UHGSIS at the NTCIR-16 Data Search 2 IR Subtask: A BERT-based Query Modification Approach

Moeri Okuda
University of Hyogo
Japan
ad21v034@gsis.u-hyogo.ac.jp

Ryota Mibayashi
University of Hyogo
Japan
aa20r511@ai.u-hyogo.ac.jp

Takafumi Kawahara
University of Hyogo
Japan
aa20m503@ai.u-hyogo.ac.jp

Naoaki Matsumoto
University of Hyogo
Japan
aa20y510@ai.u-hyogo.ac.jp

Kenji Tanaka
University of Hyogo
Japan
k5183tanaka@gmail.com

Takehiro Yamamoto
University of Hyogo
Japan
t.yamamoto@sis.u-hyogo.ac.jp

Hiroaki Ohshima
University of Hyogo
Japan
ohshima@ai.u-hyogo.ac.jp

ABSTRACT

We describe a framework using the BERT-based query modification technique for the NTCIR-16 Data Search 2 IR Subtask. In our framework, we took a 3-step procedure: (1) the query modification, (2) item filtering by BM25, and (3) item re-ranking by BERT. The experimental results showed that our framework using the query modification did not outperform the baseline method that does not use the query modification.

KEYWORDS

Query Modification, BM25, BERT

TEAM NAME

UHGSIS

SUBTASKS

IR Subtask (Japanese)

1 INTRODUCTION

The UHGSIS team participated in the NTCIR-16 Data Search 2 IR Subtask. The IR subtask is a standard ad-hoc retrieval task [2]. In this task, the system is expected to search the description from one query. To tackle this problem, we incorporated the query modification, item filtering by BM25, and item re-ranking by BERT. We propose a general framework for the IR subtask, which first split the query into some words per every single space and extract important keywords, then rank descriptions using BM25 [4], and finally re-rank descriptions using BERT [1].

2 DATASETS

This task has two kinds of data. One is the statistical data collection published by the Japanese government (e-Stat), and the other is the statistical data collection published by the US government (Data.gov). The IR Subtask consists of the Japanese Subtask and the English Subtask. We worked only on the Japanese Subtask.

Figure 1: Overview of the framework

These data collections are tabular formats and have some attributes. In addition to these collections, topics and query are given. Topic is a question-answer crawl of the question-answer pairs including the link to e-Stat in the Japanese question-and-answer

```
{
"id": "000031519435",
"url": "https://www.e-stat.go.jp/stat-search/files?page=1&toukei=00200231
        &result_page=1&layout=dataset&stat_infid=000031519435",
"attribution": "出典：政府統計の総合窓口(e-Stat)（https://www.e-stat.go.jp/）",
 "title": "地方公共団体の議会の議員及び長の所属党派別人員調 地方公共団体の
          議会の議員及び長の所属党派別人員調等（H20.12.31現在）
          選挙執行件数 | ファイルから探す | 統計データを探す | 政府統計の総合窓口",
"description": "地方公共団体の議会の議員及び長の所属党派別人員調 /
                地方公共団体の議会の議員及び長の所属党派別人員調等（H20.12.31現在）",
"data": [
      {
         "data_format": "xls",
         "data_url": "https://www.e-stat.go.jp/stat-search/file-download?statInfId=000031519435&fileKind=0",
         "data_filename": "a0abc17d8ce4715933c69132418dc7337e76c5aad06beb9f5d69b0f1c1870ff9-05%
                           E9%81%B8%E6%8C%99%E5%9F%B7%E8%A1%8C%E4%BB%B6%E6%95%B0%E8%AA%BF.xls"
      }
   ],
"data_fields": {
    "提供統計名": "地方公共団体の議会の議員及び長の所属党派別人員調等（H20.12.31現在）",
    "統計表名": "選挙執行件数", "担当機関": "総務省",
    "データセットの概要": "",
    "政府統計名": "地方公共団体の議会の議員及び長の所属党派別人員調",
    "公開年月日時分": "2017-01-06 11:34",
    "集計地域区分": "該当なし"
   }
}
```


Figure 2: Example of a Japanese data format

service Yahoo!. The example of a Japanese dataset is shown in Figure2.

3 BASELINE METHOD

In this section, we describe the baseline method that consists of the following 2 steps;

- item filtering by BM25,
- item re-ranking by BERT.

The input is the query. In item filtering by BM25, descriptions are ranked by using the BM25 score. In some cases, the top-k descriptions ranked by item filtering by BM25 are extracted and used for item re-ranking by BERT. In item re-ranking by BERT, descriptions are re-ranked by using the output score of BERT. The output is the conformity score. The conformity score is the score that indicates how well the query and the description match.

