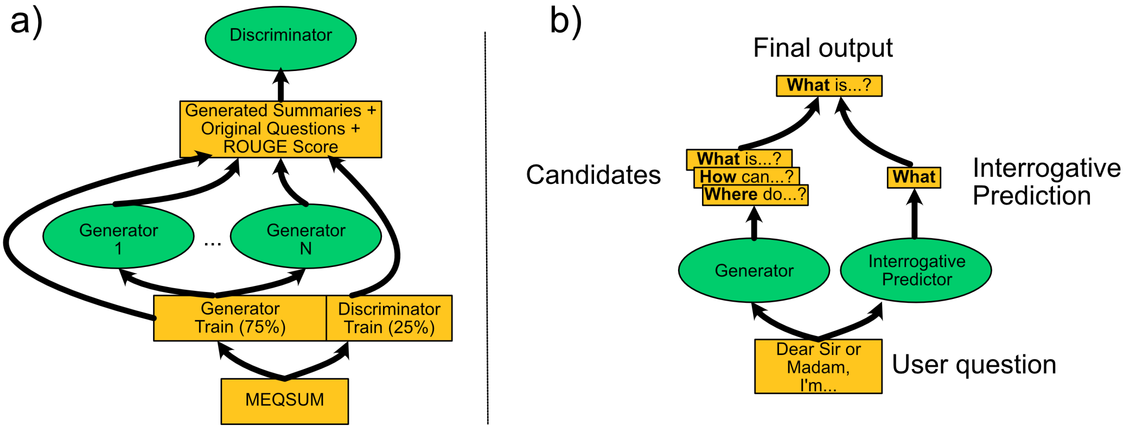

Figure 1: **(a)** Training an ensemble of multiple generators together with a discriminator. Resources are depicted as yellow rectangles and trained models as green ellipses. **(b)** Predicting summaries with the interrogative predictor. Resources are drawn as yellow rectangles and models as green ellipses.

Setting	Split	Questions	Tokens / Question			Tokens / Summary		
			Mean	Min	Max	Mean	Min	Max
Setting 1	Training (MeQSum)	1000	60.78	5	378	10.04	3	26
	Validation	50	64.16	9	234	9.34	4	19
Setting 2	Training	850	59.60	8	348	9.70	3	26
	Validation	200	66.64	5	378	10.18	3	26

Table 1: Overview about the data sets and splits used for training and evaluation in Setting 1 and 2. For Setting 2, we use all instances from the official training data (MeQSum) and validation data and randomly assign them to the two splits. We ensure that for each split the ratio of original training and validation data is equal.

3.2 Final evaluation results

Our best performing model achieved a ROUGE-2-F1 score of 15.99% on the hidden test set, leading to a second place in the competition. However, all top-5 models achieve results that are very close, and ranks change when different metrics are used. The top five of the official leaderboard is reproduced in Table 2. This best performing model is one of our baselines based on *pegasus-large* fine-tuned on the combination of MeQSum and the ST validation set. The results of our ten runs on the official hidden test set together with a description of each run can be found in Table 5.

3.3 Baseline results

In preliminary experiments on the ST validation set, we found that *pegasus-large* works better than *bart-large* when the model is fine-tuned on MeQSum and evaluated on the ST validation set (ROUGE-L-F1 of 33.32 vs. 32.82). Based on this result, we opted to select *pegasus-large* as baseline model for our submissions (refer to Section 3.7 for a discussion of challenges in model selection). In the official evaluation (i.e. Setting 1) the vanilla *pegasus-large* model achieves the best performance of all our submitted runs with an ROUGE-2-F1 score of 15.99 (see Run 1 in Table 5). In a post-hoc analysis, we noticed that in the consumer questions spelling errors for crucial pieces of information such as diseases are common and that the models tend to copy those spelling errors into the summary of the question. Thus, our approach probably could have benefited from incorporating a spell-checking tool that corrects the spelling errors in the health questions.

Setting 2 uses the same basic models, but relies on a different training setup. Table 3 shows the performance scores. The best performance is achieved by *bart-large* with ROUGE-1-, ROUGE-2 and ROUGE-L-F1 scores of 52.91, 34.06 and 49.88. This represents an improvement of 0.55pp concerning ROUGE-2-F1 to the next best model (*bart-base*). In this setting, the BART-based models achieve better results than the PEGASUS models.

