Summarizing Lengthy Questions

Tatsuya Ishigaki **Hiroya Takamura** **Manabu Okumura**
Tokyo Institute of Technology
ishigaki@lr.pi.titech.ac.jp, {takamura, oku}@pi.titech.ac.jp

Abstract

In this research, we propose the task of *question summarization*. We first analyzed question-summary pairs extracted from a Community Question Answering (CQA) site, and found that a proportion of questions cannot be summarized by extractive approaches but requires abstractive approaches. We created a dataset by regarding the question-title pairs posted on the CQA site as question-summary pairs. By using the data, we trained extractive and abstractive summarization models, and compared them based on ROUGE scores and manual evaluations. Our experimental results show an abstractive method using an encoder-decoder model with a copying mechanism achieves better scores for both ROUGE-2 F-measure and the evaluations by human judges.

1 Introduction

Questions are asked in many situations, such as conference sessions and email communications. However, questions can sometimes be lengthy and hard to understand, because they often contain peripheral information in addition to the main focus of the question. To address this issue, we propose the task of *question summarization*; summarizing a lengthy question into a simple question that concisely represents the original content.

As an example of an excerpt from a Community Question Answering (CQA) site, Yahoo Answers[1], is shown in Table 1. In this example, the gist of the question is whether the chlorine will stripe the questioner's hair. However, the question also contains the additional information that the questioner swims five days a week and has black hair. Although such information can sometimes be important for finding the exact answer, it is often peripheral when we want to grasp what is being asked.

Table 1: Example of question-summary pair

Question Text:
I'm a swimmer for my school swim team and I practice two hours a day, five days a week. I would like to dye my hair black (it is dark brown now) but I am wondering whether the chlorine will stripe it. Will it or will it not ?
Summary:
Will the chlorine stripe my hair ?

Summarizing a question, which can often be lengthy, helps respondents understand the question. The task of question summarization has not been studied yet and is worth being explored. In this work, we focus on a CQA site and examine the characteristics of the question summarization task with a CQA dataset as a case study. Specifically, we first examine CQA data consisting of pairs of a question text and its title, which we refer to as "question-title" pairs. We then propose a method for creating pairs of a question and its summary, which we refer to as "question-summary" pairs, out of the CQA data. We also propose methods for question summarization and describe an empirical evaluation we conducted.

Approaches used in generic summarization tasks are often classified into two different types: extractive and abstractive. Extractive approaches select and order units, which are usually sentences or words, from the input text. Abstractive approaches, rather than selecting units, generate a summary using words not found in the input text.

However, existing summarization approaches, whether extractive or abstractive, do not assume

[1] https://answers.yahoo.com/

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 792–800,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

a question as an input. We therefore developed a number of methods designed for question summarization: extractive methods based on simple heuristic rules, extractive methods based on sentence classification/regression, and abstractive methods based on neural networks. We compared the performance of these methods through evaluations both by human judges and automatic scoring using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004). The experimental results show that an abstractive approach using an encoder-decoder model with a copying mechanism achieves the highest score for both ROUGE and evaluations by human judges.

2 Related Work

Text summarization is one of the problems that have been studied for a long time in the field of natural language processing. In many of the existing summarization tasks including the shared tasks in Document Understanding Conference (DUC)[2], documents from newspapers or scientific articles are considered as an input. There are also other summarization tasks in which other types of input are assumed such as conversations or email threads (Duboue, 2012; Oya and Carenini, 2014; Oya et al., 2014). Unlike the researches, we assume a question as an input.

As a related attempt in Question Answering researches, Tamura et al. (2005) worked on classification of multiple-sentence questions into classes such as yes/no questions and definition questions, and attempted to extract the question sentence that was the most important in finding the correct class. However, the extracted question sentence is not always a summary of the question. Consider the last sentence "Wiil it or will it not?" for the aforementioned Table 1 question. This last sentence is important in finding the class of this question, which is a yes/no question, but is not appropriate as a summary, because it is impossible to understand what is being asked merely from "Will it or will it not?"

