Semi-supervised Extractive Question Summarization Using Question-Answer Pairs

Kazuya Machida[1], Tatsuya Ishigaki[1(✉)], Hayato Kobayashi[2], Hiroya Takamura[1,3], and Manabu Okumura[1]

[1] Tokyo Institute of Technology, Yokohama, Japan
{machida,ishigaki}@lr.pi.titech.ac.jp, oku@pi.titech.ac.jp
[2] Yahoo Japan Corporation/RIKEN AIP, Tokyo, Japan
hakobaya@yahoo-corp.jp
[3] AIST, Tokyo, Japan
takamura.hiroya@aist.go.jp

Abstract. Neural extractive summarization methods often require much labeled training data, for which headlines or lead summaries of news articles can sometimes be used. Such directly useful summaries are not always available, however, especially for user-generated content, such as questions posted on community question answering services. In this paper, we address an extractive summarization (i.e., headline extraction) task for such questions as a case study and consider how to alleviate the problem by using question-answer pairs, instead of missing-headline pairs. To this end, we propose a framework to examine how to use such unlabeled paired data from the viewpoint of training methods. Experimental results show that multi-task training performs well with undersampling and distant supervision.

Keywords: Question summarization · Headline extraction

1 Introduction

Questions are a means of acquiring knowledge, and since the advent of the Internet, many questions have been posted on community question-answering (CQA) sites. Therefore, to find questions efficiently, we need a system by which the important parts of questions can be displayed in search results. On a CQA site, as

Question: Hello, I have an AU's iPhone 5S ...	**Answer:** The iPhone's initial setup requires a SIM card and a PC that can use the Internet. If you don't have a PC, try connecting to Wi-Fi at a convenience store or other location. If you don't have a SIM card, borrow someone else's.
Hello, I have an AU iPhone 5S, but it still has the default settings **Default Headline Sent.**	
I have no Wi-Fi at home, so I cannot set it up	
Is there any way to do the iPhone's initial setup without Wi-Fi? **Actual Important Sent.**	
If there is, please tell me:)	

Fig. 1. Example of a posted question and its answer.

K. Machida and T. Ishigaki—Both equally contributed to this work.

© Springer Nature Switzerland AG 2020
J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 255–264, 2020.
https://doi.org/10.1007/978-3-030-45442-5_32

represented by Yahoo! Chiebukuro [31], the first sentence of a question tends to be displayed as a headline (or list item) because of a restriction on the display area. Note that, to reduce the burden on users who post questions, many CQA sites do not provide an input field for headlines in the submission form. The most important sentence in a question, however, should be displayed instead of the first sentence, because sometimes the first sentence does not provide enough information, as shown in Fig. 1 (translated to English).

This task can be formalized as extractive summarization, which has long been addressed, e.g., by using a graph-based method [21], a topic-based method [10], or a features-based method [22]. The development of neural networks has led some studies [7,24] to report high-performance models that use large amounts of training data. Such large amounts of training data, however, incur a high cost to create and cannot always be prepared for practical use.

In this paper, we harness question-answer (QA) pairs to alleviate this problem. Many QA pairs on CQA sites can be easily obtained without annotation costs and are expected to be useful because, in general, each answer should be closely related to the most important sentence in the question. In fact, the answer in Fig. 1 includes keywords such as "initial setup" and "Wi-Fi" in the main question sentence. Our framework can be regarded as a semi-supervised approach with a small amount of labeled data and a large amount of unlabeled (paired) data. The main difference from classical semi-supervised settings is that unlabeled data has a paired structure. This allows us to formulate our problem as a multi-task problem of sentence extraction and answer generation. One of the difficulties of this formulation is "data imbalance", meaning that there is a small amount of data for sentence extraction and a large amount for answer generation. Therefore, we focus on this data imbalance problem and investigate how to use the unlabeled paired data from the viewpoint of training methods.

The contributions of our study are as follows.

- We address extractive question summarization with QA pairs as a case study of a semi-supervised setting with unlabeled paired data and we propose a simple framework to systematically examine different ways to use these pairs.
- We compare different training methods, namely, pretraining, separate training, and multi-task training, as well as normal training. Our experimental results show that (a) multi-task training performs the best but does not work well without an appropriate sampling method to reduce the data imbalance, and that (b) the multi-task training method is further enhanced with data augmentation based on distant supervision, which can simply solve the data imbalance problem. Our data and code will be publicly available [14].