3.1 Item filtering by BM25

In this section, we describe BM25 to filter the query and the description. BM25 is based on Elasticsearch. Elasticsearch is a search engine software developed by Elastic. In this study, We used Elasticsearch to convert the query or the description into the index, and then searched for the word. The Elasticsearch was performed on the query and the description in the data. If the word is included in the query or the description, the words are sorted by the score. The results of each search were combined and sorted by score. For example, there is a query consisting of multiple words, such as "Tokyo, 2020, apple". In this case, if either "Tokyo", "2020", or "apple" is included, the query is included.

We used kuromoji [1] as the tokenizer for Elasticsearch. Kuromoji is the morphological analysis engine for Japanese.

3.2 Item re-ranking by BERT

In ranking, it is necessary to calculate the score that indicates how well the query and the description match. This score is called the conformity score.

The inputs are the query and the description. The datasets of the combination of the query and the description are randomly divided into training, validation, and test sets by 80%, 10%, and 10%, respectively. Table 1 describes the number of the combination of the query and the description used in each set. The total datasets are 1,338,402, and the numbers of training, validation, and test set are 1,003,801, 167,301 and 167,300.

Table 1: Division of images into train, validation, and test sets

	Total	Train(80%)	Validation(10%)	Test(10%)
datasets	1,338,402	1,003,801	167,301	167,300

The output is the conformity score. In this study, we use pre-trained BERT as the method to compute the score. We fine-tuned BERT as our framework.

The conformity score between the query and the description is represented by the labels "L0", "L1", or "L2". The labels "L0", "L1", or "L2" are ordinal measures. The score of "0", "1", or "2" corresponds to the label of "L0", "L1", or "L2".

The input needs to be segmented at the token. This division is called tokenization. The output is the conformity score. The conformity score is used for ranking.

3.3 Experimental Settings of BERT

BERT is used on item re-ranking by BERT. Loss function is the Cross-Entropy Loss. Settings of hyperparameters of the query modification and BERT are described below;

- Optimizer: Adam [3],
- Learning rate: 10^{-7},
- Batch size: 32,
- Max length: 512,
- Early stopping:
  - patience: 10.

4 FRAMEWORK

In this section, we describe our framework that consists of the following 3 steps;

- the query modification,
- item filtering by BM25,
- item re-ranking by BERT.

Our framework is depicted in Figure 1. The input is the query. In the query modification, the query is split into some words per every single space and extracted important keywords. In item filtering by BM25, descriptions are ranked by using the BM25 score. In some cases, the top-k descriptions ranked by item filtering by BM25 are extracted and used for item re-ranking by BERT. In item re-ranking by BERT, descriptions are re-ranked by using the output score of BERT. The output is the conformity score. The conformity score is the score that indicates how well the query and the description match.

4.1 Query Modification

First of all, by using BERT, we removed the words related to the year, the unit, and the place. This is because the words related to

[1] https://github.com/elastic/elasticsearch-analysis-kuromoji

Table 2: Examples of the classified classes

	"keyword"	"year"	"unit"	"place"
Example	"apple" and "orange"	" 2013"	"percentage"	"Tokyo"

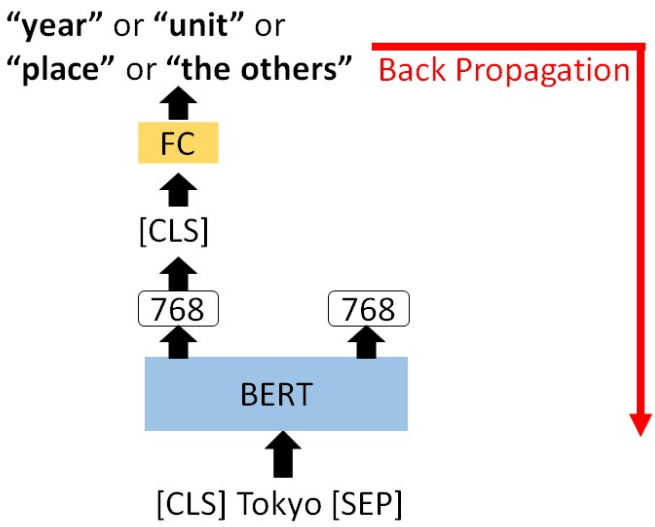

Figure 3: Overview of the query modification model of our framework

Table 3: The number of the classes

	Total	"keyword"	"year"	"unit"	"place"
Training sets	569	367	26	101	75
Test sets	278	190	9	36	43

the year, the unit, and the place are not described in the description often.

