Extractive Headline Generation Based on Learning to Rank for Community Question Answering

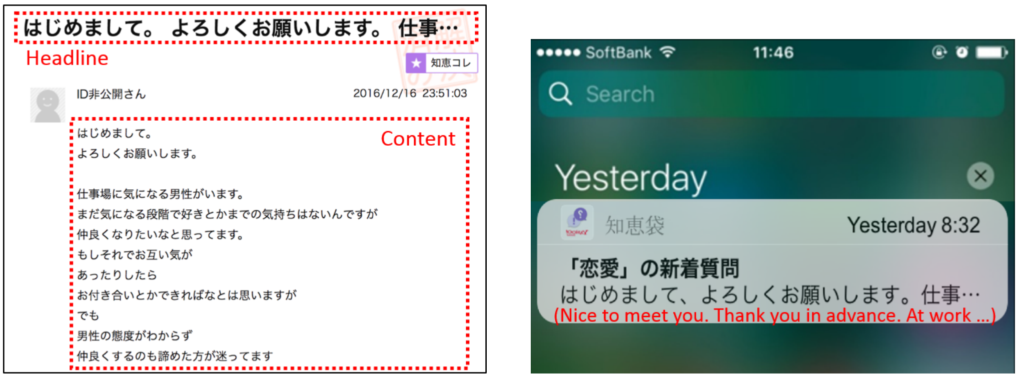
Tatsuru Higurashi[†] **Hayato Kobayashi**[†‡] **Takeshi Masuyama**[†] **Kazuma Murao**[†]
[†]Yahoo Japan Corporation [‡]RIKEN AIP
{thiguras,hakobaya,tamasuya,kmurao}@yahoo-corp.jp

Abstract

User-generated content such as the questions on community question answering (CQA) forums does not always come with appropriate headlines, in contrast to the news articles used in various headline generation tasks. In such cases, we cannot use paired supervised data, e.g., pairs of articles and headlines, to learn a headline generation model. To overcome this problem, we propose an extractive headline generation method based on learning to rank for CQA that extracts the most informative substring from each question as its headline. Experimental results show that our method outperforms several baselines, including a prefix-based method, which is widely used in real services.

1 Introduction

Community question answering (CQA) is a service where users can post their questions and answer the questions from other users. The quality of a CQA service is inevitably linked to how many questions are answered in a short amount of time. To this end, the headline of a posted question plays a key role, as it is the first thing users see in a question list or push notification on a smartphone. However, questions in a CQA service do not always have appropriate headlines because the questions are written by various users who basically do not have any specialized knowledge in terms of writing such content, in contrast to news articles written by professional editors. In fact, the biggest CQA service in Japan, Yahoo! Chiebukuro[1], does not even provide an input field for headlines in the submission form of questions, as general users do not have enough patience and tend not to post questions if even just one required field is added to the form. This service alternatively uses the prefix of a question as its headline as in Figure 1(a), where the headline "はじめまして。よろしくお願いします。仕事…(*Nice to meet you. Thank you in advance. At work ...*)" is created from the prefix of the content. Obviously, this headline is uninformative because



はじめまして。 よろしくお願いします。 仕事…

ID非公開さん 2016/12/16 23:51:03

はじめまして。
よろしくお願いします。

仕事場に気になる男性がいます。
まだ気になる段階で好きとかまでの気持ちはないんですが
仲良くなりたいなと思ってます。
もしそれでお互い気が
あったりしたら
お付き合いとかできればなとは思いますが
でも
男性の態度がわからず
仲良くするのも諦めた方が迷ってます

(a) Example of posted question.

Yesterday 8:32

「恋愛」の新着質問
はじめまして、よろしくお願いします。仕事…
(Nice to meet you. Thank you in advance. At work ...)

(b) Example of push notification.

Figure 1: Examples of (a) posted question and (b) push notification.

[1] https://chiebukuro.yahoo.co.jp/

Proceedings of the 27th International Conference on Computational Linguistics, pages 1742–1753
Santa Fe, New Mexico, USA, August 20-26, 2018.

of the lack of actual content, which is related to an initial unrequited love of a woman in the workplace. Figure 1(b) shows how ineffective such an uninformative headline is for a smartphone push notification, where users would have practically zero motivation to click on it and answer since they cannot imagine what kind of question is being asked. This negative effect has been confirmed on a commercial CQA service (Section 2).

In this work, we take an extractive approach for improving uninformative headlines. Although there have recently been many studies on abstractive headline generation (as described in Section 6), we do not follow any of these approaches because the content we deal with has no correct headlines and also the abstractive methods can yield erroneous output. This latter issue is important in practical terms because correct output is critical for a commercial service. If by some chance an erroneous headline, or one including politically incorrect phrases, is sent to all users, the service's credibility can be lost instantly. Therefore, we formalize our task as an extraction problem of a fixed-length substring from a question as its headline. In this setting, we can assume that outputs never include such errors caused by the service, as the outputs are substrings of user-posted questions. Note that they have no coherence errors in selecting multiple sentences as in normal extractive summarization tasks. While it is true that these outputs might contain inappropriate expressions authored by users, this type of error is beyond our scope since it is a different inevitable problem. Furthermore, the situation of "the service generated an inappropriate expression by itself" is significantly worse than the situation of "a user posted an inappropriate question to the service, and the service displayed it". Therefore, it is difficult to directly use abstractive methods for commercial services from a business standpoint.