2 Framework

Our framework consists of two models (Fig. 2); the **sentence extraction model (SEM)** based on a sequence labeling structure, and the **answer generation model (AGM)** based on a sequence-to-sequence structure. SEM directly solves our task, whereas AGM provides auxiliary information via attention weights.

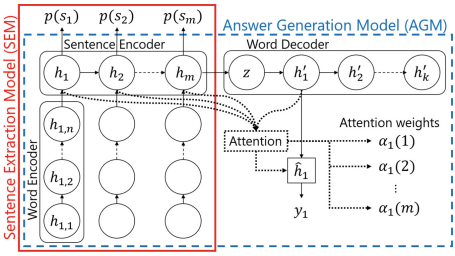

Fig. 2. Overview of our framework.

SEM first encodes a question with sentences $(s_1, \ldots, s_m)$ into sentence vectors $(h_1, \ldots, h_m)$ via a hierarchical encoder based on two LSTM units for words and sentences. Then, for each sentence s_i, the model calculates the extraction probability $p(s_i)$, which represents the importance score of s_i, by applying a binary softmax function with a linear transformation to h_i. In the training phase, we use the cross entropy loss L_{ext} based on $p(s_i)$ and the true label, similarly to classification tasks. We use SEM to define the importance score of s_i as $f_{\text{ext}}(s_i) = p(s_i)$, which is used for the evaluation phase, together with a score obtained by AGM as described below.

AGM encodes a question into sentence vectors in the same way as in SEM. The model uses these vectors to generate an answer (word sequence) by using an ordinary sequence-to-sequence model with an attention mechanism. We do not use a hierarchical decoder, because the main purpose of this study is not to improve the performance of answer generation. In the training phase, we use the negative log likelihood loss L_{gen} based on a predicted sequence and the correct sequence. In the evaluation phase, we calculate importance scores by using attention weights $\alpha_j(i)$, each of which represents the alignment level with respect to s_i at the j-th step in generation. Specifically, we define the importance score of s_i obtained by AGM as the average of the attention weights for s_i, i.e., $f_{\text{gen}}(s_i) = \frac{1}{k}\sum_{j=1}^{k} \alpha_j(i)$.

In our framework, we can thus simultaneously train two models in a multi-task setting (SEM and AGM are the respective main and auxiliary models) and combine their importance scores to estimate the most important sentence. We introduce two tuning parameters λ and κ for training and evaluation phases, respectively. The final loss function for the training phase is $\lambda L_{\text{ext}} + (1-\lambda)L_{\text{gen}}$, and the score function for the evaluation phase is $\kappa f_{\text{ext}}(s_i) + (1-\kappa) f_{\text{gen}}(s_i)$.

3 Experiment

Datasets: We prepared two datasets, `Pair` and `Label`, which were based on a publicly available CQA dataset [25] provided by Yahoo! Chiebukuro. These two datasets formed a semi-supervised setting with unlabeled paired data, in which `Pair` included many unlabeled QA pairs for training AGM, while `Label` included a few labeled questions for SEM.

`Pair` consisted of 100K QA pairs, each of which included a randomly sampled question and its best answer annotated in the CQA dataset. In the sampling procedure, we removed pairs including more than 10 sentences to reduce the computational cost, as these were less than 5% of the total. For the same reason, we removed pairs including sentences consisting of more than 50 words.

`Label` consisted of 775 questions sampled separately but in a similar way to `Pair`. Every sentence in each question had a binary label representing whether the sentence was the most important, meaning that only the best sentence had a label of 1, while the others had a label of 0. We used crowdsourcing to annotate `Label`. In the crowdsourcing, five workers were given a question and asked to select the best sentence representing the main focus of the question. We included only questions for which at least four workers selected the same sentence.

Unsupervised Baselines: We prepared the following unsupervised methods as simple baselines.

- `Lead`: Selects the initial sentence.
- `TfIdf`: Selects the sentence with the highest average tf-idf on the basis of the CQA dataset.
- `SimEmb`: Selects the sentence with the highest similarity on the basis of the word mover's distance [18] to the input question.
- `LexRank`: Uses a graph-based, unsupervised, extractive summarization model [8], which was trained with all the questions.

Compared Methods: We systematically compared the following methods to study how to effectively use `Pair` by changing the parameter settings of λ and κ in our framework.