The process of the query modification is described below;

(1) The query is divided into a word for every single space.
(2) All words are classified into 4 classes: "keyword", "year", "unit", and "place".
(3) The "keyword" class is extracted.

The query was divided into some words per every single space. A divided query is called the "word". For example, given a query, [Tokyo, 2020, apple, orange], the "word" is "Tokyo", "2020", or "apple" or "orange".

Training sets of all "words" are classified into 4 classes: "keyword", "year", "unit", and "place". For example, "apple" and "orange" is the "keyword" class, "2013" is the "year" class, "percentage" is the "unit" class, and "Tokyo" is the "place" class. Table 2 shows some examples of the classified classes. We classified training sets into these 4 classes.

A query modification model is trained to classify test sets into 4 classes by using training sets. The model is shown in Figure 3. This model is based on BERT. BERT is the model to transform the text into the 768-dimensional vector. This model is provided by

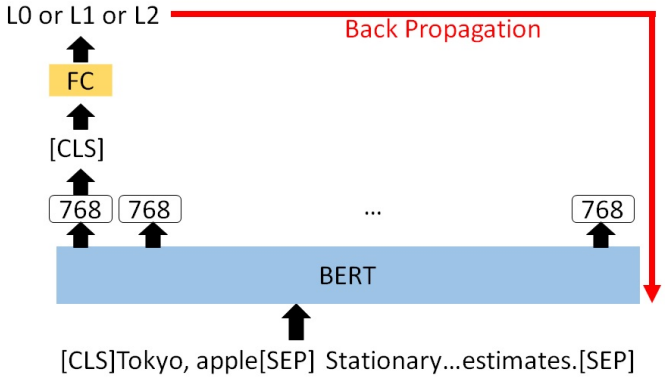

Figure 4: Overview of item re-ranking by BERT model of our framework

Inui Laboratory at Tohoku University. This model is pre-trained by Japanese Wikipedia. The output of BERT is the 768-dimensional vector of the input tokens. The input of BERT is the query between [cls] token and [sep] token. We extracted the 768-dimensional vector of [cls] token only. The fully-connected layer is added on BERT. The output is "keyword", "year", "unit", or "place" class.

The number of training sets of "keyword", "year", "unit", or "place" class is 367, 26, 101, or 75, respectively. The number of test sets of "keyword", "year", "unit", or "place" class leads to 190, 9, 36, or 43, respectively. Table 3 shows the number of the classes of the datasets.

Finally, we extracted the "keyword" class. In our framework, we only use "words" of the "keyword" class.

4.2 Item filtering by BM25

In our framework, we use BERT for ranking. The input of BERT is usually filtered by using BM25. Ranking the input of BERT has the effect of reducing the computation time of BERT.

In this section, we describe BM25 to filter the "word" and the description. In the baseline method, We used Elasticsearch to convert the "word" or the description into the index, and then searched for the given word. The Elasticsearch was performed on the "word" and the description in the data. If the given word is included in the "word" or the description, the given words are sorted by the score. The results of each search were combined and sorted by score. For example, there is a given word, such as "Tokyo". In this case, if "Tokyo" is included in the "word" or the description, the word is included.

4.3 Item re-ranking by BERT

In this section, we describe the method to rank the filtered documents. Figure 4 shows the fine-tuning overview by BERT.

The inputs are the "word" and the description. The output is the conformity score. The conformity score is used for ranking. In this study, we use pre-trained BERT as the method to compute the conformity score. We fine-tuned BERT as our framework.

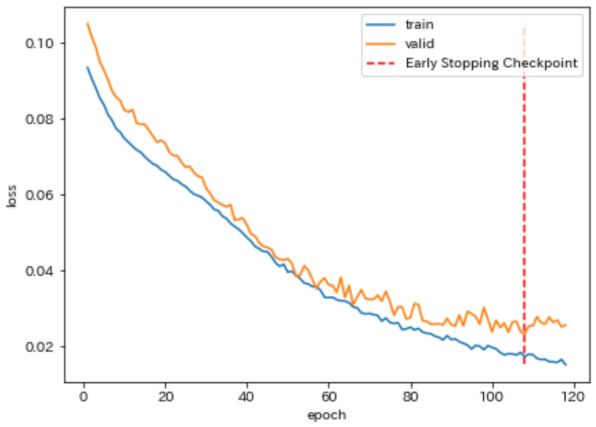

Figure 5: Training loss value of BERT for the query modification of our framework