Our approach involves preparing headline candidates and ranking them. The formal description is as follows. Let q be a target question to be translated into a headline. We first prepare a set $S(q)$ of headline candidates from the question q. Note that the set $S(q)$ is restricted to a set of fixed-length substrings given a length n, i.e., $S(q) \subseteq \{x \mid x \preceq q, |x| = n\}$, where $x \preceq q$ means that x is a substring of q. Then we extract the best headline that maximizes a score function $f_q(x)$, which represents the "headline-ness" score of a candidate $x \in S(q)$ with respect to the target question q, as follows:

$$\mathrm{argmax}_{x \in S(q)}\, f_q(x). \tag{1}$$

To ensure simplicity of implementation and understandability from users, we use a set of the fixed-length prefixes of all sentences in a question q as the candidate set $S(q)$ (Section 3). Because the problem is to select the best candidate from among several targets, the score function $f_q(x)$ is naturally trained by learning to rank (Section 4).

The main contributions of this paper are as follows.

- We report empirical evidence of the negative effect of uninformative headlines on a commercial CQA service (Section 2). We additionally show that our task can reduce uninformative headlines by using simple dictionary matching, which dramatically improves the average answer rate of questions by as much as 2.4 times.
- We propose an extractive headline generation method based on learning to rank for CQA (Section 4) that extracts the most informative substring (prefix of mid-sentence) from each question as its headline. To the best of our knowledge, our work is the first attempt to address such a task from a practical standpoint, although there have been many related studies (Section 6). Experimental results show that our method outperforms several baselines, including the dictionary-based method (Section 5).
- We create a dataset for our headline generation task (Section 3), where headline candidates extracted from questions are ranked by crowdsourcing with respect to "headline-ness", that is, whether or not each headline candidate is appropriate for the headline of the corresponding question.

2 Negative Effect of Uninformative Headlines

We conducted A/B testing on the push notifications of smartphones in collaboration with Yahoo! Chiebukuro, as shown in Figure 1(b). We first prepared a dictionary of typical first sentences that cause uninformative headlines. This dictionary was manually selected from frequent first sentences and consists of 913 sentences including greetings such as "おはようございます (Good morning)", "こんにち

Changed uninformative	Unchanged uninformative	Informative
0.75%	0.31%	0.45%

Table 1: Average answer rates of three question groups in A/B testing.

は (Good afternoon)", and "はじめまして (Nice to meet you)", and fixed phrases such as "質問させてください (Can I ask you something)", "教えてください (Please tell me)", and "よろしくお願いします (Thank you in advance)". We assumed that a question with a prefix match in the dictionary has an uninformative prefix headline and classified such questions into an *uninformative group*. For convenience, we also classified the other questions into an *informative group*, although they might include not so informative headlines. We further randomly divided the uninformative group into two equal groups: *changed* and *unchanged*. In the changed uninformative group, each headline is extracted as the prefix of the first (informative) sentence that does not match with the dictionary, which is the same as DictDel explained in Section 5.2. The unchanged group remains in uninformative. For comparison of these groups, we used the average answer rate over notified questions in each group as an evaluation measure, defined as

$$\text{Average answer rate} = \frac{\text{No. of questions answered from the notification}}{\text{No. of notified questions}}. \quad (2)$$

Note that we use a percentage expression (%) for easy reading.

Table 1 shows the evaluation results of the A/B testing during a 1-month period (Feb. 2 – Mar. 4, 2018), where about three million questions were sent to users. Comparing the unchanged uninformative group with the informative group, we can see that the average answer rate of the uninformative questions, 0.31%, is actually lower than that of the informative questions, 0.45%. Comparing the changed and unchanged uninformative groups, the average answer rate of the changed questions, 0.75%, is much higher than that of the unchanged questions, 0.31%. This means that even a simple dictionary-based method can dramatically improve the quality of the uninformative headlines, i.e., by as much as 2.4 times. We confirmed that the difference is statistically significant on a one-tailed Wilcoxon signed-rank test ($p < 0.05$). Note that the average answer rate represents a conversion rate (or rate of target actions), which is more important than a click-through rate (or rate of initial actions). The average answer rate is one of the most important indicators for a CQA service, while the click-through rate can be meaninglessly high if the service sends headlines that are fake or too catchy. We should point out that low answer rates are sufficient for the service, since it has 44M users: i.e., each question has an average of 2.4 answers, as the service currently has 189M questions and 462M answers.