Many existing extractive approaches select the sentences to be included in the summary on the basis of calculated scores for each sentence. The scores are often calculated by the TF-IDF measure (Luhn, 1958), the similarity measure between sentences (Mihalcea and Tarau, 2004) and an approximation made by a regression model of the ROUGE score or the bigram frequency included in a summary (Peyrard and Eckale-Kohler, 2016; Li et al., 2013). Other approaches consider summarization tasks as a classification problem. They adopt supervised machine learning techniques to solve them (Hirao et al., 2002; Shen et al., 2007).

Abstractive summarization approaches include methods based on syntactic transduction (Dorr et al., 2003; Zajic et al., 2004) or statistical machine translation models (Bank et al., 2010; Wubben et al., 2012; Cohn and Lapata, 2013) and templates (Oya et al., 2014). In addition, encoder-decoder approaches have been proposed in recent years. They were originally applied to machine translation tasks (Luong et al., 2015; Bahdanau et al., 2015; Cao et al., 2017), and have been actively applied to other sequence-to-sequence tasks including sentence summarization (Rush et al., 2015; Kikuchi et al., 2016; Gu et al., 2016).

3 Data Analysis

We first carried out an analysis on the dataset provided by Yahoo! Answers, "Yahoo! Answers Comprehensive Question and Answers version 1.0"[3]. The data contains 4,484,032 question-title pairs posted between June 28, 2005 and October 25, 2007. On the CQA site, users can freely write a question text and its title. Thus, some of the pairs in the data can be regarded as question-summary pairs, but some others cannot. To obtain a dataset that can be used for training, we need to filter out the pairs that are not suitable for our objective. Furthermore, it is not clear how questions are summarized, i.e., whether by extractive or abstractive methods. Therefore, in this study, we first analyzed the data to clarify the following issues:

1. What are the characteristics of the pairs that cannot be regarded as question-summary pairs?

2. Can an extractive approach generate a summary equivalent to the title, or are abstractive approaches required?

3.1 Question text length

To characterize the question-title pairs that could not be regarded as question-summary pairs, we first focus on the number of sentences in question text. We randomly extracted the question-title

[2] http://duc.nist.gov

[3] https://webscope.sandbox.yahoo.com/

Table 2: Number of sentences in question text and proportion of question-summary pairs

No. of sentences in question text	Proportion
1	3/20
2	8/20
3	14/20
4	15/20
5	15/20

Table 3: Number of titles generated by extractive and abstractive approaches

Not a question-summary pair	5/20
Extractive approach can summarize	8/20
Abstractive approach needed	7/20

pairs that contained 1-5 sentences in the question text, and manually classified them as to whether they could be regarded as a question-summary pair or not. Table 2 shows the number of question-title pairs that can be regarded as question-summary pairs for question text size measured by the number of sentences.

This analysis showed that if there are two or fewer sentences in the question text, the pairs are unlikely to be question-summary pairs because the question texts tend to contain only peripheral information to support the question presented in the title. In contrast, if there are three or more sentences, the proportion of the question-summary pairs becomes high and substantially constant. This suggests that the number of sentences in a question text is one of the clues to find question-summary pairs.

3.2 Nouns overlapping between question and title

We show here an example that cannot be regarded as a question-summary pair.

> Title:
>
> Why is there often a mirror in an elevator?
>
> Question text:
>
> I just realized this when I was in an elevator. Does anybody know the reason? What is the history behind it?

This is not a question-summary pair because the question text does not express the content of the title. In such cases, people cannot grasp the gist of the question text when only the title is presented. Here we focus on the words “mirror” and “elevator”; they appear in the title, but not in the question text. We actually observed many similar instances. This suggests that noun overlapping between a question and its title can be considered an important clue to determine whether the pair can be regarded as a question-summary pair or not.

3.3 Extractive vs. abstractive

We next analyzed question-summary pairs in terms of whether the summaries can be generated by an extractive method or an abstractive method is required. Specifically, we randomly selected pairs whose question text had 3-5 sentences and manually classified them into one of the following 3 categories: 1) The pair cannot be regarded as a question-summary pair, 2) An extractive method can generate a summary that is equivalent to the title, and 3) Others (i.e., an abstractive method might be needed to generate a summary that is equivalent to the title).