- `Ext`: Trains and uses SEM only ($\lambda = 1, \kappa = 1$).
- `Gen`: Trains and uses AGM only ($\lambda = 0, \kappa = 0$).
- `Sep`: Trains SEM ($\lambda = 1$) and AGM ($\lambda = 0$) separately and combines them in the evaluation phase. Then, κ is tuned with the development set.
- `Pre`: Trains SEM ($\lambda = 1$) after initializing the encoder's parameters by using AGM ($\lambda = 0$). Prediction is done with SEM ($\kappa = 1$).
- `Multi`: Trains SEM and AGM simultaneously. Mini-batches are created for each dataset and shuffled, with the loss calculated per mini-batch. Then, λ and κ are tuned with the development set.

Oversampling/Undersampling: We additionally prepared two variants of `Multi` to reduce the data imbalance problem of `Label` and `Pair`, because the data size of the subtask is much larger than that of the main task. Specifically, we used oversampling and undersampling to reduce the imbalance as follows.

- `MultiOver`: Oversamples `Label` multiple times to be the same size as `Pair`.
- `MultiUnder`: Undersamples `Pair` to be the same size as `Label` in every epoch.

Distant Supervision: Furthermore, we prepared a pseudo labeled dataset `Pseudo`, which included pseudo (noisy) labels for all the questions in `Pair`. This pseudo labeling approach is often called distant supervision, in where unlabeled data is automatically annotated with some heuristic rules. Following Ishigaki et al. [15], we adopted their heuristic rule that single-sentence questions are basically self-contained and have summary-like characteristics. Because their labels for single-sentence questions could not be directly used for our questions with multiple sentences, we first trained a classifier with their labels and used it to make `Pseudo`. Thus, using `Pseudo`, we prepared the following variants of `Multi`, `Ext`, `Sep`, and `Pre` for comparison.

- `MultiDist`: `Multi` trained with `Label`, `Pair`, and `Pseudo`.
- `ExtDist`/`SepDist`/`PreDist`: Variants of `Ext`/`Sep`/`Pre`, similar to `MultiDist`.

Evaluation: For evaluating the performance, we used an accuracy measure calculated by dividing the number of questions for which the target method correctly selected the most important sentence by the number of questions used. Note that well-known metrics such as ROUGE and precision/recall were not appropriate, because our task was to find only one sentence as a (snippet) headline. We divided the labeled data `Label` into five sets (train:develop:test = 3:1:1) and performed five-fold cross-validation to evaluate the methods.

Results: Table 1 lists the results. The three row groups from top to bottom correspond to unsupervised, semi-supervised, and distantly supervised settings. In the first group, `Lead` performed the best, whereas the other methods (`TfIdf`, `SimEmb`, and `LexRank`) did not work well. This indicates the difficulty of our task and confirms that we need supervision to develop practical models.

Table 1. Accuracy on the question summarization task. Each "✓" indicates that the corresponding dataset was used.

	Label	Pair	Pseudo	Acc
Lead	–	–	–	.690
TfIdf	–	–	–	.237
SimEmb	–	–	–	.472
LexRank	–	–	–	.587
Ext	✓	–	–	.813
Gen	–	✓	–	.649
Sep	✓	✓	–	.828
Pre	✓	✓	–	.788
Multi	✓	✓	–	.770
MultiOver	✓	✓	–	.833
MultiUnder	✓	✓	–	.857
ExtDist	✓	–	✓	.838
SepDist	✓	✓	✓	.855
PreDist	✓	✓	✓	.834
MultiDist	✓	✓	✓	**.875**

In the second group, `MultiUnder` performed the best, although `Multi` (without sampling) performed worse than `Ext` did. This suggests that reducing the data imbalance is a key factor for our setting. `MultiOver` also worked well but did not reach the performance of `MultiUnder`. The reason seems to be that sampling the same data many times yields overfitting. Among other methods, `Sep` performed well because of an ensemble effect of `Ext` and `Gen`, whereas `Gen` by itself performed the worst because it did not use any labels. `Pre` unexpectedly performed worse than `Ext` did, although Shimizu

et al. [27] reported that sentiment classifiers were more enhanced by pretraining with tweet-reply pairs than by language model pretraining. This implies that the performance depends on the task settings, so our framework can be useful for other tasks.

In the third group, `MultiDist` (without sampling) performed the best. The differences from the other methods in this group were statistically significant according to the sign test ($p < 0.05$). Although distant supervision itself has positive effects as shown by the improvement for `ExtDist`, it has an extra bonus in that pseudo labels can simply solve the data imbalance. These results suggest that we have room to study the combinations of multi-task training and distant supervision for other NLP tasks. We also prepared a larger labeled dataset than `Label`. The experiments on this dataset showed similar tendencies. We will study how the data size of labeled data affects the performances in future work.

