NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001302

Omuokdlb at the NTCIR-17 QA Lab-PoliInfo-4 Task

Hidenori Yamato
Osaka Metropolitan University
Japan
sb22704r@st.omu.ac.jp

Takaaki Fukunaga
Osaka Metropolitan University
Japan
sb22958c@st.omu.ac.jp

Makoto Okada
Osaka Metropolitan University
Japan
okada@omu.ac.jp

Naoki Mori
Osaka Metropolitan University
Japan
mnao@omu.ac.jp

ABSTRACT

The omuokdlb team participated in two subtasks in NTCIR 17 QA Lab-Poliinfo-4: Question Answering-2 and Answer Verification. In Question Answering-2, we use Bidirectional Encoder Representations from Transformers (BERT) to match the question summary and the answer utterances. Then, we generated a summary of the answer to the question by using Text-to-text Transfer Transformer (T5). In Answer Verification, we created binary classifiers using BERT to determine whether or not answers, and we confirmed the effectiveness of the combination of the training data.

KEYWORDS

BERT, Sentence-BERT, T5

TEAM NAME

omuokdlb

SUBTASKS

Question Answering-2, Answer Verification

1 INTRODUCTION

The omuokdlb participated in Question Answering-2 and Answer Verification tasks of the NTCIR-17 QA Lab-PoliInfo-4 Task[3]. Question Answering-2 aims to generate a summary of answers by inputting a summary of questions and minutes. This task can be divided into two steps. The first step is the Alignment step. Figure 1 shows an overview of the Alignment step. This step is similar to the QA Alignment in NTCIR 16 QA-Lab Poliinfo-3[2]. In Tokyo Metropolitan assembly, a single questioner asks multiple questions, and a respondent answers only his or her part of the question, so a series of statements by a single person may include statements on multiple topics. Therefore, it is necessary to extract the relevant parts of the correct answers in order to obtain a summary of the correct answers. In this step, we match the statements that best answer the question summary. We propose a method to obtain a vector representation suitable for this task using Bidirectional Encoder Representations from Transformers (BERT)[1]. In this task, the question summary is already given. Therefore, we match the question summary and the answer paragraph directly. The next step is the Summarization step. Figure 2 shows an overview of the Summarization step. By summarizing the relevant parts of the answers obtained in the previous step, we generate a summary of the answers, which is the original purpose of this step. We used Text-to-Text Transfer Transformer (T5)[4] to generate the summary.

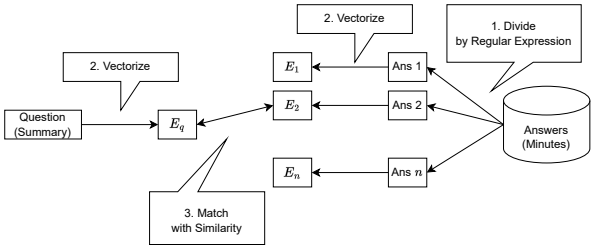

Figure 1: Overview of Alignment step

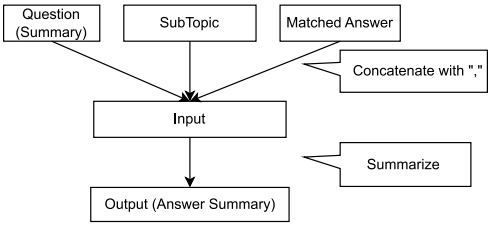

Figure 2: Overview of Answer summarization step

In Answer Verification task, it is necessary to determine whether or not an answer is appropriate to the question for given as evaluation data. In order to make this judgment, we created binary classifiers using BERT. We investigated the difference of the combination of the training data influenced accuracy of the binary classifier trained by the data.

2 QUESTION ANSWERING-2

2.1 Method

2.1.1 Making paragraphs. Paragraph division is used to divide sentences into topic-based groups. Since assembly is subject to topic transition using standardized expressions, regular expressions based on heuristic rules are effective for segmentation. In this study, we used the regular expression rules proposed by ditlab [6] in QA-Lab Poliinfo-3.