The manual classification results are shown in Table 3. We also show representative examples of question-summary pairs in Table 4.

Five out of 20 cases cannot be regarded as question-summary pairs. In Example 1 of Table 4, the questioner accidentally spilled buttered popcorn and needs to know how to remove it. However, the title “Please help!” does not contain enough information to grasp the gist of the question. In some cases, pronouns in the titles refer to nouns in the question text.

In eight of the 20 pairs, an extractive approach can generate a summary that is almost equivalent to the title. Example 2 in Table 4 is such a case. In this example, a summary can be generated by extracting the last sentence in the original content. However, if one takes the actual title into consideration, the idiom “get rid of” in the original question can be replaced by the word “remove”. Even if a question text can be summarized by an extractive method, the actual titles are often generated by abstractive approaches.

In the remaining seven pairs, the question texts cannot be summarized by an extractive approach, and abstractive approaches might be required. The category is further split into two subcategories. In the first subcategory, pronoun resolution as well as sentence extraction is needed to generate a summary. In the second subcategory, a short question

Table 4: Representative examples of pairs in Yahoo! Answers dataset

Example 1 (The title does not express the content of the question text): I accidentally spilled buttered popcorn on my leather hospital shoe. It has dark spots on it now and I don't know how i can get them off. ... Title: Please help!
Example 2 (Extractive approaches can be applied.): I keep getting annoying Winfixer Pop Ups . I have tried all sorts of ad removal programs to get rid of them but without success . How can I get rid of them ? Title: How can one remove annoying pop ups ?
Example 3 (Abstractive approaches are required.): "The Simpsons" is one of the funniest shows ever . It's one of my favorites . Do you like it ? Title: Do you like "The Simpsons" ?
Example 4 (Abstractive approaches are required.): I want my chocolate chip cookies to be thicker and kind of gooey–crispy outside, chewy inside. I've experimented with various recipes and various oven temperature, but my cookies always turn out thin and flat. Why? What am I doing wrong? Title: Why do my chocolate chip cookies always turn out thin and flat?

follows a lengthy explanation.

In Example 3 in Table 4, the pronoun "it" in the main question "Do you like it?" refers to the program name "the simpsons", which appeared previously. Therefore, the summary does not contain enough information to grasp the gist if we naively apply any extractive approaches. The short main question "why" in Example 4 follows the explanation about baking cookies. In examples such as these, we cannot naively use extractive approaches to generate an understandable summary, because the title is generated by picking up the information from multiple sentences.

4 Dataset and Methodology

In this section, we describe how we created a dataset consisting of question-summary pairs, and a number of methods for question summarization.

4.1 Dataset

As training data for extractive and abstractive models, we use question-title pairs posted to a CQA site, namely, "Yahoo! Answers Comprehensive Question and Answers version 1.0". The original data contains 4,485,032 question-title pairs. However, not all of them are question-summary pairs. To filter out the pairs that are not question-summary pairs, we removed the pairs that match at least one of the following conditions:

Multiple sentences in the title Comprising two or more sentences

Long Title The title consists of over 16 words.

Short Title The title consists of three or less words.

Overlap of nouns
No nouns in the title appear in the question text.

Short Question Text
The question text consists of two or less sentences.

Long Question Text
The question text consists of over five sentences.

After applying the filtering, we obtained 251,420 pairs[4]. We use the pairs for training extractive and abstractive models, and also for evaluations.

[4] Note that the data still contains non-English question-title pairs, since the language recognition is not perfect.

4.2 Extractive approaches

As extractive approaches, we adopted rule-based approaches and machine learning based approaches.

4.2.1 Rule-based approaches

As rule-based approaches, we used three rules to compare: “Lead Sentence”, “Lead Question”, and “Last Question”. The first sentence presented (Lead Sentence) is known as a strong baseline for generic summarization tasks. However, in the question summarization, the summaries should be also questions. Therefore, we adopted methods to select a question in the input by heuristic rules, choosing the first question (Lead Question) and the last question (Last Question). A sentence was determined to be a question if the last character is “?” or the first word is an interrogative word.

