# MNLP at MEDIQA 2021: Fine-Tuning PEGASUS for Consumer Health Question Summarization

[1]Huong Ngoc Dang* [1]Jooyeon Lee* [2]Samuel Henry [1]Özlem Uzuner
[1] Department of Information Science and Technology
George Mason University, Virginia, United States
[2] Department of Physics, Computer Science and Engineering
Christopher Newport University, Virginia, United States
[1]{hdang20,jlee252,ouzuner}@gmu.edu, [2]samuel.henry@cnu.edu

## Abstract

This paper details a Consumer Health Question (CHQ) summarization model submitted to MEDIQA 2021 for shared task 1: Question Summarization. Many CHQs are composed of multiple sentences with typos or unnecessary information, which can interfere with automated question answering systems. Question summarization mitigates this issue by removing this unnecessary information, aiding automated systems in generating a more accurate summary. Our summarization approach focuses on applying multiple pre-processing techniques, including question focus identification on the input and the development of an ensemble method to combine question focus with an abstractive summarization method. We use the state-of-art abstractive summarization model, PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization), to generate abstractive summaries. Our experiments show that using our ensemble method, which combines abstractive summarization with question focus identification, improves performance over using summarization alone. Our model shows a ROUGE-2 F-measure of 11.14% against the official test dataset.

## 1 Introduction

The MEDIQA 2021 shared task consists of several independent tasks: task 1 is Question Summarization, task 2 is multi-answer summarization, and task 3 is Radiology Report Summarization. We participated in task 1, Question Summarization. We approached the task by developing an ensemble learning method that combines information from automatic question focus identification with information from a state-of-the-art summarization model. We also studied the effects of different preprocessing techniques for this challenge. The descriptions of the dataset are shown in the task guidelines (Ben Abacha et al., 2021). The training datasets are from Ben Abacha and Demner-Fushman (2019b) along with the focus of each question. The test dataset contains consumer health questions only.

## 2 Related Works

The goal of Consumer Health Question Answering (CHQA) is to construct an automated question answering system aimed toward answering questions from individuals who are unlikely to possess professional medical knowledge. Typical consumer health questions include requests for information regarding symptoms of particular diseases, queries regarding possible diseases from individuals experiencing symptoms, and whether an individual would be safe to mix specific medications and so forth. In this field, there are circumstances in which individuals submit straightforward questions, but there are many cases where people list extra background and other unnecessary information which are not required to answer their question. In fact, this additional information can essentially serve as a source of noise which can reduce the effectiveness of the QA system as a whole.

Recent CHQA systems employ pipeline architectures that utilize Question Understanding, Information Retrieval and Answer Generation components sequentially (Demner-Fushman et al., 2019). This architecture facilitates modular optimization. Furthermore, it allows individual components to be swapped, either for need or to provide special features. This allows the entire QA system to adapt to the specific nature of the problem at hand. As previously mentioned, many CHQs possess extraneous information in addition to the primary question. Therefore, the Question Understanding component of such an architecture is especially important, and improvements to it can be particularly beneficial to the overall CHQA system. Facilitating Ques-