2.1.2 Vectorization. Next, the question and answer sentences obtained from the segmentation were converted to vectors. The vectors were obtained by mean pooling in BERT, which is a pooling method that averages the embedded representation of all tokens in the final layer of BERT in each dimension. We used pretrained BERT model published by Tohoku University[1]. We adopted Sentence-BERT (SBERT) [5] as the fine-tuning method to obtain features suitable for this task. The training data used was the Gold data of the QA Alignment task of NTCIR 16 QA Lab-PoliInfo-3. [2]. The data was processed to allow fine tuning by SBERT. We created three sets of (anchor, positive, negative), where the question is the anchor, the answer corresponding to the question is the positive, and the answer not corresponding to the question is the negative. This was done for all possible combinations in each minutes of the training data. By using Triplet Loss [7] as the loss function, the correct question-answer pairs were learned to be close together in the space of embedded representations, while the incorrect question-answer pairs were learned to be far apart. By learning in this way, we could obtain a sentence vector specialized for the proceedings of Tokyo Metropolitan assembly. Since the questions and answers mention the same topic, we expected the sentence vectors of the questions and answers to be close in the vector space. For comparison, the following three input methods were used during fine tuning as follows;

- Input 1 : SubTopic + Answer utterances (S + A)
- Input 2 : Only Answer utterances (A)
- Input 3 : SubTopic + Headline + Answer utterances (S + H + A)

2.1.3 Matching. The question-answer matching was performed by calculating the Cosine Similarity from the text vectors of the questions and answers obtained from SBERT. Cosine similarity was used for the similarity calculation, and the answer with the highest score to the question was matched as the answer to the question.

2.1.4 Summarization. As with the baseline method, the T5 model [3], which was pre-trained on public data, was used to generate the answer summaries. The input to T5 was the answers to the questions obtained from the matching section, the question summaries, and the subtopics.

2.2 Experiments

Table 1 and 2 show the experimantal parameters of Vectorization model and Summarization model.

2.3 Results and discussion

Table 3 shows the results of the Formal Run with ROUGE-1 F-score. Baseline is the task organizer's method (TO). Input 1 outperformed the baseline method, indicating that the proposed method is effective in generating effective sentence vectors for question-answer matching. On the other hand, Input 2 and Input 3 were lower than the baseline method. From the results of Input 2, the removal of SubTopic decreased the score. From the results of Input 3, we can see that the addition of the Headline decreased the score. Since the Headline mentions the overall topic, it is thought to have become noise for the subtopics, which are more detailed topics. Table 4 shows The results of the Formal Run by manual evaluation. 90% (90 / 100) of the sentences were rated A or B for Content, which is consistent with the matching between questions and answers. This result confirms that the generation of sentence vectors was effective in matching questions and answers. On the other hand, the most common content errors in the generated sentences were numerical and positional errors, which may have contributed to the lower scores.

Table 1: Training parameters (Vectorization model)

Parameter	Value
Max Input Length	512
Pooling Method	mean pooling
Number of Epochs	5
Optimizer	AdamW
Learning Rate	2.0×10^{-5}
Loss	Triplet Loss
Batch Size	16

Table 2: Experimental parameters (Summarization model)

Parameter	Value
Max Input Length	1024
Max Output Length	64
Number of Epochs	6
Optimizer	AdamW
Learning Rate	3.0×10^{-4}
Loss	Cross-Entropy Loss
Batch Size	2
Temperature	1.0
Repetition Penalty	1.5

Table 3: Scores in QA2 (ROUGE-1 F-score)

Input Methods	ROUGE-1
Input 1 (S+A)	0.3130
Input 2 (A)	0.2600
Input 3 (S+H+A)	0.2638
Baseline	0.2736

Table 4: Scores in QA2 (human evaluation)

	Correspondence	Content	Well-formed	Overall
A	84	32	86	49
B	12	58	10	32
C	4	10	4	23
Score	180	112	182	130

[1]https://huggingface.co/cl-tohoku
[2]https://github.com/poliinfo3/PoliInfo3-FormalRun-QAAlignment
[3]https://huggingface.co/sonoisa/t5-base-japanese

Table 5: Experimental parameters

Learning Rate	1.0×10^{-5}
Number of Epoch	10
Input Length	512
Batch Size	8
Optimizer	Adam

3 ANSWER VERIFICATION

3.1 Method

In the Answer Verification task, it is necessary to determine whether the answer is appropriate or not for the question given as evaluation data. To confirm it, we created several binary classifiers using BERT pre-trained by Tohoku University and investigated which combination of the learning data is effective to create a better classifier.