3 Dataset Creation

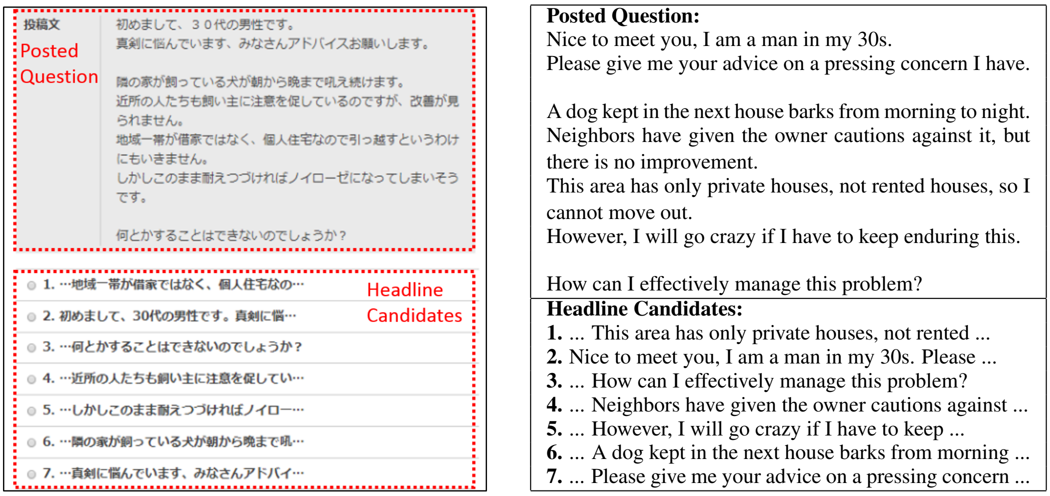
We created a dataset for our headline generation task based on the Yahoo! Chiebukuro dataset[2], which is a dataset including questions and answers provided from a Japanese CQA service, Yahoo! Chiebukuro. We first prepared headline candidates from this dataset as in Section 3.1 and then conducted a crowdsourcing task specified in Section 3.2. In Section 3.3, we report the results of the crowdsourcing task.

3.1 Preparation of Headline Candidates

We extracted only questions from the Chiebukuro dataset and split each question into sentences by using punctuation marks (i.e., the exclamation ("！"), question ("？"), and full stop ("。") marks). We regarded 20 Japanese characters that are basically extracted from each sentence as a headline candidate $x \in S(q)$ in Eq. (1), since this setting is used for push notifications in the actual service in Figure 1(b). More specifically, the headline candidate is created as follows:

1. If the sentence is the first one in the question, we extract the first 19 characters and put an ellipsis mark ("⋯") at the end.
2. Otherwise, we extract the first 18 characters and put ellipses at both the beginning and the end. This expression explicitly represents being mid-sentence so as to avoid weird headlines.

[2] http://www.nii.ac.jp/dsc/idr/en/yahoo/yahoo.html



投稿文

初めまして、３０代の男性です。
真剣に悩んでいます、みなさんアドバイスお願いします。

隣の家が飼っている犬が朝から晩まで吼え続けます。
近所の人たちも飼い主に注意を促しているのですが、改善が見られません。
地域一帯が借家ではなく、個人住宅なので引っ越すというわけにもいきません。
しかしこのまま耐えつづければノイローゼになってしまいそうです。

何とかすることはできないのでしょうか？

○ 1. …地域一帯が借家ではなく、個人住宅なの…
○ 2. 初めまして、30代の男性です。真剣に悩…
○ 3. …何とかすることはできないのでしょうか？
○ 4. …近所の人たちも飼い主に注意を促してい…
○ 5. …しかしこのまま耐えつづければノイロー…
○ 6. …隣の家が飼っている犬が朝から晩まで吼…
○ 7. …真剣に悩んでいます、みなさんアドバイ…

Posted Question:
Nice to meet you, I am a man in my 30s.
Please give me your advice on a pressing concern I have.

A dog kept in the next house barks from morning to night.
Neighbors have given the owner cautions against it, but there is no improvement.
This area has only private houses, not rented houses, so I cannot move out.
However, I will go crazy if I have to keep enduring this.

How can I effectively manage this problem?

Headline Candidates:
1. ... This area has only private houses, not rented ...
2. Nice to meet you, I am a man in my 30s. Please ...
3. ... How can I effectively manage this problem?
4. ... Neighbors have given the owner cautions against ...
5. ... However, I will go crazy if I have to keep ...
6. ... A dog kept in the next house barks from morning ...
7. ... Please give me your advice on a pressing concern ...

(a) Example of our crowdsourcing task.

(b) English translation of left example.

Figure 2: Examples of (a) our crowdsourcing task and (b) its English translation.

In the case where the length of a candidate is less than 20 characters, we include some of the next sentence in order to maximize use of display space. We included questions with more than five sentences for the purpose of efficiently collecting ranking information. All told, we prepared 10,000 questions containing more than five headline candidates each.