*These authors contributed equally to this work



*Proceedings of the BioNLP 2021 workshop*, pages 320–327

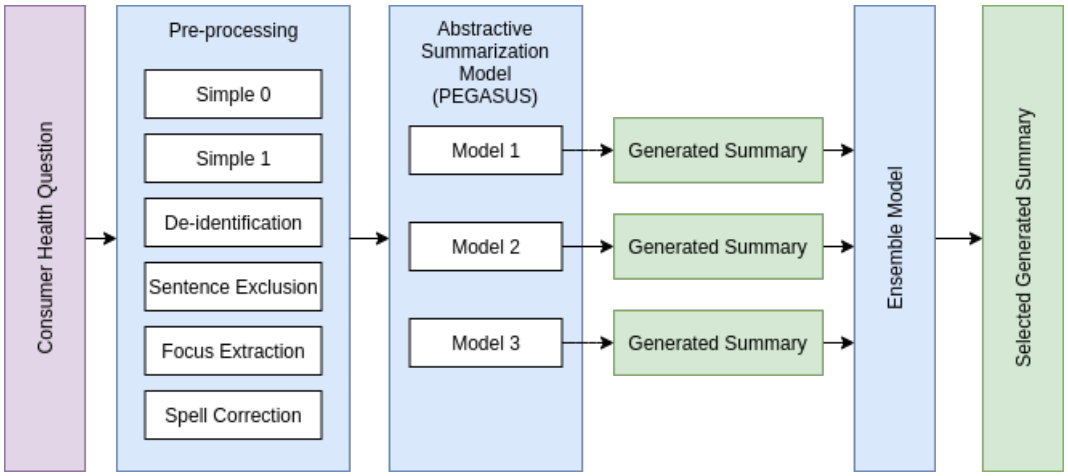


Figure 1: System Architecture.

tion Understanding through summarizing the consumer health questions has demonstrated significant improvement as shown by Ben Abacha and Demner-Fushman (2019b) and Ben Abacha and Demner-Fushman (2019a). Thus, given the benefits to the overall QA system, improving upon existing summarization methods was selected as task 1 in MEDIAQA 2021. In this component, the extraneous information can be removed via preprocessing prior to inputting into further stages of the QA system. Different preprocessing methods have also been explored to perform this task, and a performance improvement on deep learning models has been shown (Camacho-Collados and Pilehvar 2017; Husain et al. 2020).

### 2.1 Consumer Health Question Understanding

Robust CHQA systems could serve as a component in a broader solution to inform the public of the latest medical updates and breakthroughs, leading to more optimal outcomes for both individuals and the public as a whole.

In Question Understanding, recent breakthroughs relevant to CHQA have included: Ben Abacha and Demner-Fushman (2019), which demonstrated that retrieving entailment answers for CHQA systems many not gather any answers; Ben Abacha and Demner-Fushman (2019b), which studied the role of summarization on CHQA; and Roberts et al. (2014) proposes decomposition methods and techniques for consumer health datasets.They suggest decomposing the questions into focus of the question, exemplification, question sentence(s), background sentence(s) and "ignore" sentence(s).

### 2.2 Abstractive Summarization

Abstractive Summarization aims to re-write the given input in a shorter form. This is opposed to Extractive Summarization, which aims to select essential sentences from the given input only. There are different approaches to Abstractive Summarization, such as structured-based, semantic-based, deep learning-based, discourse, and rhetoric-based (Gupta and Gupta, 2019).

In this paper, we selected a deep learning approach. Deep learning methods include Pointer Generator Networks See et al. (2017), Pre-training with Extracted Gap-sentences for Abstractive Summarization (PEGASUS) (Zhang et al., 2019), Multi-Document Summarization by Niu et al. (2017) and others (Kouris et al., 2019; Khatri et al., 2018). We selected Pointer Generator Networks as our baseline method, because it showed high performance in summarizing consumer health questions Ben Abacha and Demner-Fushman (2019b) and compare the results with PEGASUS.

## 3 Methodology

Our model follows a traditional language generation pipeline: pre-processing, abstractive summarization, and post-processing. We experimented with several different combinations of preprocessing to generate multiple summaries from a single given question. We selected the best summary in the post-processing stage from these numerous generated summaries, which use ensemble learning.

### 3.1 Dataset

The MEDIQA 2021 event organizers provided three different datasets: a training set, a validation set, and a testing set. The training dataset, called MeQSum, is from Ben Abacha and Demner-Fushman (2019b) and consists of 1000 pairs of consumer health questions and corresponding summaries. Of the questions in the training dataset, 658 questions had both SUBJECT and MESSAGE entered by users, while 342 Questions lacked a SUBJECT. Information such as [SUBJECT], [CONTACT], [NAME] and [LOCATION] were de-identified. The validation dataset consists of 50 raw consumer health questions with corresponding summaries, focus, and type for each question. The testing dataset consists of 100 raw consumer health questions.

### 3.2 Pre-processing

The goal of pre-processing is 1) to make the abstractive summarization model focus on the important information by removing the redundant strings from both the training and validation/test set, and 2) minimize the difference between the training dataset versus both the validation and test datasets as described in 3.1. We try multiple pre-processing techniques for the training dataset and both the validation and test datasets. The outputs generated by the different combinations of pre-processing techniques served as inputs into our ensemble post-processing stage.