3.2 Experiments

In the experiment, BERT, a pre-trained model, was given the minute data and fine-tuned to determine correctness. The minutes data were input into BERT in four different ways. Figure 3 shows an overview of the experimental model. To construct the experimental models, we created four types of data as follows;

- input1 : QuestionSummary + AnswerSummary & QuestionSummary + AnswerOriginal (Q + A & Q + Ori)
- input2 : QuestionSummary + AnswerSummary + AnswerOriginal (Q + A + Ori)
- input3 : QuestionSummary + AnswerSummary + Headlines (Q + A + Head)
- input4 : QuestionSummary + AnswerSummary + SubTopics (Q + A + Sub)

In input1, we prepared two types of data, one combining QuestionSummary and AnswerSummary data and the other combining QuestionSummary and AnswerOriginal data, and trained them together. In input2 to input4, three sets of data were prepared and trained. The segment embedding of the first data was set to 1, the segment embedding of the second data to 0, and the segment embedding of the third data to 1, so the three data sets could be distinguished correctly. This training data was then used for 4-fold cross-validation. In the cross-validation, the training data was divided 3:1, with 3/4 of the data used for training and 1/4 of the data used for validation (this is called the validation data). Table 5 shows the various parameters used in training. For each training, the classification accuracy and model of the validation data are stored at each epoch, and The model with the highest classification accuracy at the end of each training was used as the model for that training. After completing the cross-validation, the test data were classified with each of the four models obtained for each training, and the experimental results were obtained. Table 6 shows the mean and variance of Accuracy, F1, Precision, and recall obtained in the experiments.

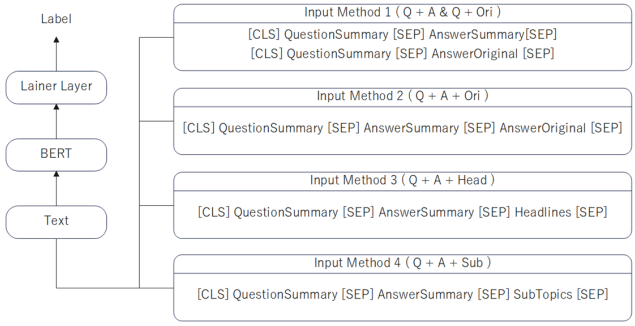

Figure 3: Overview of experimental model

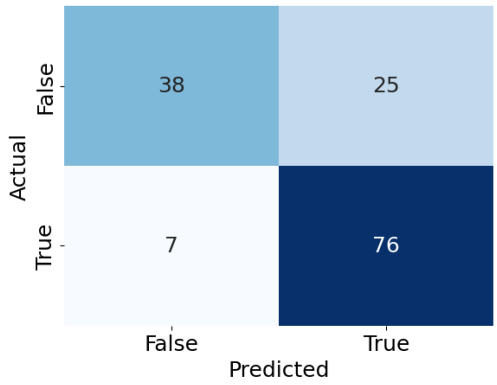

Figure 4: Confusion matrix of prediction results at Q + A & Q + Ori

3.3 Results and discussion

Table 6 shows the experimental results for each input method. The accuracies of the input methods that contained AnswerOriginal data were better than other methods.

In addition, the error rate of the fake data classified as the fact data was higher than the error rate of the fact data as the fake data. Figures 4 to 7 show the confusion matrix of prediction results for each input method. Figures 4 to 7, the vertical axis is the actual label, and the horizontal axis is the label predicted by the machine. The darker the color, the higher the value.

The discrimination results for the fake data showed that the system was able to discriminate between grammatically correct and incorrect data, however, there were many errors in the discrimination of grammatically correct data.

In the future, we will devise other experimental models that can effectively utilize all the information from AnswerOriginal, Headlines, and SubTopic, and investigate which part of the sentence is used as the basis for the prediction by machine learning.