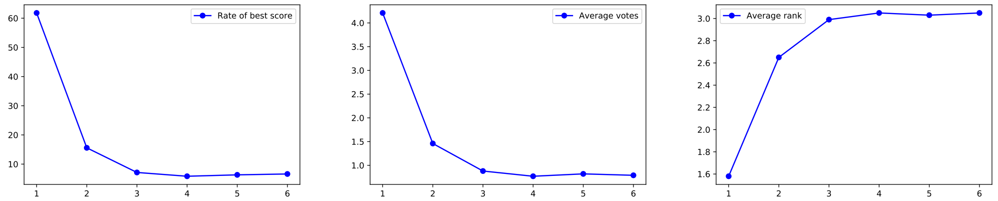
3.2 Crowdsourcing Task

Figure 2 shows an example of our crowdsourcing task (a) and its English translation (b). This task involves a posted question and headline candidates corresponding to the question. We asked workers to select the best candidate from options after reading the posted question. A relative evaluation, where workers select the best candidate, was used instead of an absolute evaluation, where workers select a score from 0 to 10 for each candidate, because we wanted to obtain as accurate a headline as possible, and it might be difficult for workers to select an appropriate absolute score. The workers were instructed as follows (English translation):

> Various candidate headlines are listed as options for a posted question in a Q&A service. After reading the question, please select the best option from the list so that users can guess the content of the question and distinguish it from other ones. Please remove uninformative ones such as greetings, self-introductions, and unspecific expressions.

We explain how to judge the appropriateness of each candidate by means of the example in Figure 2. After examining the posted question, we can assume that the most important content is "he is annoyed by the barking of a dog kept in the next house". On the basis of this assumption, option 6 is the best one, since the headline "A dog kept in the next house barks from morning" is enough to help answerers deduce that "the questioner is annoyed by the sound". Option 1 is inappropriate because although the answerers might be able to guess that "the questioner cannot move out", this matter is not the central one. Option 2 is uninformative because it consists merely of greetings and self-introduction, and option 3, while a question sentence, is unspecific. Option 4 enables answerers to guess that this question is related to an issue involving pets, but they cannot grasp the specific content. Option 5 specifies a likely damage due to the trouble, but the reason (trouble) is more important for the answering than the result (damage). Option 7 directly shows that "the questioner is annoyed and wants some advice", but answerers cannot understand why he/she is annoyed.

The detailed implementation of our task is as follows. First we randomly sorted the candidates of each question (shown in Figure 2(a)) to avoid position bias by the workers. We included ten actual questions and a dummy question so that workers would always have to answer one dummy question per every ten actual questions. A dummy question is a question with a clear answer inserted to eliminate fraud



(a) Ratio of true best candidates.

(b) Average votes for position.

(c) Average rank for position

Figure 3: Ratio of true best candidates and average votes/rank (for position) versus sentence position.

workers (i.e., workers who randomly select answers without actually reading them). Each question was answered by ten workers so each headline candidate had a vote score from 0 to 10 representing whether or not the candidate was appropriate for a headline. This task took nine days and was answered by 1,558 workers. As a result, our dataset consists of 10,000 questions, each of which has more than five headline candidates with accompanying "headline-ness" scores.

3.3 Analysis of Crowdsourcing Results

We analyzed our dataset to determine how much room for improvement our task has compared to the prefix headline. It is well known that the first sentence can be a strong baseline in many summarization tasks, so the prefix headline is also expected to perform well. Figure 3 shows the statistical information based on sentence position that includes the (a) ratio of the true best (most voted) candidates, (b) average votes, and (c) average rank (in order of votes) over the candidates at each sentence position. Looking at the true best candidates (a), the ratio 61.8% for the 1st sentence clarifies the effectiveness of the prefix headline, as expected. Conversely, we still have room for improvement for the prefix headline up to 38.2%. Our goal is to improve the uninformative headlines of 38.2% while keeping the remaining 61.8% unchanged. The other two figures (b) and (c) also support the above discussion.

Furthermore, we qualitatively checked the crowdsourcing quality. Workers successfully eliminated uninformative candidates including greetings and self-introductions, while one or two workers sometimes chose ones that included a fixed phrase such as "Please tell me ...". This is probably because workers had different criteria regarding the "unspecific expressions" described in the instructions. Since we cannot enumerate all of the concrete bad examples, we ignore this phenomenon with the expectation that a learning algorithm will reduce its negative effect.

4 Proposed Method

In this section, we explain how to construct a headline generation model from the dataset presented in Section 3. We took a ranking approach, i.e., learning to rank, for our task, rather than a simple regression one, since estimating absolute scores is not required for our purpose. Even if two headline candidates (of different questions) have the same expression, their votes can be significantly different since the votes in our dataset are based on relative evaluation. For example, the best candidate for Figure 2 was No. 6, but it might not be selected in other questions such as "A dog kept in the next house barks from morning. Does anybody know why dogs generally want to bark?".

Learning to rank is an application of machine learning that is typically used for ranking models in information retrieval systems. The ranking models are basically learned from a supervised dataset consisting of triples (q, x, y), where q is a user's query, x is a document corresponding to q, and y is a relevance score of x with respect to q. In this work, we formalize our task by regarding q, x, and y, as a posted question, a headline candidate, and a voted score in our dataset, respectively.