#### 3.2.1 Simple Pre-processing

We employed two different simple pre-processing steps:

1. “Simple0” which removes the text "SUBJECT: " and "MESSAGE: ", replaces "\n" by " ", and removes already tagged named entities: [LOCATION], [NAME], [CONTACT], [DATE], [PROFESSION], [AGE], [ID] from the training set.

2. “Simple1” which removes the text "SUBJECT: " and "MESSAGE: " and replace "\n" by " " from the training set.

#### 3.2.2 De-identification

[SUBJECT], [CONTACT], [NAME] and [LOCATION] terms are de-identified in training set, but not in the validation/test set. For consistency and to reduce variation between these terms, we apply de-identification on the dataset with Spark NLP (Kocaman and Talby, 2021). The Spark De-identification model was trained on n2c2 2014: De-identification and Heart Disease Risk Factors Challenge (Stubbs and Uzuner, 2015). This model allows us to mask information such as [LOCATION], [NAME], [CONTACT], [DATE], [PROFESSION], [AGE] and [ID], which were de-identified. To prevent inadvertently masking essential medical terms, we used stanza Bio NER models (Zhang et al., 2020) to identify these medical terms and omit them from masking.

#### 3.2.3 Sentence Exclusion

Sentences such as "Hi", "Thank you in advance, regards", "kindly advise me" and others do not improve summarization performance, yet also exhaust the computational time and resources by increasing the input sequence size. Thus, before input into the summarization model, we remove these sentences. For this effort, we used 10 different Stanza Bio NER models. The differentiating factors between these models are the datasets they were trained on. The datasets consist of one of 8 biomedical datasets or 2 clinical datasets, specifically: i2b2-2010, Radiology, NCBI-Disease, BC5CDR, BioNLP13CG, JNLPBA, AnatEM, BC4CHEMD, Linnaeus, and S800. If none of these 10 models found any medical terms in a sentence, we excluded that sentence from the dataset. The models are ordered by priority, high to low, and once an entity was found using one model, we kept the sentence and began processing the next.

#### 3.2.4 Focus Extraction

Roberts et al. (2014) defines *Focus* as a Noun Phrase indicating the theme of the consumer health question. We believe that by incorporating the focus into our summarization model, we can increase the overall performance. We test focus impact on both pre-processing and post-processing. We added the focus in front of the question during pre-processing and used the combined strings as an input of the abstractive summarization model. During post-processing, we used focus to rank the output accuracy as described in more detail in Section 3.4.

To extract a focus, we explored two different methods: Focus Detection and Focus Generation. For Focus Detection, we employed Named Entity Recognition (NER) with hybrid of two neural networks suggested by Chiu and Nichols (2015). The paper shows high performance with bidirectional

Long Short-term Memory (Bi-LSTM) and Convolutional Neural Network(CNN) architecture with the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003). The architecture automatically detects word and character level features by having the CNN extract features from words and by using a Bi-LSTM to tag Named Entities. We test Bi-LSTM with CNN and Recurrent Neural Network (RNN) with CNN, LSTM with CNN, and Gated Recurrent Unit (GRU) with CNN.

The risk of using NER methods to detect focus is that there is a possibility of not extracting any focus from a given question. However, focus generation uses language generation techniques, which ensures there is a focus for each given question, though the accuracy of focus is often lower compared to NER techniques. We chose Pointer Generator Networks (PG) (Ben Abacha and Demner-Fushman, 2019b) for focus generation. The model is hybrid of a sequence-to-sequence model (Sutskever et al., 2014) and a pointer network (Vinyals et al., 2017). This hybrid model allows copying words from the source text via pointing that handles out-of-vocabulary words efficiently while retaining the ability of generating new words. The Question Decomposition dataset provided by Roberts et al. (2014) was used to train and evaluate the focus extraction. The dataset includes manually annotated 1496 questions.

#### 3.2.5 Spell Correction

Consumer health questions tend to include misspelled words. This can lead to many problems in downstream question processing. A problem unique to summarization models is that summarization models generate summaries based on words it has seen in the dataset before. Therefore the model may generate summaries with misspellings. To reduce incorrect word generation, we use Microsoft Bing Spell Check API (Microsoft, 2016) to correct misspelled words. This API recognizes misspelled words in the input sentence and provides suggestions with confidence scores. We replace these words with the suggested words with the highest confidence score.