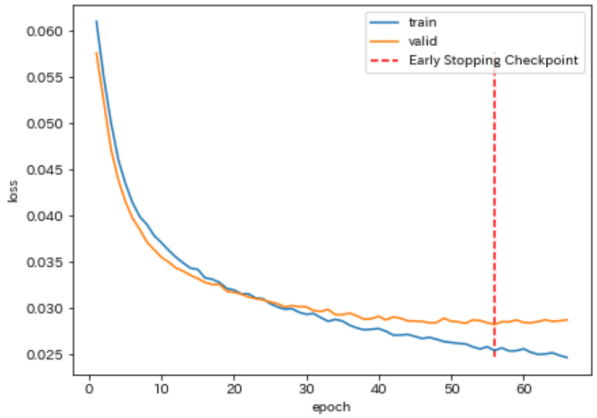

Figure 6: Training loss value of BERT for item re-ranking by BERT of our framework

4.4 Experimental Settings of BERT

BERT is mainly used on the query modification and item re-ranking by BERT. Loss function is the Cross-Entropy Loss. Settings of hyperparameters of the query modification and item re-ranking by BERT are described below;

- Optimizer: Adam [3],
- Learning rate: 10^{-7},
- Batch size: 32,
- Max length: 512,
- Early stopping:
  - patience: 10.

The early stopping of the query modification is applied at 108 epochs. Figure 5 shows the loss value for training.

The early stopping of BERT is applied at 57 epochs. Figure 6 shows the loss value for training.

5 RESULTS AND DISCUSSION

Table 4 shows the results of the submitted method that uses the query modification and the submitted method that does not use the query modification. On all of the measured scores, the method with the query modification is lower than the method without the query modification.

We proposed the method using the query modification, item filtering by BM25, and item re-ranking by BERT. Table 5 shows the used methods of our submissions. UHGSIS-J-2, 4, 6, 8, 10 do not use the query modification. UHGSIS-J-1, 3, 5, 7, 9 use the query modification.

The experimental results show our framework that uses the query modification did not outperform the baseline method that does not use the query modification. In general, query modification tends to improve the performance. However, in our framework, query modification decreases the performance. The reason was that the "place" class could be important. For example, the results are very different when "Tokyo" is used as the query and when "Kyoto" is used as the query.

6 CONCLUSIONS

In this paper, we described our information retrieval framework in the NTCIR-16 Data Search Task. Our framework is the method based on query modification, item filtering by BM25, and item re-ranking by BERT. In the query modification, the query is split into some words per every single space and extracted important keywords. In item filtering by BM25, descriptions are ranked by using the BM25 score. In some cases, the top-k descriptions ranked by item filtering by BM25 are extracted and used for item re-ranking by BERT. In item re-ranking by BERT, descriptions are re-ranked by using the output score of BERT. The output is the conformity score. The conformity score is the score that indicates how well the query and the description match.

Our method using the query modification did not outperform the baseline method that does not use the query modification. The reason is likely not to use the "place" class. For example, the results are very different when "Tokyo" is used as the query and when "Kyoto" is used as the query.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (2019), 4920–4928.

[2] P. Makoto Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. 2022. Overview of the NTCIR-16 Data Search 2 Task. In *Proceedings of the 2022 NTCIR Conference on Evaluation of Information Access Technologies.*

[3] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 2015 International Conference on Learning Representations.*

[4] Stephen Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 2000. Experimentation as a Way of life: Okapi at TREC. *Information Processing and Management* (2000), 95–108.

Table 4: Evaluation Results of Japanese Subtask

Run	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-measure
UHGSIS-J-2, 4, 6, 8, or 10 (**without the query modification**)	0.237	0.241	0.260	0.186	0.257	0.268	0.279
UHGSIS-J-1, 3, 5, 7, or 9 (with the query modification)	0.213	0.220	0.234	0.164	0.230	0.243	0.252

Table 5: Used Methods

Submission Name	Query modification	Item filtering by BM25	Item re-ranking by BERT	The number of descriptions by item filtering
UHGSIS-J-1	✓	✓	✓	All
UHGSIS-J-2		✓	✓	All
UHGSIS-J-3	✓	✓		All
UHGSIS-J-4		✓		All
UHGSIS-J-5	✓	✓	✓	Top3000
UHGSIS-J-6		✓	✓	Top3000
UHGSIS-J-7	✓	✓	✓	Top2000
UHGSIS-J-8		✓	✓	Top2000
UHGSIS-J-9	✓	✓	✓	Top1000
UHGSIS-J-10		✓	✓	Top1000