We used a pairwise ranking method that is also implemented as an instance of the well-known SVM tools LIBLINEAR (Lee and Lin, 2014) and LIBSVM (Kuo et al., 2014). We used a linear model based on LIBLINEAR, an L2-regularized L2-loss linear rankSVM, for the experiments. Let D be a dataset that consists of triples including a posted question q, a headline candidate x, and a voted score y, i.e., $(q, x, y) \in D$. In the pairwise ranking method, we train a ranking model as a binary classifier that determines whether the condition $y_i > y_j$ is true or false for two candidates x_i and x_j in the same question ($q_i = q_j$). Specifically, we first define the index pairs of positive examples by $P = \{(i, j) \mid q_i = q_j, y_i > y_j, (x_i, y_i, q_i) \in D, (x_j, y_j, q_j) \in D\}$. Note that we do not need to consider the negative examples $N = \{(j, i) \mid (i, j) \in P\}$ since they yield the same formula as P in the optimization process. The training of the pairwise ranking method is achieved by solving the following optimization problem using the set P of the index pairs:

$$\min_w \ \tfrac{1}{2}w^\top w + C \textstyle\sum_{(i,j)\in P} \ell(w^\top \tilde{x}_i - w^\top \tilde{x}_j), \tag{3}$$

where w is a weight vector to be learned, $\tilde{x}_i$ is a feature vector extracted from a headline candidate x, and C is the regularization parameter. The function ℓ is a squared hinge loss, which is defined as $\ell(d) = \max(0, 1 - d)^2$. Finally, we define the score function in Eq. (1) as $f_q(x) = w^\top \tilde{x}$, where $\tilde{x}$ can be created by using q as well as x. This score means the relative "headline-ness" of x.

5 Experiments

5.1 Basic Settings

The basic settings of the experiments are as follows. We split our dataset into training and test sets consisting of 9,000 and 1,000 examples, respectively. We used an implementation[3] based on LIBLINEAR for training our ranking model, i.e., a linear L2-regularized L2-loss rankSVM model (Lee and Lin, 2014), as described in Section 4. The regularization parameter was optimized by cross validation and set as $C = 0.125$.

The feature vector for a headline candidate consists of three kinds of features: bag-of-words, embedding, and position information. The bag-of-words feature is a sparse vector of 30,820 dimensions based on the tf-idf scores of nouns, verbs, interjections, conjunctions, adverbs, and adjectives in a candidate, where we used a Japanese morphological analyzer, MeCab[4] (Kudo et al., 2004), with a neologism dictionary, NEologd[5] (Toshinori Sato and Okumura, 2017). The embedding feature is a dense vector of 100 dimensions based on a doc2vec model (Le and Mikolov, 2014) trained with all 3M sentences in the Chiebukuro dataset using the Gensim tool[6]. The position feature is a binary vector of ten dimensions, where each dimension represents the coarse position (or coverage) of a headline candidate for a question. Specifically, we equally split a question (character sequence) into ten parts and set one to each dimension if and only if the corresponding part overlaps a candidate. For example, candidate No. 2 in Figure 2 had a position feature $(1, 1, 0, \cdots, 0)$, since the candidate covers the first 2/10 of the whole question. Similarly, No. 6 and No. 3 had $(0, 0, 1, 1, 0, \cdots, 0)$ and $(0, \cdots, 0, 1)$, respectively. For constructing the feature vector of each headline candidate, we used the previous and next candidates in sentence order, in addition to the target candidate. This is based on the idea that near candidates might have useful information for the target candidate. Finally, we prepared each feature vector by concatenating nine feature vectors, i.e., the above three kinds of features for three candidates, and normalizing them.

5.2 Compared Methods

We compared our method, `MLRank`, with the baselines listed below. `Prefix`, `DictDel`, and `Random` are simple baselines, while `Prefix` and `DictDel` are practically strong. `ImpTfidf`, `SimTfidf`, `SimEmb`, and `LexRank` are unsupervised baselines, and `SVM` and `SVR` are supervised ones.

- `Prefix`: Selects the first candidate in sentence order.
- `DictDel`: Selects the first (informative) candidate that does not match in the dictionary of uninformative headlines (Section 2).
- `Random`: Randomly selects a candidate.
- `ImpTfidf`: Selects the most important candidate with the highest tf-idf value, where a tf-idf value is calculated by the sum of the elements in a bag-of-words feature (described in Section 5.1).

[3] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#large_scale_ranksvm`

[4] `https://taku910.github.io/mecab/`

[5] `https://github.com/neologd/mecab-ipadic-neologd`

[6] `https://radimrehurek.com/gensim/models/doc2vec.html`

- `SimTfidf`: Selects the most similar candidate to the original question, which is calculated by the cosine similarity between the bag-of-words features (in Section 5.1) of each candidate and the question.
- `SimEmb`: An embedding-based variation of `SimTfidf` with embedding features (in Section 5.1).
- `LexRank`: Selects the candidate with the highest score based on LexRank[7] (Erkan and Radev, 2004), which is a widely used unsupervised extractive summarization method based on the PageRank algorithm. The graph expression of each question was constructed on the basis of cosine similarity of the tf-idf vectors corresponding to candidates.
- `SVM`: Selects the candidate with the highest confidence based on a model learned as a classification task, where candidates with nonzero votes were labeled as positive. This setting was the best in our preliminary experiments. We used the L2-regularized L2-loss support vector classification model ($C = 0.0156$) in LIBLINEAR. The other settings were the same as those described in Section 5.1.
- `SVR`: Selects the candidate with the highest predicted votes based on a model learned as a regression task, where the target variable is the number of votes. We used the L2-regularized L2-loss support vector regression model ($C = 0.0625$) in LIBLINEAR. The other settings were the same as above.
- `MLRank`: Proposed method described in Section 4.