### 3.3 Abstractive Summarization

We compare two different abstractive summarization models, Pointer Generator (PG) networks and PEGASUS.

#### 3.3.1 PEGASUS

PEGASUS (Zhang et al., 2019) is a Sequence-to-Sequence model based on Transformer. It is pre-trained on massive text corpora with a self-supervised objective called Gap Sentences Generation (GSG). This objective is tailored for abstractive text summarization because the authors of PEGASUS model hypothesize that a pre-training objective that more closely resembles the downstream task leads to better and faster fine-tuning performance. In fact, PEGASUS model using this GSG objective pre-trained on newswire C4 and HugeNews corpora push forward state-of-the-art models on 12 summarization tasks.

In real-world practice, to generate summaries on a specific domain such as news, science, emails, and patents, PEGASUS should be fine-tuned using some supervised samples in that specific domain. Particularly in our shared task, the biomedical questions which need summarizing are related to the biomedical domain. To generate summarized answers on the validation dataset, we use a pretrained model that is fine-tuned on the PubMed dataset by continuing training the model with the MedQSum dataset to obtain a biomedical question summarizer. Hyperparameters we used to fine-tune PEGASUS are described in Table 1.

#### 3.3.2 Pointer Generator Networks

We compare PEGASUS with the Pointer Generator Network described in Section 3.2.4. We train this model with the pre-processed dataset. Pointer Generator Networks (Ben Abacha and Demner-Fushman, 2019b) generate summaries using 128 dimensions of word embedding trained with the summary dataset, hidden state vectors of 256 dimensions, a learning rate of 0.15, and with beam search of size 4. For our experiment, we use the hidden vector size of 256 dimensions, learning rate of 0.01 and 210 size of word vectors. We use pretrained word vectors with the size of 200. The vectors are from BioWordVec (Yijia et al., 2019), which are trained on PUBMED and MIMIC-III. 10 vectors are zeros and ones of Named Entities (NE). If a word is a medical-related entity, it is set as ones. Otherwise, it is set as zeros. The NEs are decided using spaCy pretrained NER models. Detailed hyperparameters are shown in Table 1.

### 3.4 Post-processing

For the post-processing, we employ an ensemble learning technique. Ensemble learning aims to re-

| | PG Networks | | PEGASUS |
|---|---|---|---|
| Dataset | QD | MeQSum | MeQSum |
| LR | 0.01 | 0.01 | 1e-4 |
| Batch # | 25 | 25 | 1 |
| Training steps | - | - | 20 K |
| Beam size | 8 | 8 | 8 |
| Beam $\alpha$ | N/A | N/A | 0.8 |
| Max input | 155 | 155 | 512 |
| Max target | 6 | 35 | 64 |
| Min target | 1 | 6 | N/A |

Table 1: Hyperparameters used in the PG Network (Baseline) and PEGASUS model. In PG Networks, QD indicates Question Decomposition Dataset used to train the question focus model, MeQSum is used to train the abstractive summarization model. LR is Learning Rate. We stop training PG Networks when the loss score converges less than 0.1, where the number of epochs varies from 5K to 150K depends on the architecture of Neural Network or different input pre-processing. Thus the epoch number is omitted for PG Networks.

duce the variance of a single model by training multiple models with different parameters or dataset and then selecting the optimal result. This method is widely used in predictive models (Huang et al. 2020; Dang et al. 2020).

Our ensemble method generates multiple outputs by training our model numerous times on the same data. We vary the number of training steps and create one model trained for 80,000 steps, and another model trained for 150,000 steps. Due to the limitation of our resources, we do not further increase the number of steps. We consider the outputs of these two systems and select the optimal output based on Equation 1. We hypothesized that this would balance drawbacks caused by potentially over-fitting and under-fitting the training data.

$$Score = \alpha * Similarity(Focus, Y) + \beta * Similarity(X, Y) \quad (1)$$

As mentioned, Equation 1 is used to determine which generated output is optimal. This equation calculates the similarity between the generated output (question summary) and the given question and the generated output and the focus of the question. We do this because, in our error analysis, we found frequent problems where the generated text was syntactically and often factually correct but was the focus of the summary was incorrect. We set $\alpha$ and $\beta$ set as 0.5 to equally balance the importance of similarity between focus and question.

