ABSTRACT

The ditlab team participated in the QA Alignment and Question Answering task of the NTCIR-16 QA Lab-PoliInfo-3 task. First, we developed a QA Alignment system that associates each question with its answer by using heuristic rules to make paragraphs composed of related sentences and then matches them. Heuristic rules were optimized for government minutes. We prepared four types of features for matching. Second, we built a QA system that uses a similarity measure to find the original question of which contents are similar to that of the question summary. It then identifies the answers associated with the original question by using the results of the QA Alignment described above. A Text-to-Text Transfer Transformer (T5) was used to summarize the associated answer.

KEYWORDS

BM25, BERT, Gale-Shapley algorithm, T5

TEAM NAME

ditlab

SUBTASKS

QA Alignment task (Japanese)
Question Answering task (Japanese)

1 INTRODUCTION

The ditlab team participated in the QA Alignment and Question Answering tasks of the NTCIR-16 QA Lab-PoliInfo-3 task [3]. First, we proposed heuristic rules to make a paragraph composed of related sentences for question and answer. We prepared three types of features to calculate similarities between question and answer paragraphs: BM25 [6], Bidirectional Encoder Representations from Transformers (BERT) [1], and Wikipedia2Vec [7].

In this task, government minutes are composed of sentences. Each sentence has a Q/A/O tag. For the QA Alignment subtask, it is necessary to make paragraphs and associate a question with its answer. QA Alignment is performed in three steps (shown in Fig. 1). First step finds the corresponding part from the entire minutes by date and questioner ID. Second step combines multiple related sentences with "Q" or "A" tags to form a paragraph. Third step matches question and answer paragraphs based on the similarities between them.

We also developed a QA system that utilizes the results of the QA Alignment, as shown in Fig. 2. The first step is associating a question summary (input of the system) with the original question asked in the Tokyo Metropolitan Assembly. Then, the corresponding answer of the original question is identified. Lastly, the answer summary is generated using a summarizer.
2 METHODS (QA ALIGNMENT)

2.1 Heuristic rules to make paragraphs

We can accurately combine sentences by regular expressions that are optimized for the minutes because questions and answers in the Diet have a fixed format.

2.1.1 Fixed phrases at the beginning or the ending of the sentence that start paragraphs. New paragraphs start when the pattern

\[ r^* \text{match begins} \]

\[ \text{match ends} \]

\[ r^* \]

2.1.2 Fixed phrase at the ending of the sentence that terminates paragraphs. Paragraphs terminate when the pattern below matches the ending of the sentence.

\[ r^* \text{match begins} \]

\[ \text{match ends} \]

\[ r^* \]

2.2 Features for matching

2.2.1 n-gram. The baseline provided by the task organizers uses character n-grams. In addition to this, we prepared word n-grams by morphological analysis, processed by MeCab. Word n-grams are a set of morpheme n-grams excluding tokens. Similarities were calculated by counting the number of common n-grams between question and answer paragraphs. For the three features below, we used the same word n-grams.

2.2.2 BM25. BM25 models [6] were constructed on the morphemes excluding tokens, auxiliary verbs, and post-positional particles. BM25 values are high-dimensional sparse vectors that only have BM25 values at the existing morpheme. Cosine similarities between sparse vectors were used.

2.2.3 BERT. BERT [1] converts words in the sentence into embedded vectors. At the beginning of the sentence, a special token “CLS” is added, which represents the meaning of the sentence. The effectiveness of this vector was shown in the document classification task [1]. Here, cosine similarities between vectors corresponding to the “CLS” token were used.

2.2.4 Wikipedia2Vec. Wikipedia2Vec [7] can acquire the embedded vectors of words and entities appearing in Wikipedia by considering their similarities. It has been widely used for various tasks and particularly for QA tasks [4] because of its higher performance than word2vec. To convert word-wise vectors into a paragraph-wise vector, we took their average and then used the cosine similarities between averaged vectors.

2.3 Matching algorithm

We used the hospital and resident [2] matching algorithm, which is the most basic algorithm, for matching question and answer paragraphs.

3 METHODS (QUESTION AND ANSWERING)

We took the approach of generating an answer summary from the original answer in minutes. Thus, we first tried to find the original question asked in the Tokyo Metropolitan Assembly from a question summary as input, and then used the results of QA Alignment to find the original answer.

3.1 Associating a question summary with the original question

Again, we used MeCab to tokenize both a question summary and candidate questions of the original question. For the calculation

\[ \frac{1}{\text{atom}_{0, t}} * \text{word}_{0, t} \]

\[ \frac{1}{\text{atom}_{0, t}} * \text{word}_{0, t} \]

\[ \frac{1}{\text{atom}_{0, t}} * \text{word}_{0, t} \]
of the similarity between a question summary and questions, we reused the n-gram, BM25, and BERT features introduced in section 2.2. We add the Jaccard index as a new (fourth) feature. Additionally, we applied two-stage retrieval using BM25 and BERT. We first filtered top-ranked questions in descending order of BM25 scores and then reranked them by BERT.