5.3 Evaluation Measures

We defined three evaluation measures for evaluating each method on our headline generation task.

Change Rate from Prefix Headline

We measured how much each method changed the default prefix headline in order to determine the effect of application to an actual CQA service. We defined this measure as *change rate from the prefix headline*, as follows:

$$\text{Change rate} = \frac{\text{No. of questions where the best candidate is not the prefix headline}}{\text{No. of all questions}}. \tag{4}$$

Clearly, the change rate of the default method that selects the prefix headline is 0%. If the value is small, the effect on the service will be small, but if the value is higher than the ideal change rate of 38.2% (Section 3), there can be side effects even if the average result is good. A higher change rate up to the ideal rate is desirable from a practical standpoint.

Winning Rate against Prefix Headline

We measured how much each method won against the prefix headline to directly assess the quality of changed headlines. We defined this measure as *winning rate against the prefix headline*, as follows:

$$\text{Winning rate} = \frac{\text{No. of questions where the best candidate got more votes than the prefix headline}}{\text{No. of questions where the best candidate is not the prefix headline}}. \tag{5}$$

We did not consider the first candidate (in sentence order) selected by each method, which is the same as the prefix headline, since they obviously have the same number of votes.

Average Votes

We measured how appropriate the candidates selected by each method are in order to determine the overall performance. We defined this measure as *average votes*, as follows:

$$\text{Average votes} = \frac{\text{Sum of votes for the best candidates for all questions}}{\text{No. of questions}}. \tag{6}$$

Note that the average votes score is different from the average votes score for position (Figure 3(b)) in that the former is the average over the selected candidates while the latter is the average over the candidates at a sentence position.

[7] `https://pypi.org/project/lexrank/0.0.1b0`

E.g.	Prefix method (`Prefix`)	Proposed method (`MLRank`)
1	27 歳女です。環境的になかなか新しい出… I am a 27-year-old woman. Owing to my environment, there is little chance of new ...	…環境的になかなか新しい出会いがなくて… ... Owing to my environment, there is little chance of new encounters with men ...
2	情緒不安定なものです。回答して下さった… I am emotionally unstable. Those who answered ...	…新婚旅行は、行きは別々で行き、現地で… ... For the honeymoon, we went separately and in the field ...
3	カテ違いならゴメンナサイ。今、財布が破… I am sorry if the category is wrong. Now, my wallet is torn	…今、財布が破れてツライ状況です。新し… ... Now, my wallet is torn, and I'm having a hard time. A new one ...
4	前にとても大変で嫌な思いをしたので携帯… Because things felt very hard and painful before, my mobile ...	…携帯メールお断りというのはこの落札者… ... Rejecting e-mails from a mobile phone means this winning bidder ...
5	現在、私の父は 60 歳で定年退職を迎えま… Currently my father is 60 years old and will retire from employment ...	…厚生年金の支払いは何歳までなんですか… ... Until what age should he pay the welfare pension ...

Table 2: Examples of prefix method `Prefix` and proposed method `MLRank`.

This measure is related to (normalized) discounted cumulative gain (DCG), which is widely used as an evaluation measure of ranking models. We often use DCG@k for evaluating top-k rankings, and the above definition acutally corresponds to DCG@1. According to a well-known paper (Järvelin and Kekäläinen, 2002) in the information retrieval field, DCG is appropriate for graded-relevance judgments like our task, while precision (described below) is appropriate for binary-relevance ones. Average votes is expected to be more appropriate than precision for our task because we want "averagely better headlines than default ones" rather than "best ones" from a practical standpoint.

Precision

Precision is a widely used evaluation measure for classification tasks, and we added it to support an evaluation based on average votes. We defined it with respect to the best candidate, i.e., precision@1, as follows:

$$\text{Precision} = \frac{\text{No. of questions where the best candidate had the maximum votes}}{\text{No. of questions}}. \tag{7}$$

5.4 Results

Qualitative Analysis

Table 2 shows examples of headlines generated by the prefix method, `Prefix`, and the proposed method, `MLRank`. Looking at the first example, we can see that our method successfully eliminated the self-introduction phrase ("I am a 27-year-old woman"). The headline (right) of our method allows answerers to know that the questioner is discouraged about how to encounter men from the phrase "little chance of new encounters with men", while the headline (left) of the prefix method lacks this important clue. The second and third examples show similar effects to the first example. In the second one, although there are few clues about what the question is with the prefix method, our method correctly included the important clue ("honeymoon"). In the third one, our method appropriately eliminated the uninformative long phrase ("I am sorry if the category is wrong"), which is not a frequent fixed phrase. The fourth example shows a slightly challenging case, where both headlines make it difficult to understand the question. However, the headline of our method included the term "winning bidder", so at least the answerer can assume that the question is about some sort of auction trouble. The fifth example is a clearly successful result, where our method extracted the main question point about "welfare pension" as a headline. These results qualitatively demonstrate the effectiveness of our method.