4.2.2 Machine learning-based approaches

We will here introduce two types of machine learning-based methods: a classification-based method and a regression-based method.

The regression-based model predicts the ROUGE-2 F-measure score for each sentence in the input question. After the prediction step finishes, the model outputs the sentence with the highest predicted ROUGE score. To train the regression model, we first calculated the ROUGE score for each sentence in the training set, regarding the title as a reference summary. After the calculation, we trained Support Vector Regression (SVR) (Basak et al., 2007).

The classification-based method predicts the sentence with the largest ROUGE-2 F-measure. In the training phase, we regarded the questions which had the highest ROUGE score and consisted of at least four words as positive instances. Other questions were used as negative instances. We adopted Support Vector Machine (SVM) (Suykens and Vandewalle, 1999) as a classifier. If the SVM classifies more than one sentence in a single input document as positive, then our method outputs the first question. In contrast, if the SVM classifies all questions in the input as negative, then the model outputs the first question. It outputs the first sentence if there is no question in the output. A sentence is regarded as a question if the last character is “?” or the first word is an interrogative word.

We trained the regression and classification models by using the following features:

- Word unigram
- Sentence length
- Whether the sentence is the initial sentence
- Whether the sentence is the first question
- Presence of other question sentences

All features are expressed as binary: i.e., 0 or 1. For unigram features, we adopted features that appeared at least five times in the training set. We used four sentence-lengthy features: the sentence had less than three, less than five, more than 10, and more than 15 unigrams.

4.3 Abstractive approaches

We adopted encoder-decoder based methods for abstractive approaches. Specifically, we trained three models: a vanilla encoder-decoder model, an encoder-decoder model with an attention mechanism and an encoder-decoder model with a copying mechanism.

Questions in the CQA site are usually composed of 3-5 sentences, which are longer than in the usual settings used in machine translations tasks. Therefore, in addition to the vanilla model, we trained a model with the attention mechanism (Luong et al., 2015). To reduce the model size, we replaced low-frequency words with the special token UNK, which is a well-known technique. After the preprocessing, the vocabulary size was reduced to approximately 136,000. In summarization tasks, the words in the input are often likely to appear in the summary. Therefore, we also adopted the encoder-decoder model with the copying mechanism (Gu et al., 2016), which can select words in the input as words in the output.

We briefly describe those models below.

4.3.1 Vanilla encoder-decoder

The encoder-decoder model is composed of two elements: an encoder and a decoder. The encoder receives the input question $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, and converts it into the fixed-length continuous value vector $\boldsymbol{h}_\tau$:

$$\boldsymbol{h}_\tau = f(x_\tau, \boldsymbol{h}_{\tau-1}), \tag{1}$$

where f represents an activation function used in any Recurrent Neural Network (RNN). In this study, the vanilla encoder-decoder and the attention models use Long Short-Term Memory (Hochreiter and Schmidhuber, 1997), and Gated

Recurrent Unit (GRU) (Cho et al., 2014) is used in the model with the copying mechanism. All encoder-decoder models adopt bidirectional-RNNs as the encoder.

The decoder receives the last hidden state of the encoder and generates a sentence. Each node in the decoder receives the previously generated word and the hidden state generated in the previous time step to calculate the hidden state $\boldsymbol{s}_t$ and the $softmax$ function. The occurrence probability of y_t is calculated by using the hidden state $\boldsymbol{s}_t$ and a $softmax$ function:

$$\boldsymbol{s}_t = f(y_{t-1}, s_{t-1}), \tag{2}$$

$$p(y_t|y_{<t}, \boldsymbol{x}) = softmax(g(\boldsymbol{s}_t)). \tag{3}$$

The conditional probability given the input sequence $\boldsymbol{x}$ can be decomposed to the product of the probabilities of generating words as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{m} p(y_t|y_{<t}, \boldsymbol{x}). \tag{4}$$

In the training, we estimated the parameter values so that they maximize the log-likelihood of the training set:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \log \sum_{j=1}^{m} p(y_t|y_{<t}, \boldsymbol{x}). \tag{5}$$

In the test phase, the model generates the output by beam-search.