3.4 Entity embedding results

We evaluate the addition of entity embeddings to a generative transformer using bart-base. For detecting entities, we experiment with the two different NER models HunFlair and a custom Flair model trained on the *PC*-harmonization (Passonneau and Carpenter, 2014) of the CHQA corpus. The results for Setting 2 can be found in Table 3. Adding entity embeddings to the input representation improves results consistently, leading to a gain of 0.3pp and 1.01pp in ROUGE-2-F1 over our bart-base baseline. However, we did not observe any gains in our preliminary experiments on the ST validation set and thus did not evaluate the models with entity embeddings in Setting 1. The submission of new runs was not possible at the time of writing.

3.5 Ensemble results

All results for the generator-discriminator ensembles in Setting 1 (on the hidden test set) can be found in Table 5, where each row with Type 'GD' corresponds to one configuration of a generator-discriminator ensemble. Considering ROUGE-2-F1, the best generator-discriminator result (run 7) still performs 1.4 pp worse than our best baseline model. This run used only one generator based on pegasus-large to produce ten candidates per question and a bert-large discriminator to select the most promising summary. The only setting in which a generator-discriminator model outperforms our strongest baseline on the hidden test set is run 8 which gains 0.2 pp under the BERTScore metric (Zhang et al., 2020b), making it the overall top ranking run of the ST under this metric. This run uses a single pegasus-large generator proposing ten candidate summaries per question and an ensemble of three different bert-large discriminators.

In Setting 2, we observed considerable gains by using an ensemble of bart-base, bart-large, pegasus-large and pegasus-xsum, while using a single bert-base as the discriminator, using only the most probable output sequence per model as candidate. Compared to pegasus-large, this configuration leads to an improvement of 2.16pp in ROUGE-1-F1, 1.46pp in ROUGE-2-F1 and 2.27pp in ROUGE-L-F1.

We also investigated the performance ceiling for our ensembling approach by evaluating the ensemble under a perfect discriminator, which always selects the summary yielding the highest Rouge-L-F1 score. Under this setting, our ensemble achieved a Rouge-2-F1 score of 44.87 which is an improvement of 10.9 pp. This shows the promise of our ensembling approach and suggests that a worthwhile path to obtain better results would be to improve the discriminator.

Rank	Team name	ROUGE-1-F1	ROUGE-2-F1	ROUGE-L-F1	HOLMS	BERTScore-F1
1	damo_nlp (summc)	35.14	16.08	31.31	56.77	68.98
2	**WBI**	**33.40**	**15.99**	**31.49**	**57.67**	**69.96**
3	NCUEE-NLP	33.52	15.97	30.90	57.87	69.60
4	yamr	32.80	15.25	30.38	57.86	68.77
5	Saama	33.33	15.18	29.50	57.72	69.38

Table 2: Top five of the official results for subtask one (ranked by ROUGE-2-F1). All scores are given in percent. In total 23 teams participated in this subtask. Our contribution is displayed in bold. These numbers correspond to our evaluation Setting 2.

Model type	Gen. model(s)	Add-on	ROUGE-1-F1	ROUGE-2-F1	ROUGE-L-F1
Baseline	bart-large	-	52.91 (± 0.91)	34.06 (± 1.01)	49.88 (± 0.66)
	bart-base	-	52.17 (± 0.14)	33.49 (± 0.84)	49.36 (± 0.32)
	pegasus-large	-	51.06 (± 0.78)	32.51 (± 0.72)	48.28 (± 0.68)
	pegasus-xsum	-	51.47 (± 0.28)	32.65 (± 0.58)	48.90 (± 0.30)
Entity embeddings	bart-base	HunFlair	52.16 (± 0.45)	33.79 (± 0.46)	49.24 (± 0.27)
	bart-base	CHQA flair model	53.17 (± 1.58)	34.5 (± 1.30)	50.22 (± 1.43)
Generator-discriminator	bart-base bart-large pegasus-large pegasus-xsum	bert-base	53.22 (± 1.81)	33.97 (± 1.40)	50.55 (± 1.75)
Interrogative prediction	pegasus-large	bert-base	52.11 (± 0.36)	33.71 (± 0.85)	49.21 (± 0.66)
	pegasus-large	bio-bert	52.22 (± 0.60)	33.42 (± 0.70)	49.26 (± 0.53)
	pegasus-large	biomed-roberta	52.66 (± 0.67)	33.71 (± 0.81)	49.58 (± 0.85)
	pegasus-large	bio-bert biomed-roberta	52.28 (± 0.58)	33.47 (± 0.69)	49.40 (± 0.67)