In Equation 1, the function $Similarity()$ measures the similarity between two strings. $X$ is given question, $Y$ is generated summary of given question $X$ and $Focus$ is Focus phrase extracted from the given question $X$. $\alpha$ and $\beta$ were used to impose the weight of each score. The Sum of $\alpha$ and $\beta$ is 1. We use the same method used in section 3.2.4. We use spaCy (Honnibal et al., 2020), a library for advanced natural language processing, which includes state-of-the-art neural network models for similarity measures and NER. To measure the similarities for our output, which determines the similarity by comparing word vectors, we used the "en_core_web_lg" model for the word vectors. This model has 684,830 unique vectors with 300 dimensions.

## 4 Results

All experiments are done on Google Colab Pro with Tesla V100 GPU, RAM 25.51 GB0, CPU of Intel(R) Xeon(R) (2.20GHz). Pointer Generator Networks training took up to 1 hour for both Focus Extraction and Abstractive Summarization. To fine-tune PEGASUS took 1.5 hours for 20K training steps, 12 hours for 80K steps, and 23 hours for 150K steps.

### 4.1 Performance of the Summarization

#### 4.1.1 Pre-processing Combination Testing

We train our models on the training dataset and report results on the provided validation dataset (Table 2). We withheld the test set from all model development and hyper-parameter tuning and report results in Table 3.

Accuracy is measured using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) measuring overlapping words between reference and summaries. We use ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L). R-L refers to the longest common subsequence-based ROUGE score, R-1 is 1-gram based, and R-2 is bi-gram based ROUGE score.

Among all pre-processing combinations, we found Simple1, Spell Check, De-identification, and Sentence Exclusion applied on both validation and training datasets produced the highest score across all ROUGE metrics. Pre-processing with PEGASUS output provides higher accuracy generally compared to pre-processing with PG Networks. We choose the PEGASUS model for abstractive summarization to generate output with the official

test dataset. During the manual evaluation, we found not only ROUGE scores are higher with PEGASUS, but also outputs of PG Networks tend to generate repetitive words within an output, while PEGASUS outputs mostly have grammatically correct form. Scores of all 11 experiments can be found in a Table 2.

#### 4.1.2 Official Results

Our highest-scoring submission produced an R2-F score of 11.14%. This submitted system consists of 3 steps: (1) pre-processing, which corrects misspelling words, removes sentences without biomedical/clinical related terms, (2) abstractive summarizing by PEGASUS with 150k training iteration, (3) post-processing where we ensembled the outputs of two systems together. System 1 was trained for 80K training steps, while system 2 was trained for 150K training steps. Our other submitted system performed only the first two steps. No ensemble method was used. As shown in Table 3, we see there is an increase in performance by ensembling the two outputs rather than relying on the output of a single model. We used both the training and validation dataset to train the models to generate the summaries for the test dataset.

### 4.2 Performance of Focus Extraction

### 4.3 Performance of Focus Extraction

We measure Focus Detection with precision, recall, and f-measure, and Focus Generation with ROUGE-1, ROUGE-2, and ROUGE-L. The exact scores are shown in Table 4. Experiments No 1, 2, and 3 are Focus Detection results, and No 4 and 5 are Focus Generation results. Model No.5 is Focus Generation using PG Network with duplicated term removal resulted in an accuracy of 85%. We choose the Focus Generation method over Focus Detection, even though Focus Detection accuracy is considerably high to avoid the possibility of not detecting any focus, which may occur for some questions if the NER technique were to be used. As mentioned previously, Focus Generation will always generate focus for every question.

## 5 Discussion and Future Work

We found many incorrect summaries with the wrong focus during the experiment of different combinations of pre-processing. For example, given input question "I have **chronic renal disease** and worry that **Magnesium silicofluoride** treatment of moth infestation of a large living room rug will be harmful to my health. If the rug is treated in house how long before any toxic fumes or skin contact would be a hazard .", PEGASUS generated output (a) and (b):

(a) What are the side effects of **silicofluoride** treatment?

(b) What is the treatment for **moth infestation of a rug**?