Quantitative Analysis

We compared our method `MLRank` with the baselines in Section 5.2 on the headline generation task for our dataset in Section 3. Table 3 shows the evaluation results based on the change rates, winning rates, average votes, and precision. Looking at the average votes and precision, which represent the overall performances, our method `MLRank` clearly performed the best among all methods. We confirmed that

	Change rate (%)	Winning rate (%)	Average votes	Precision (%)
Ref (reference)	38.2	100.0	5.56	100.0
Prefix (default)	0	–	4.19	61.8
DictDel	2.2	72.0	4.23	61.3
Random	85.9	11.7	1.39	16.1
ImpTfidf	81.1	12.7	1.68	20.0
SimTfidf	79.3	18.7	2.27	20.0
SimEmb	**88.2**	13.0	1.40	15.4
LexRank	55.7	19.3	2.27	29.9
SVM	16.7	50.1	4.09	60.3
SVR	52.5	25.7	3.00	42.1
MLRank (proposed)	9.9	**94.9**	**4.28**	**62.6**

Table 3: Evaluation results of our headline generation task for proposed method MLRank and baselines.

the relative improvement of the average votes of our method MLRank against every baseline including the prefix method Prefix is statistically significant on the basis of a one-tailed Wilcoxon signed-rank test ($p < 0.01$). The change and winning rates of our method are 9.9% and 94.9%, respectively. This means that our method detected 9.9% of the uninformative headlines and improved them with the high accuracy of 94.9%. In other words, our method could successfully improve the overall performance while simultaneously avoiding any negative side effects. The ideal results (Ref) based on correct labels suggest that our method still has room for improvement, especially for the change rate.

The results of the other baselines are as follows. Not surprisingly, the prefix method Prefix performed well. This is consistent with the fact that the first sentence can be a good summary in many summarization tasks. The random method Random performed the worst, also as expected. The dictionary-based deletion method DictDel was relatively useful, although the change rate was small. The reason the winning rate of DictDel is relatively low compared with MLRank is that there are some cases where a combination of uninformative expressions can yield likely clues. For example, the self-introduction "I am a newbie of this forum" itself is basically uninformative for a question, but a combination with additional information such as "I am a newbie of this forum. Where can I change the password ..." can be more informative than only the additional information "... Where can I change the password since I forgot it after ..." because the combination specifies the target site by the expression "this forum".

The unsupervised methods, SimTfidf, SimTfidf, SimEmb, and LexRank, which are widely used for summarization tasks, performed worse than the prefix method Prefix. Although the change rates are higher than our method, the winning rates are lower. In other words, they yielded many bad headlines. These results suggest that supervised learning specialized to the target task would be required. Comparing the important sentence extraction method ImpTfidf and the similarity-based summarization method SimTfidf, we found that SimTfidf performed better. This implies that the content information of each question is useful for our headline generation task, as is the case with other summarization tasks. The similarity-based method SimEmb with embeddings performed worse than our expectation. The reason seems to be that it was difficult to obtain meaningful document embeddings from long questions. The graph-based method LexRank had a similar performance to SimTfidf, because LexRank tends to select a candidate similar to the question when only one candidate was selected. The supervised methods, SVM and SVR, performed relatively well compared to the unsupervised methods, but they did not outperform the strong simple baselines, Prefix and DictDel, nor our method MLRank. These results support the appropriateness of our approach.

6 Related Work

In this section, we briefly explain several related studies from two aspects: headline generation task and CQA data. As discussed below, our work is the first attempt to address an extractive headline generation task for a CQA service based on learning to rank the substrings of a question.

After Rush et al. (2015) proposed a neural headline generation model, there have been many studies on the same headline generation task (Takase et al., 2016; Chopra et al., 2016; Kiyono et al., 2017; Zhou et al., 2017; Suzuki and Nagata, 2017; Ayana et al., 2017; Raffel et al., 2017). However, all of them are abstractive methods that can yield erroneous output, and the training for them requires a lot of paired data, i.e., news articles and headlines. There have also been several classical studies based on non-neural approaches to headline generation (Woodsend et al., 2010; Alfonseca et al., 2013; Colmenares et al., 2015), but they basically addressed sentence compression after extracting important linguistic units such as phrases. In other words, their methods can still yield erroneous output, although they would be more controllable than neural models. One exception is the work of Alotaiby (2011), where fixed-sized substrings were considered for headline generation. Although that approach is similar to ours, Alotaiby only considered an unsupervised method based on similarity to the original text (almost the same as `SimTfidf` in Section 5.2), in contrast to our proposal based on learning to rank. This implies that Alotaiby's method will also not perform well for our task, as shown in Section 5.4. There have been several studies on extractive summarization (Kobayashi et al., 2015; Yogatama et al., 2015) based on sentence embeddings, but they were basically developed for extracting multiple sentences, which means that these methods are almost the same as `SimEmb` in Section 5.2 for our purpose, i.e., extraction of the best candidate. This also implies that they will not be suitable for our task. Furthermore, recent sophisticated neural models for extractive summarization (Cheng and Lapata, 2016; Nallapati et al., 2017) basically require large-scale paired data (e.g., article-headline) to automatically label candidates, as manual annotation is very costly. However, such paired data do not always exist for real applications, as in our task described in Section 1.