4 Related Work

Several studies have considered semi-supervised settings for summarization tasks [1,20,30], but in contrast to our main focus, none of them considered multi-task settings, especially using paired data. In the multi-task field, there have been several studies on summarization tasks. Guo et al. [11] improved an abstractive summarization model by using multi-task training with entailment and question generation tasks. Their work used human-annotated data from SQuAD dataset for these auxiliary tasks, whose sizes were much smaller than that of the main task, so their setting was completely different from ours. Angelidis et al. [2] addressed summarization of opinions from Amazon reviews by using multi-task training with aspect extraction and sentiment prediction tasks. Their work is related to ours in that they targeted user-generated content, but their auxiliary tasks were basic subtasks of opinion summarization with explicit aspect or sentiment labels. This implies that their task's usefulness was clearer than that of our task, in which we only assume a paired structure without any explicit labels. The study most related to ours is the work by Isonuma et al. [17], who proposed an extractive summarization method for news articles through multi-task training with a document classification task. Their strategy was similar to ours in that they used categories originally attached to news articles without costly annotation, but in many cases, we cannot access such categories or useful meta-information for documents, like CQA sites.

Several studies have used QA or similar structures for summarization tasks. Chen et al. [6] used a QA system to predict summarization quality in the evaluation phase, in contrast to our study, which uses QA paired data in the training phase. Arumae and Liu [3] used QA data to calculate a reward function for reinforcement learning in the training phase. They used Cloze-style (fill in the blank) questions, however, and we cannot directly apply their method to our task. Gao et al. [9] used an article-comments structure to personalize summaries in a multi-modal setting with multiple inputs, i.e., article and comments, rather

than multi-task settings with multiple outputs, as in our study. Note that we did not consider such a multi-modal setting, as we assumed that answers would not always available for posted questions.

Many studies have used CQA data, but most have addressed different tasks, i.e., dealing with answering questions [4,5,23,28], retrieving similar questions [19,23,26], and generating questions [12]. Tamura et al. [29] focused on extracting a core sentence and identifying the question type as a classification task for answering multiple-sentence questions. Higurashi et al. [13] proposed a learning-to-rank approach for extracting an important substring from a question. Although their models are useful for retrieving important information, they considered methods that are trained with only labeled data. Finally, Ishigaki et al. [16] addressed neural abstractive and extractive approaches to summarize lengthy questions by using much paired data consisting of questions and headlines. Therefore, their method is not applicable to our task, in which we assume questions without headlines.

5 Conclusion

We have addressed an extractive question summarization task with QA pairs as a case study of a semi-supervised setting with unlabeled paired data. Our results suggest that multi-task training is effective especially with undersampling and distant supervision. For future work, we will apply our framework to other tasks with similar structures, such as news articles with comments.

References

1. Amini, M.R., Gallinari, P.: The use of unlabeled data to improve supervised learning for text summarization. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), pp. 105–112. ACM (2002). https://doi.org/10.1145/564376.564397
2. Angelidis, S., Lapata, M.: Summarizing opinions: aspect extraction meets sentiment prediction and they are both weakly supervised. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), pp. 3675–3686. Association for Computational Linguistics (2018). http://www.aclweb.org/anthology/D18-1403
3. Arumae, K., Liu, F.: Reinforced extractive summarization with question-focused rewards. In: Proceedings of ACL 2018, Student Research Workshop, pp. 105–111. Association for Computational Linguistics (2018). http://www.aclweb.org/anthology/P18-3015
4. Bhaskar, P.: Answering questions from multiple documents - the role of multi-document summarization. In: Proceedings of the Student Research Workshop Associated with RANLP 2013, pp. 14–21. INCOMA Ltd., Shoumen (2013). http://www.aclweb.org/anthology/R13-2003