In the given question, we see that the person is concerned with the effect of Magnesium silicofluoride on individuals with chronic renal disease. Thus, the focus of the given question would be **chronic renal disease** and **Magnesium silicofluoride**. In contrast, both generated output (a) and (b) summaries are built on incorrect focus. We believe extracting the correct focus and studying how to incorporate the focus would improve accuracy. In this paper, we applied focus in the post-processing step. We ranked the output using the Equation 1, and then select the output with the highest score.

The limitation of Equation 1 is that the extracted focus may not be accurate. If the extracted focus is not correct, the ensemble model may choose a non-relevant output. For example, input "**hydroxychloroquine for rheumatoid arthritis**. Can you tell me if this medication that my doctor put me on could make me sweat profusely at the slightest little strenuous activity I'm also **methotrexate** 6 2.5 mg once a week ." gives the following answers:

(a) Can **hydroxychloroquine** and **methotrexate** be taken together?

(b) What are the dosage side effects and drug interactions for **rheumatoid arthritis**?

The question asks if the **hydroxychloroquine for rheumatoid arthritis** and **methotrexate** be taken together. The generated summary (a) shows a reasonably accurate answer. In contrast, the focus extraction model assumed **rheumatoid arthritis** to be a focus, which leads the model to choose summary (b) over (a).

Despite this limitation, our experiments showed performance improvements after applying focus detection and the ensemble method in post-processing. The post-processing effect is limited to the performance of the summarization model, accuracy of focus for each question, and the number of outputs from the summarization models. Due to time limitations, we use two outputs in the ensemble process, while typical ensemble learning

| No. | Preprocessing on Training Data | Preprocessing on Validation Data | R-1 | R-2 | R-L |
|---|---|---|---|---|---|
| 1 | Simple0 | - | 29.69 | 13.23 | 28.91 |
| 2 | - | Simple1 + Deid | 29.28 | 13.48 | 28.72 |
| 3 | Simple1 + Deid + SE | Simple1 + Deid | 31.03 | 14.46 | 29.26 |
| 4 | Simple1 + Deid + SE + Merge(Subject, Message) | Simple1 + Deid + Merge(Focus(FD), Question) | 28.35 | 11.69 | 26.96 |
| 5 | Simple1 + Deid + SE + Merge(Focus, Subject, Message) | Simple1 + Deid + Merge(Focus(FD), Question) | 27.47 | 11.25 | 26.31 |
| **6** | **Simple1 + Spell Check + Deid + SE** | * | **31.60** | **13.93** | **30.55** |
| 7 | Simple1 + Deid + SE + Merge(Focus(FG), Question) | * | 28.41 | 11.90 | 26.27 |
| 8 | Simple1 + Deid + SE + Merge(Focus(Gold), Question) | * | 26.26 | 10.65 | 25.51 |
| 9 | Simple1 + Spell Check + Deid + SE + Message Only | * | 28.93 | 12.47 | 27.83 |
| 10 | Simple1 + Spell Check + Deid + SE | * | 18.98 | 10.50 | 17.32 |
| 11 | Simple1 + Spell Check + Deid + SE + Message Only | * | 19.22 | 10.31 | 18.23 |

Table 2: Performance testing with validation dataset. '-' indicates no pre-processing technique applied, '*' indicates that the method applied for the training dataset was applied to validation data. SE is an abbreviation of Sentence Exclusion. Deid is De-identification. Merge() is concatenating strings. FD is Focus Detection, and FG is focus generation. Experiments 1-9 are done with PEGASUS, and 10-11 are done with PG Networks.

| No. | Method | R1-P | R1-R | R1-F1 | R2-P | R2-R | R2-F1 | RL-R | RL-F1 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Pre-processing + PEGASUS (150K) | **0.321** | 0.285 | 0.283 | 0.120 | 0.105 | 0.106 | 0.257 | 0.257 |
| 2 | **Pre-processing + PEGASUS (150K & 80K) + Ensemble** | 0.315 | **0.291** | **0.284** | **0.123** | **0.112** | **0.111** | **0.265** | **0.259** |

Table 3: Performance testing with official test dataset. Experiment 1 output is with 150K training epochs. Experiment 2 were ensemble of outputs of the model trained for 80K epochs and the model trained for 150K epochs.