4.3.2 Encoder-decoder with attention

In the vanilla encoder-decoder model, the input document is encoded into the hidden state $\boldsymbol{h}_n$. The decoder receives the hidden state as the initial state $\boldsymbol{s}_0$ of the decoder ($s_0 = \boldsymbol{h}_n$). In contrast, the encoder-decoder model with the attention mechanism uses the context vector $\boldsymbol{c}_t$ represented as a weighted sum of the hidden states of the encoder:

$$\boldsymbol{c_t} = \sum_{\tau=1}^{n} \alpha_{t\tau} \boldsymbol{h}_\tau, \tag{6}$$

where $\alpha_{t\tau}$ is the weight of the t-th word of the input at time step τ and can be calculated as

$$\alpha_{t\tau} = \frac{exp(\boldsymbol{s}_t \cdot \boldsymbol{h}_\tau)}{\sum_{h'} exp(\boldsymbol{s_t} \cdot \boldsymbol{h'})}. \tag{7}$$

Finally, the conditional probability of the word y_t is calculated by the $softmax$ function:

$$\tilde{\boldsymbol{h}} = tanh(\boldsymbol{W_c}[\boldsymbol{c_t}; \boldsymbol{h_t}]), \tag{8}$$

$$p(y_t|y_{<t}, \boldsymbol{x}) = softmax(\boldsymbol{W_s} \tilde{\boldsymbol{h}}_t). \tag{9}$$

Table 5: Evaluation on ROUGE-2

	Recall	F-measure
Lead	39.4	27.0
Last-Q	42.6	33.9
Lead-Q	45.3	34.5
Classification	44.3	35.1
Regression	44.7	29.7
EncDec	3.5	2.6
EncDec+Attn	38.5	38.5
CopyNet	47.4	42.2

4.3.3 Encoder-decoder with copying mechanism

As the encoder-decoder model with the copying mechanism, we used the model proposed by Gu et al. (2016). In the model, the decoder calculates the probability of generating y_t at time step t by using a mixed probabilistic model of two modes: the generate-mode and the copy-mode:

$$p(y_t|y_{<t}, \boldsymbol{x}) = p_{gen}(y_t|\boldsymbol{s}_t, y_{t-1}, \boldsymbol{c}_t, \boldsymbol{x}) + p_{copy}(y_t|\boldsymbol{s}_t, y_{t-1}, \boldsymbol{c}_t, \boldsymbol{x}), \tag{10}$$

where p_{gen} is the probability calculated by the generate-mode using the same scoring function proposed by Bahdanau et al. (2015) and p_{copy} is the probability that the copy-mode will “copy” the word y_t from the input document if $y_t \in \boldsymbol{x}$. If $y_t \notin \boldsymbol{x}$, then p_{copy} is set to 0. Thus, the model increases the probability that words in the input will be generated. Refer to the original paper by Gu et al., (2016) for more detailed explanations.

5 Experiments and Evaluation

We adopted ROUGE-2 (Lin, 2004) as a metric for the automatic evaluation. Additionally, we carried out an evaluation on a 5-point scale scored by human judges. In this section, we will describe the details of model training, automatic and manual evaluations we conducted.

5.1 Experimental setting and training

The created dataset contained 251,420 pairs. We used 90% of the data for training. We separate the remaining equally for the development and the test set. Thus, we split the dataset into 18 (train):1 (development):1 (test).

As an implementation of SVM and SVR, we used Liblinear (Fan et al., 2008). The linear kernel was used as the kernel function for SVM and

SVR. We tuned the regularization parameter C on the development set.