3.2 Summarizing the answer
We utilized a commonly used transfer learning model, Text-to-Text Transfer Transformer (T5)[5], as the model of the summarizer. The pre-trained model of the summarizer was sonoisa/45-base-japanese trained with a 100 GB Japanese corpus. We fine-tuned the model using answer–answer summary pairs extracted from the training data.

4 EXPERIMENTS (QA ALIGNMENT)

4.1 Experimental setup
We evaluated the performance by using “formal run” of the NTCIR-16 QA Lab-PoliInfo-3. BM25 models were trained using the distributed data3 for “Himawari” derived from the minutes of the plenary session and budget committee of the national Diet. For BERT, we used a pre-trained Japanese model4. To obtain useful embedded vectors for document classification, it is necessary to fine-tune the pre-trained model with document classification tasks. We fine-tuned the pre-trained model using a news classification task5 because we could not prepare an appropriate document classification task for this application. Wikipedia2Vec was trained according to the procedure in the website6. We prepared 100 and 300-dimensional vectors.

4.2 Results and discussion
Table 1 lists the scores in terms of the way of making paragraphs. The feature for matching was a character n-gram for both cases. Refinement of the heuristic rules was very effective, demonstrating a 13-point improvement in the F-value score.

Table 1: Scores in terms of way of making paragraphs.

<table>
<thead>
<tr>
<th></th>
<th>F-value</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.6166</td>
<td>0.5991</td>
<td>0.6437</td>
</tr>
<tr>
<td>New heuristic</td>
<td>0.7458</td>
<td>0.7606</td>
<td>0.7349</td>
</tr>
</tbody>
</table>

Table 2: Scores in terms of features for matching.

<table>
<thead>
<tr>
<th></th>
<th>F-value</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>word n-gram</td>
<td>0.7677</td>
<td>0.7861</td>
<td>0.7533</td>
</tr>
<tr>
<td>BM25</td>
<td>0.8348</td>
<td>0.8739</td>
<td>0.8045</td>
</tr>
<tr>
<td>BERT</td>
<td>0.5968</td>
<td>0.6187</td>
<td>0.5799</td>
</tr>
<tr>
<td>W2V (100 dim)</td>
<td>0.6821</td>
<td>0.7139</td>
<td>0.6575</td>
</tr>
<tr>
<td>W2V (300 dim)</td>
<td>0.7382</td>
<td>0.7659</td>
<td>0.7160</td>
</tr>
</tbody>
</table>

Table 3: Scores in terms of QA.

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-1 F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>word n-gram</td>
<td>0.2992</td>
</tr>
<tr>
<td>Jaccard index</td>
<td>0.2732</td>
</tr>
<tr>
<td>BM25</td>
<td>0.3013</td>
</tr>
<tr>
<td>BERT</td>
<td>0.1423</td>
</tr>
<tr>
<td>BM25 + BERT</td>
<td>0.1715</td>
</tr>
</tbody>
</table>

Wikipedia2Vec (W2V in the table) was worse than n-gram, the performance using a 300-dim vector outperformed that using a 100-dim vector, which demonstrates that a higher dimensional vector is effective.

5 EXPERIMENTS (QUESTION AND ANSWERING)

5.1 Experimental setup
We also evaluated the performance of the QA task by using a “formal run” of the NTCIR-16 QA Lab-PoliInfo-3. BM25 and BERT models were the same as in the previous experiment.

5.2 Results and discussion
The results in Table 3 show that BM25 performed the best among the compared methods. BERT did not work as well as for the QA Alignment task. Similarly, the two-stage retrieval approach (BM25 + BERT) was worse than BM25 only.

6 CONCLUSION

6.1 QA Alignment
In order to associate each question with its answer, we refined heuristic rules that make a paragraph and prepared three types of features for matching question and answer paragraphs. The refinement of heuristic rules improved the F-value by 13 points. Word n-gram and BM25 improved the F-value by 2.2 and 8.9 points, respectively. BERT and Wikipedia2Vec degraded the performance because training data were inappropriate.

6.2 Question and answering
We generated an answer summary from the original answer in minutes. In this method, we first find the original question from a question summary using similarity calculation, then identify the answer using the results of QA Alignment, and lastly summarize the answer with T5 to generate an answer summary. Experimental results showed that BM25 was the best term-weighting scheme.
REFERENCES