| No. | FD Method | P | R | F |
|---|---|---|---|---|
| 1 | GRU + CNN | 75.56 | 81.13 | 78.24 |
| 2 | RNN + CNN | 67.89 | 64.21 | 66 |
| 3 | **LSTM + CNN** | **78.79** | **82.21** | **80.47** |
| **No.** | **FG Method** | **R-1** | **R-2** | **R-L** |
| 4 | PG | 0.64 | 0.37 | 0.63 |
| 5 | **PG-duplicates** | **0.85** | **0.58** | **0.84** |

Table 4: Performance of Focus Extraction. Experiment No 1, 2, 3 are the results of Focus Detection and Experiment No 4 and 5 are results of Focus Generation. PG is short for PG Network and PG-duplicates is PG Network with removal of duplicated terms in generated output.

models use considerably larger numbers than 2. Thus, we believe there is a significant potential improvement by investigating: 1) Methods to generate more trained models with different parameters and datasets 2) Method to generate multiple models with less training time 3) Method to increase the performance of the focus extraction 4) Develop better methods for incorporating question focus information into the summary generation system. The current ensemble method is applied at a fairly late stage in the process. Study the effect of incorporating ensembling as early as the training step is an area of exploration.

## 6 Conclusion

In this paper, we present our Question Summarization system for Consumer Health Questions(CHQ). We explored effect of multiple pre-processing methods (De-Identification, Sentence Exclusion and Focus Extraction) and on state-of-art Abstractive Summarization. Our results show the best F-measure score of 11.14% through applying Ensemble Learning to different combinations of the pre-processing outputs. In our analysis, we identified future directions, including investigating the use of Extracted Focus, Ensemble Learning for the generative model.

## References

Asma Ben Abacha and Dina Demner-Fushman. 2019a. On the role of question summarization and information source restriction in consumer health question answering. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2019:117–126.

Asma Ben Abacha and Dina Demner-Fushman. 2019b. 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. Association for Computational Linguistics.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *CoRR*, abs/1901.08079.

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

José Camacho-Collados and Mohammad Taher Pilehvar. 2017. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *CoRR*, abs/1707.01780.

Jason P. C. Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *CoRR*, abs/1511.08308.

Huong Dang, Kahyun Lee, Sam Henry, and Özlem Uzuner. 2020. Ensemble BERT for classifying medication-mentioning tweets. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 37–41, Barcelona, Spain (Online). Association for Computational Linguistics.

Dina Demner-Fushman, Yassine Mrabet, and Asma Ben Abacha. 2019. Consumer health information and question answering: helping consumers find answers to their health-related information needs. *Journal of the American Medical Informatics Association : JAMIA*, 27.

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

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

Tongwen Huang, Qingyun She, and Junlin Zhang. 2020. Boostingbert:integrating multi-class boosting into bert for nlp tasks.

Fatemah Husain, Jooyeon Lee, Sam Henry, and Ozlem Uzuner. 2020. SalamNET at SemEval-2020 task 12: Deep learning approach for Arabic offensive language detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2133–2139, Barcelona (online). International Committee for Computational Linguistics.

Chandra Khatri, Gyanit Singh, and Nish Parikh. 2018. Abstractive and extractive text summarization using document context vector and recurrent neural networks. *CoRR*, abs/1807.08000.

Veysel Kocaman and David Talby. 2021. Spark nlp: Natural language understanding at scale.

Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. 2019. Abstractive text summarization based on deep learning and semantic content generalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5092, Florence, Italy. Association for Computational Linguistics.

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

Microsoft. 2016. Bing spell check.

J. Niu, H. Chen, Q. Zhao, L. Su, and M. Atiquzzaman. 2017. Multi-document abstractive summarization using chunk-graph and recurrent neural network. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6.

Kirk Roberts, Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. 2014. Decomposing consumer health questions. In *Proceedings of BioNLP 2014*, pages 29–37, Baltimore, Maryland. Association for Computational Linguistics.

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

Amber Stubbs and Özlem Uzuner. 2015. Annotating risk factors for heart disease in clinical narratives for diabetic patients. *Journal of biomedical informatics*, 58 Suppl:S78—91.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2017. Pointer networks.

Zhang Yijia, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong lu. 2019. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific Data*, 6.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D. Manning, and Curtis P. Langlotz. 2020. Biomedical and clinical english model packages in the stanza python nlp library. *arXiv preprint arXiv:2007.14640*.