Table 3: Overview of Setting 2 evaluation results. For each experiment, we list the used generative transformer(s) and (if applicable) utilized complementary models (Add-on). For entity embeddings add-on models are named entity recognition models. In case of the generator-discriminator framework it's the discriminator model and regarding interrogative prediction it defines the applied classification model(s). For each experiment, we compute three different runs with different random seeds and report the average and standard deviation.

3.6 Interrogative-predictor results

For evaluating our interrogative prediction approach we experimented with different transformer-based models, pre-trained on either general domain or biomedical data, for classification: BERT[7], BioBERT (Lee et al., 2020)[8], BioMed-RoBERTa (Gururangan et al., 2020)[9] and multiple of these models arranged in an ensemble. All models are learned on the training portion (for each evaluation setting). For all models we use *pegasus-large* as generative model and produce 10 candidate summaries per user question.

As shown in Table 3 we observe clear performance improvements of this approach compared to the baseline when evaluated in Setting 2. Here, the best results are achieved with the BioMed-RoBERTa model. In this configuration, the model achieves a ROUGE-2-F1 score of 33.71 which represents an increase of 1.20 pp compared to the vanilla *pegasus-large* result. Again, the results achieved in the official evaluation (Setting 1) show a different picture. In this setting, the usage of an ensemble of three interrogative classification models lowers the performance by 2.6 pp (see Run 3 in Table 5).

We also investigated the accuracy of the interrogative prediction models. Table 4 highlights the achieved accuracy and macro $F1$-scores of the three models. All models predict the correct interrogative for only half of the consumer questions. An analysis of the predictions showed that all models are biased towards the majority classes, i.e. interrogatives with a high support in the training data.

Like in the generative ensemble setting, we further checked the potential performance gains of the interrogative prediction using a perfect classifier. For this, we took the gold standard interrogative and use the first generated summary candidate which starts with this interrogative as prediction. If no generated summary starts with the gold interrogative we use the highest ranked candidate. Using this selection scheme we reached an ROUGE-2-F1 score of 39.72 in Setting 2 which represents an increase by 7.21 pp over the baseline *pegasus-large* model. Again, this accentuates the suitability of the proposed approach.

Model	Accuracy	$F1$
bert-base	0.530	0.103
bio-bert	0.525	0.095
biomed-roberta	**0.555**	**0.228**

Table 4: Overview of the performance of the three interrogative classification models. For each model we report accuracy and macro $F1$ score. Bold figures highlight the highest value per column.

3.7 Discussion of result differences between Setting 1 and Setting 2

Tables 2 and 3 reveal enormous performance differences between Setting 1 (the official evaluation results) and Setting 2 (our post-hoc experimental analysis). In Setting 1, none of our proposed extensions leads to consistent quantitative improvements of the results and the best performance is achieved by an vanilla generative transformer. In contrast in Setting 2, we see (at least) slight benefits from all three strategies.

Explaining these results and differences is difficult for several reasons. Concerning Setting 2, the high variance of the results (see Table 3) prevents a clear conclusion. Results of the methods vary with different random initializations and are also quite sensitive to hyperparameter settings. Often the differences of the methods lie within the range of the standard deviation making it unclear whether the findings would hold up in further analysis or other contexts.