There have been many studies using CQA data, but most of them are different from our task, i.e., dealing with answering questions (Surdeanu et al., 2008; Celikyilmaz et al., 2009; Bhaskar, 2013; Nakov et al., 2017), retrieving similar questions (Lei et al., 2016; Romeo et al., 2016; Nakov et al., 2017), and generating questions (Heilman and Smith, 2010). Tamura et al. (2005) focused on extracting a core sentence and identifying the question type as classification tasks for answering multiple-sentence questions. Although their method is useful to retrieve important information, we cannot directly use it since our task requires shorter expressions for headlines than sentences. In addition, they used a support vector machine as a classifier, which is almost the same as `SVM` in Section 5.2, and it is not expected to be suitable for our task, as shown in Section 5.4. The work of Ishigaki et al. (2017) is the most related one in that they summarized lengthy questions by using both abstractive and extractive approaches. Their work is promising because our task is regarded as the construction of short summaries, but the training of their models requires a lot of paired data consisting of questions and their headlines, which means that their method cannot be used to our task.

7 Conclusion

We proposed an extractive headline generation method based on learning to rank for CQA that extracts the most informative substring in each question as its headline. We created a dataset for our task, where headline candidates in each question are ranked using crowdsourcing. Our method outperformed several baselines, including a prefix-based method, which is widely used for cases where the display area is limited, such as the push notifications on smartphones. The dataset created for our headline generation task will be made publicly available[8]. Although our task is basically designed for extractive summarization, this dataset can also be used for abstractive summarization as a side information for training abstractive models.

In future work, we will investigate how effectively our method can perform in practical situations, e.g., push notifications. In addition, we will consider how to improve the change rate of our method while keeping its winning rate and how to create a useful dataset even if removing the length limitation.

Acknowledgements

We would like to thank the engineers at Yahoo! Chiebukuro for supporting the A/B testing in Section 2 and the anonymous reviewers for giving us many useful comments.

[8] `https://research-lab.yahoo.co.jp/en/software/`

References

Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. HEADY: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1243–1253. Association for Computational Linguistics.

Fahad Alotaiby. 2011. Automatic Headline Generation using Character Cross-Correlation. In *Proceedings of the ACL 2011 Student Session*, pages 117–121. Association for Computational Linguistics.

Ayana, Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, and Mao-Song Sun. 2017. Recent Advances on Neural Headline Generation. *Journal of Computer Science and Technology*, 32(4):768–784.

Pinaki Bhaskar. 2013. Answering Questions from Multiple Documents – the Role of Multi-Document Summarization. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 14–21. INCOMA Ltd. Shoumen, BULGARIA.

Asli Celikyilmaz, Marcus Thint, and Zhiheng Huang. 2009. A graph-based semi-supervised learning for question-answering. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 719–727, Suntec, Singapore, August. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 484–494. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 93–98. Association for Computational Linguistics.

Carlos A. Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, pages 133–142. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22(1):457–479.

Michael Heilman and Noah A. Smith. 2010. Good Question! Statistical Ranking for Question Generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 609–617. Association for Computational Linguistics.

Tatsuya Ishigaki, Hiroya Takamura, and Manabu Okumura. 2017. Summarizing Lengthy Questions. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017)*, pages 792–800. Asian Federation of Natural Language Processing.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Shun Kiyono, Sho Takase, Jun Suzuki, Naoaki Okazaki, Kentaro Inui, and Masaaki Nagata. 2017. Source-side Prediction for Neural Headline Generation . *CoRR*, abs/1712.08302.

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization based on embedding distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal, September. Association for Computational Linguistics.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain, July. Association for Computational Linguistics.

Tzu-Ming Kuo, Ching-Pei Lee, and Chih-Jen Lin. 2014. Large-scale Kernel RankSVM. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM 2014)*, pages 812–820.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML 2014)*, pages 1188–1196. JMLR.org.

Ching-Pei Lee and Chih-Jen Lin. 2014. Large-scale Linear Ranksvm. *Neural Computation*, 26(4):781–817.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised Question Retrieval with Gated Convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 1279–1289. Association for Computational Linguistics.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 3075–3081. AAAI Press.

Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and Linear-Time Attention by Enforcing Monotonic Alignments. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pages 2837–2846.

Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James Glass. 2016. Neural Attention for Learning to Rank Questions in Community Question Answering. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 1734–1745. The COLING 2016 Organizing Committee.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389. Association for Computational Linguistics.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to Rank Answers on Large Online QA Collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2008)*, pages 719–727. Association for Computational Linguistics.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 291–297. Association for Computational Linguistics.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural Headline Generation on Abstract Meaning Representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1054–1059. Association for Computational Linguistics.

Akihiro Tamura, Hiroya Takamura, and Manabu Okumura. 2005. Classification of Multiple-Sentence Questions. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, pages 426–437. Springer-Verlag Berlin Heidelberg.

Taiichi Hashimoto Toshinori Sato and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing*, pages NLP2017–B6–1. The Association for Natural Language Processing.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title Generation with Quasi-Synchronous Grammar. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 513–523. Association for Computational Linguistics.

Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive Summarization by Maximizing Semantic Volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1095–1104. Association for Computational Linguistics.