For the encoder-decoder model, we adopted 256 dimensions for word embedding and hidden layers, setting the batch size to 64. The words that appeared at least twice in the training set were used in training, and other words were replaced by the special token UNK. The end of a sentence was represented by another special token, i.e., EOS. For testing, we used the model which achieves the minimum loss function in the development set. When the encoder-decoder model does not output the EOS token within 20 words in the decoding step, the model outputs the first question in the input text. If there is no question in the input, the first sentence is output.

5.2 Evaluation with ROUGE

In our task setting, the number of sentences in the output was limited to one. There was no length constraint in terms of the number of characters or words. However, we assumed that a better summary would contains more focused content in a shorter output. Therefore, as the evaluation metric, we adopted the ROUGE-2 F-measure in addition to Recall.

Table 5 shows ROUGE-2 scores for a number of methods. Lead method (Lead), Last Question (Last-Q) and Lead Question (Lead-Q) are rule-based methods. The classification-based model (Classification) and regression-based model (Regression) are non-neural machine-learning based methods. Vanilla encoder-decoder (EncDec), encoder-decoder with an attention mechanism (EncDec+Attn) and encoder-decoder with a copying mechanism (CopyNet) are neural-network based methods.

In rule-based extractive methods, the lead method, which simply outputs the first sentence, is known as a strong baseline. However, in question summarization, selecting questions such as Lead Question or Last Question increases the ROUGE score. Lead Question is a strong baseline in particular.

Classification was as good as Lead-Q, because most input texts contained only one to two question sentences; as a result, the two methods mostly output the same results. The encoder-decoder models with an attention and with a copying mechanism achieved a significantly higher ROUGE score than the extractive approaches. Note that the vanilla encoder-decoder model yielded significantly low ROUGE score, because it generated mostly the same question for all input texts. The input sequences in this task were longer than those in machine translation. As Loung et al. (2015) mentioned, encoder-decoders models without an attention do not work well for long sentences. Therefore, the model failed to decode the sequence. On average, outputs of extractive methods are longer than those of abstractive methods. This accounts for the relatively low F-measure and competitive recall obtained with extractive methods.

5.3 Manual evaluation

Since our work is the first attempt to address the task, no other annotated data exists. To make up for this, we adopted manual evaluation in addition to the evaluation using ROUGE scores. The manual evaluation was performed using the "Crowdflower", which is a crowdsourcing service[5].

The evaluators were presented with a question and four summaries from different models: Human, and the best model in each group, Lead-Q, Classification and CopyNet. They were asked to rate each summary on 1-5 scale: very poor(1), poor(2), acceptable(3), good(4) and very good(5). The evaluation criteria were "grammaticality" and "focus", which are based on the criteria used in DUC. We asked the evaluators to give a higher score for the aspect of "focus" if a summary expressed the main focus of the input text. We also asked them to give a high "grammaticality" score to a grammatical summary. To control the quality of the evaluation, we randomly presented clearly ungrammatical and non-focused summaries as test to all evaluators, and excluded the the evaluations by evaluators who failed the test questions. The data for the manual evaluation consists of 100 randomly selected instances from the test set for the automatic evaluation. Each instance was evaluated by 3 evaluators.

The results on "grammaticality" and "focus" are respectively shown in Tables 6 and 7. The tables show the number of times each method on a row was evaluated higher than another method on a column. Evaluations on both criteria showed a trend similar to that of the automatic evaluation: Human achieved the highest score, Lead-Q and Classification were competitive, and CopyNet got

[5] https://www.crowdflower.com

Table 6: Human Evaluation-Focus-

	Human	Lead-Q	Classification	CopyNet
Human	-	135	135	103
Lead-Q	69	-	11	72
Classification	70	10	-	68
CopyNet	89	107	103	-

Table 7: Human Evaluation - Grammaticality -

	Human	Lead-Q	Classification	CopyNet
Human	-	85	86	63
Lead-Q	51	-	10	54
Classification	54	10	-	53
CopyNet	69	79	82	-

Table 8: Example outputs from each model.

Question Text
The Simpsons is one of the funniest shows ever . its one of my favorites . do you like it ?
Human : Do you like The Simpsons?
Lead-Q : Do you like it?
Classification : Do you like it?
EncDec+Attn: Do you like UNK?
CopyNet : Do you like The Simpsons?

the better score than Lead-Q and Classification.