Regarding Setting 1, the small size of the evaluation data (only 100 instances) puts any conclusions about the quality of the proposed methods into question. In Setting 2, we tried to mitigate the problem of small test data by increasing the number of test instances, however the results remain unstable. Furthermore, weaknesses of the ROUGE metric, e.g. handling of synonyms, abbreviations or enumerations, must be taken into account in the result interpretation (Schluter, 2017; Kané et al., 2019). The automatic evaluation of generated summaries remains a research field in itself (Zhang et al., 2020b). In summary, we neither believe that the results from Setting 1 provide strong evidence of the extension's inappropriateness, nor that the results from Setting 2 allow a convincing statement about their positive effects. To this end, further investigation is necessary in order to draw definitive conclusions about our proposed modifications.

[7] https://huggingface.co/bert-base-cased

[8] https://huggingface.co/dmis-lab/biobert-v1.1

[9] https://huggingface.co/allenai/biomed_roberta_base

Run	Type	Description	ROUGE-2	HOLMS	BERTScore-F1
1	B	pegasus-large finetuned on MeQSum and validation data	**16.0**	**57.7**	70.0
2	B	pegasus-large first finetuned on MeQSum and then on validation data	12.4	55.5	69.3
3	IP	pegasus-large finetuned on MeQSum and validation data with ensemble of interrogative predictors consisting of two biobert and one biomed-roberta model	13.4	56.4	69.0
4	GD	Generator ensemble of bart-base, bart-large, pegasus-large and pegasus-xsum with one candidate summary per model and bert-base as discriminator	11.8	55.5	68.4
5	B	pegasus-xsum finetuned on MeQSum and validation data	12.4	55.5	68.7
6	GD	Same configuration as in run 4 but with an ensemble of discriminators consisting of bert-base, roberta-base and biobert	11.4	55.4	68.2
7	GD	pegasus-large trained on MeQSum with ten candidate summaries and a bert-large discriminator trained on MeQSum to select the best one	14.6	57.3	69.8
8	GD	Same configuration as in run 7 but with an ensemble of three different bert-large discriminators trained on MeQSum	14.2	57.0	**70.2**
9	GD	Same configuration as in run 7 but the bert-large discriminator is trained on MeQSum and validation data	12.0	55.4	68.9
10	GD	Same configuration as in run 8 but the the discriminators are trained on MeQSum and validation data	12.0	55.4	69.5

Table 5: Official results for our submitted runs for subtask one. In total we submitted 10 runs. The runs can be categorized according to their type into baseline models (B), models using interrogative prediction (IP) or the generator-discriminator framework (GD). The highest value per metric is highlighted in bold. This corresponds to our evaluation Setting 1.

4 Conclusion

In this work we investigate the large-scale pre-trained generative transformers PEGASUS and BART for the task of health-related consumer question summarization in the context of the MEDIQA 2021 shared task (Task 1). We propose and evaluate three different strategies, i.e. integrating structured knowledge via entity embeddings, utilizing a generator-discriminator framework and applying interrogative prediction, to extend these strong baseline models. Our best performing model, a fine-tuned pegasus-large transformer, reaches an ROUGE-2-F1 score of 15.99 and is ranked second place in the competition. Experimental results for our proposed extensions show a mixed picture and further analysis is needed to assess the quality of these extensions.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the role of question summarization and information source restriction in consumer health question answering. *AMIA Summits on Translational Science Proceedings*, 2019:117.

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 SIG-BioMed Workshop on Biomedical Language Processing, NAACL-BioNLP 2021*. Association for Computational Linguistics.

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Hassan Kané, Yusuf Kocyigit, Pelkins Ajanoh, Ali Abdalla, and Mohamed Coulibali. 2019. Towards neural similarity evaluator. In *Workshop on Document Intelligence at NeurIPS 2019*.

Halil Kilicoglu, Asma Ben Abacha, Yassine Mrabet, Sonya E. Shooshan, Laritza Rodriguez, Kate Masterton, and Dina Demner-Fushman. 2018. Semantic annotation of consumer health questions. *BMC Bioinform.*, 19(1):34:1–34:28.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. 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, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018b. 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*, pages 1797–1807.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Trans. Assoc. Comput. Linguistics*, 2:311–326.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45.

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.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6151–6161.

Leon Weber, Mario Sänger, Jannes Münchmeyer, Maryam Habibi, Ulf Leser, and Alan Akbik. 2021. HunFlair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. *Bioinformatics*. Btab042.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.