4 CONCLUSION

We challenged two tasks in NTCIR-17 QA Lab-Poliinfo-4; Question Answering-2 and Answer Verification. In both tasks, our proposed methods better results than baseline methods. The future work of

Table 6: Experimental results for each input method

Input Method	Q + A & Q + Ori	Q + A + Ori	Q + A + Head	Q + A + Sub
$\text{Accuracy}_{\text{mean}}$	0.7483	0.7466	0.6969	0.7021
$\text{Accuracy}_{\text{variance}}$	4.446×10^{-4}	6.568×10^{-4}	1.994×10^{-4}	9.539×10^{-4}
F1_{mean}	0.7983	0.8059	0.6903	0.6820
$\text{F1}_{\text{variance}}$	7.289×10^{-4}	3.978×10^{-4}	2.893×10^{-4}	9.892×10^{-5}
$\text{Precision}_{\text{mean}}$	0.7432	0.7153	0.8618	0.8963
$\text{Precision}_{\text{variance}}$	8.945×10^{-4}	1.583×10^{-3}	$6.594 \times 10^{-}$	9.507×10^{-3}
$\text{Recall}_{\text{mean}}$	0.8594	0.9380	0.7628	0.7695
$\text{Recall}_{\text{variance}}$	7.452×10^{-4}	6.347×10^{-3}	9.743×10^{-5}	1.565×10^{-3}

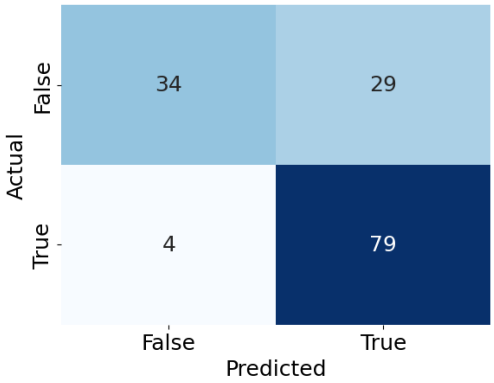

Figure 5: Confusion matrix of prediction results at Q + A + Ori

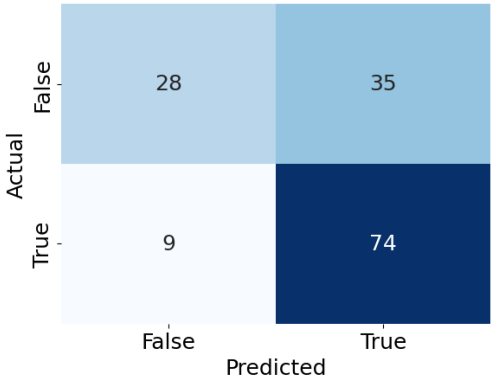

Figure 6: Confusion matrix of prediction results at Q + A + Head

Question Answering-2 is to propose the method to divide paragraphs without heuristic rules by using language models. Answer Verification is to devise other experimental models that can effectively utilize all the information from AnswerOriginal, Headlines, and SubTopic, and investigate which part of the sentence is used as the basis for the prediction by machine learning.

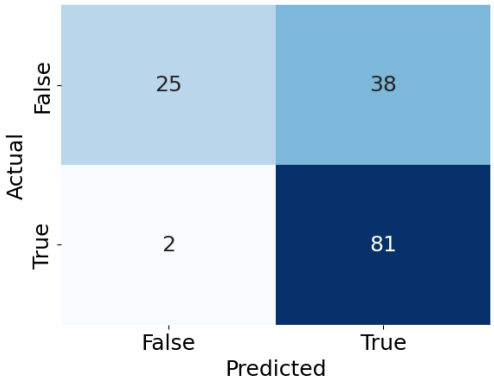

Figure 7: Confusion matrix of prediction results at Q + A + Sub

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[2] Yasumoto Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Kazuma Kadowaki Tatsunori Mori, Kenji Araki, Teruo Mitamura, and Satoshi Sekine. 2022. Overview of the NTCIR-16 QA Lab-PoliInfo-3 Task. *The 16th NTCIR Conference Evaluation of Information Access Technologies* (2022), 156–174.

[3] Yasuhiro Ogawa, Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Kazuma Kadowaki, Tomoyoshi Akiba, Minoru Sasaki, Akio Kobayashi, Masaharu Yoshioka, Tatsunori Mori, Kenji Araki, Satoshi Sekine, and Teruko Mitamura. 2023. Overview of the NTCIR-17 QA Lab-PoliInfo-4 Task. *Proceedings of The 17th NTCIR Conference.* https://doi.org/10.20736/0002001326

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (jan 2020), 67 pages.

[5] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[6] Yuuki Tachioka and Atsushi Keyaki. 2022. ditlab at the NTCIR-16 QA Lab-Poliinfo-3. *proceeding of The 16th NTCIR Conference* (2022), 207–210.

[7] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning Fine-Grained Image Similarity with Deep Ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1386–1393. https://doi.org/10.1109/CVPR.2014.180