CopyNet was judged better than Human in terms of focus in 89 cases, and in terms of grammaticality in 69 cases, because Human sometimes removes specific information. For example, CopyNet generated "How do you stop the itching after shaving?", while Human summary omits "after shaving". In terms of grammaticality, some Human summaries are not complete sentences such as "The best way to get money?". Therefore, Human was sometimes judged lower than CopyNet.

5.4 Qualitative analysis

In this section, we review the outputs of each model. Table 8 shows examples of each model.

The outputs of Lead-Q and Classification, which are generated by extraction, include the unresolved pronoun "it". This makes the summary not clearly focused. Such cases are often seen in lengthy questions that contain long supplementary explanations followed by a short question. These examples suggest that extractive approaches are intrinsically not suitable for cases where information needs to be picked up from multiple sentences in the input. In contrast to extractive approaches, the output of CopyNet properly resolves this; "it" is resolved by "The Simpsons" even if the model needed to use information across sentences. The EncDec+Attn model faces the difficulty in generating low frequency words such as "The Simpsons"; its output includes the special token UNK. This problem was also reported in other papers on encoder-decoder models (Bahdanau et al., 2015). Adding the copying mechanism effectively solved the problem.

6 Conclusion

We proposed a novel task of summarizing lengthy questions into simple questions that clearly express the focus of the original content. We created a dataset by filtering out inappropriate instances from a dataset provided by a CQA site, and developed extractive/abstractive models. Our results show that abstractive approaches outperform extractive approaches both in automatic and human evaluations. Since all the methods were inferior to the Human method in terms of performance, we believe there is still room for improvement. As a subject for future work, we will extend the approach to cover question summarization tasks that have multiple focuses. We are also interested in how existing analyzers such as coreference resolvers can improve the performance.

Acknowledgment

This work was supported by JST PRESTO (Grant Number JPMJPR1655).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR2015*.

Michele Bank, Vibhu O. Mittal, and Michael J. Witbrock. 2010. Headline generation based on statistical translation. In *Proceedings of ACL2010*. pages 318–325.

Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabis. 2007. Support vector regression. *Neural Information Processing-Letters and Reviews* 11(10):203–224.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Proceedings of AAAI17*. pages 3152–3158.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP2014*. pages 1724–1734.

Trevor Cohn and Mirella Lapata. 2013. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology (TIST)* 4(3):41.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of HLT-NAACL03 Text Summarization Workshop*. pages 1–8.

Pablo Ariel Duboue. 2012. Extractive email thread summarization: Can we do better than he said she said? In *Proceedings of INLG2012*. pages 85–89.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research* .

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL2016*.

Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of EMNLP2016*. pages 1328–1338.

Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of ACL2013*. pages 1004–1013.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL2004 Workshop*. pages 74–81.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.* 2(2):159–165.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP2015*. pages 1412–1421.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bridging order into texts. In *Proceedings of EMNLP2004*. pages 404–411.

Tatsuro Oya and Giuseppe Carenini. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probalistic approach. In *Proceedings of SIGDIAL2014*. pages 133–140.

Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of INLG2014*. pages 45–53.

Maxime Peyrard and Judith Eckale-Kohler. 2016. Optimizing an approximation of rouge- a problem-reduction approach to extractive multi-document summarization. In *Proceedings of ACL2016*. pages 1825–1836.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of EMNLP2015*. pages 379–389.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document summarization using conditional random fields. In *IJCAI-07*. pages 2862–2867.

Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9(3):293–300.

Akihiro Tamura, Hiroya Takamura, and Manabu Okumura. 2005. Classification of multiple-sentence questions. In *Proceedings of IJCNLP-05*. pages 426–437.

Sander Wubben, Yansong Feng, and Mirella Lapata. 2012. Title generation with quasi-synchronous grammar. In *Proceedings of ACL2012*.

David Zajic, Bonnie J Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004. In *Proceedings of NAACL-HLT04 Document Understanding Workshop*. pages 112–119.