5. Celikyilmaz, A., Thint, M., Huang, Z.: 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, pp. 719–727. Association for Computational Linguistics (2009). http://www.aclweb.org/anthology/P/P09/P09-1081
6. Chen, P., Wu, F., Wang, T., Ding, W.: A semantic QA-based approach for text summarization evaluation. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 4800–4807. AAAI Press (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16115
7. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pp. 484–494. Association for Computational Linguistics (2016). http://www.aclweb.org/anthology/P16-1046
8. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. (JAIR) **22**(1), 457–479 (2004). https://doi.org/10.1613/jair.1523
9. Gao, S., et al.: Abstractive text summarization by incorporating reader comments. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI-2019. AAAI Press (2019). https://www.aaai.org/ojs/index.php/AAAI/article/view/4603/4481
10. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2001), pp. 19–25 (2001). https://doi.org/10.1145/383952.383955
11. Guo, H., Pasunuru, R., Bansal, M.: Soft layer-specific multi-task summarization with entailment and question generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), pp. 687–697. Association for Computational Linguistics (2018). http://www.aclweb.org/anthology/P18-1064
12. Heilman, M., Smith, N.A.: 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), pp. 609–617. Association for Computational Linguistics (2010). http://www.aclweb.org/anthology/N10-1086
13. Higurashi, T., Kobayashi, H., Masuyama, T., Murao, K.: Extractive headline generation based on learning to rank for community question answering. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), pp. 1742–1753. Association for Computational Linguistics (2018). http://www.aclweb.org/anthology/C18-1148
14. Ishigaki, T.: Scripts for preprocessing Yahoo Chiebukuro dataset (2020). http://lr-www.pi.titech.ac.jp/~ishigaki/chiebukuro/
15. Ishigaki, T., Machida, K., Kobayashi, H., Takamura, H., Okumura, M.: Distant supervision for question summarization. In: Proceedings of the 42nd Annual European Conference on Information Retrieval (ECIR 2020) (2020)
16. Ishigaki, T., Takamura, H., Okumura, M.: Summarizing lengthy questions. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017), pp. 792–800. Asian Federation of Natural Language Processing (2017). http://www.aclweb.org/anthology/I17-1080

17. Isonuma, M., Fujino, T., Mori, J., Matsuo, Y., Sakata, I.: Extractive summarization using multi-task learning with document classification. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pp. 2101–2110. Association for Computational Linguistics (2017). https://www.aclweb.org/anthology/D17-1223
18. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), vol. 37, pp. 957–966. PMLR (2015). http://proceedings.mlr.press/v37/kusnerb15.html
19. Lei, T., et al.: 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), pp. 1279–1289. Association for Computational Linguistics (2016). http://www.aclweb.org/anthology/N16-1153
20. Li, Y., Li, S.: Query-focused multi-document summarization: combining a topic model with graph-based semi-supervised learning. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014), pp. 1197–1207. Dublin City University and Association for Computational Linguistics (2014). http://www.aclweb.org/anthology/C14-1113
21. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004). http://aclweb.org/anthology/W04-3252
22. Naik, S.S., Gaonkar, M.N.: Extractive text summarization by feature-based sentence extraction using rule-based concept. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp. 1364–1368 (2017). https://ieeexplore.ieee.org/document/8256821
23. Nakov, P., et al.: SemEval-2017 task 3: community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 27–48. Association for Computational Linguistics (2017). http://www.aclweb.org/anthology/S17-2003
24. Nallapati, R., Zhai, F., Zhou, B.: 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), pp. 3075–3081. AAAI Press (2017). http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14636
25. NII: Yahoo! Chiebukuro Data, 2nd edn (2018). https://www.nii.ac.jp/dsc/idr/en/yahoo/
26. Romeo, S., et al.: Neural attention for learning to rank questions in community question answering. In: Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), pp. 1734–1745. The COLING 2016 Organizing Committee (2016). http://aclweb.org/anthology/C16-1163
27. Shimizu, T., Shimizu, N., Kobayashi, H.: Pretraining sentiment classifiers with unlabeled dialog data. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), pp. 764–770. Association for Computational Linguistics (2018). http://www.aclweb.org/anthology/P18-2121
28. Surdeanu, M., Ciaramita, M., Zaragoza, H.: 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), pp. 719–727. Association for Computational Linguistics (2008). http://www.aclweb.org/anthology/P08-1082

29. Tamura, A., Takamura, H., Okumura, M.: Classification of multiple-sentence questions. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 426–437. Springer, Heidelberg (2005). https://doi.org/10.1007/11562214_38
30. Wong, K.F., Wu, M., Li, W.: Extractive summarization using supervised and semi-supervised learning. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008), pp. 985–992. COLING 2008 Organizing Committee (2008). http://www.aclweb.org/anthology/C08-1124
31. Yahoo Japan Corp.: Yahoo! Chiebukuro (2019). https://chiebukuro.yahoo.co.jp/