Uhai at the NTCIR-15 Data Search Task

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

Pham HuuLong
University of Hyogo
aa19e510@ai.u-hyogo.ac.jp

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

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

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

ABSTRACT

In this paper, we present our approach and results of the NTCIR-15 Data Search Task. The NTCIR-15 Data Search Task is a shared task aimed at improving the information retrieval techniques. As the first round of the task, the organizers have set up a problem of ad-hoc information retrieval on government data. The task is divided into two sub-tasks: English and Japanese. Each of them uses statistical data from the government agencies.

We have approached both tasks in two major ways: query modification and learning to rank. In addition to these approaches, we also tried several combinations of them.

As a result, the English subtask ranked 4th out of all the teams and 5th in the order of submission and the Japanese subtask ranked 3rd out of all teams and 4th in the order of submission by the evaluation metric of nDCG@10.

TEAM NAME

uhai

SUBTASKS

Japanese subtask, English subtask

KEYWORDS

BERT, Query Modification, Learning to Rank

1 INTRODUCTION

The NTCIR-15 Data Search Task is a shared task aimed at the improvement of the information retrieval technology. Through this task, organizers aim to improve the following three technologies.

- Query understanding for data search
- Data understanding for data search
- Retrieval models for data search

For details, please refer to Overview paper [9].

In this first round, NTCIR-15 Data Search Task performs ad-hoc information retrieval on the data published by the government.

The task is divided into English and Japanese subtasks. Each of them uses statistical data from government agencies. The Japanese subtask uses the Japanese government data (e-Stat)[1], the English subtask uses the US government data (Data.gov) [2]. The details of these data are described in Section 2.

Datasets and tools for the tasks are provided by the organizers. Participants submit the ranked lists of search results using them. There is a limit to the number of submissions, five per subtask.

[1]https://www.e-stat.go.jp/
[2]https://data.gov/

```
{
    "id": "0063664a-d0d7-4ce2-9462-0463a89fc274",
    "url": "https://catalog.data.gov/dataset/0063664a-d0d7-4ce2-9462-0463a89fc274",
    "attribution": "CRED REA Fish Team Stationary Point Count Surveys at Sarigan, Marianas Archipelago,
                  2005 (https://catalog.data.gov/dataset/0063664a-d0d7-4ce2-9462-0463a89fc274) is licensed
                  under U.S. Government Work (http://www.usa.gov/publicdomain/label/1.0/)"
    "title": "CRED REA Fish Team Stationary Point Count Surveys at Sarigan, Marianas Archipelago, 2005",
    "description": "Stationary Point Counts at 4 stations at each survey site were surveyed as part of
                   Rapid Ecological Assessments (REA) conducted at 3 sites around Sarigan in the Marianas Archipelago (MA)
                   during 3 September - 1 October 2005 in the NOAA Oscar Elton Sette (OES 0511)
                   Reef Assessment and Monitoring Program (RAMP) Cruise.
                   Raw survey data included species level abundance estimates.",
    "data": [
        {
            "data_format": "excel",
            "data_organization": "National Oceanic and Atmospheric Administration, Department of Commerce",
            "data_url": "https://data.nodc.noaa.gov/coris/data/NOAA/nmfs/pifsc/cred/REAFish/
                        CNMI_2005/CRED_REA_FISH_SAIPAN_2005.xls",
            "data_filename": "076342d026a0feec762ce5cb18e047db61e24db557958f75ca7aaa668b5e1342-CNMI_2005/
                             CRED_REA_FISH_SAIPAN_2005.xls“
        }
    ],
    "data_fields": {
        "Resource Type": "Dataset",
        "Metadata Date": "June 20, 2018",
        "Metadata Created Date": "February 7, 2018",
        "Metadata Updated Date": "February 27, 2019",
        ...
        "metadata_sources": [
            "https://catalog.data.gov/harvest/object/fc5a39b7-4c9f-49b8-af95-2812d9b3264c"
        ]
    }
}
```

Figure 1: Example of an English data format.

The participants can work on one or both of the subtasks. We worked on this task in both Japanese and English. Specially, we worked mainly on Japanese language tasks.

We detail our works in the following sections. Section 2 explains the datasets. Section 3 presents an explanation of the proposed method. Section 4,5 presents the results and a discussion of the results.

2 DATASETS

There are two kinds of data[3] in this task. 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). These data collections are given in tabular format and have some attributes. An example is shown below Figures 1 and 2.

In addition to these collections, topics and queries 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 service Yahoo!.

The queries were transformed from the extracted topics. For the conversion, we used crowd sourcing and gave topics to 10 workers. This query task seems to be performed in both Japanese and English in different tasks.

3 METHOD

In this research, we constructed a learning-to-rank model, which use scores of multiple ranking methods as learning features to produce ranked documents list given a query. In addition, we also

[3]https://ntcir.datasearch.jp/data/

```
{
"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 an Japanese data format.

proposed a method called query modification, which automatically fix user's query in order to archive better result.

3.1 Query Modification

Sometimes input query may contain unnecessary words that decrease search result's accuracy. Query Modification is our method to detect those unnecessary words and automatically remove them from the query. We constructed a Random Forest Regressor (RFR) Model [2] which predicts contributing score of a word to the search, and use the predicted score to decide whether the word itself is necessary or not. If a word's contributing score is lower than 0.4, it will be removed from the query, and new query without that word is then used for the search, instead of the original one.

Figure 4 illustrates our Query Modification method when the query is "Births by month". After contributing scores of each words is predicted, the word "by" will be removed from the query due to its low score. The query is then modified to "Births month".

Our RFR model take a word vector and output its contributing score. We used fastText for word embedding, which convert each word into a 100-dimensional vector. Both fastText's official model for English and Japanese was used.

In order to train the RFR model, true contributing score of each word is required. We used the competition's dataset to calculate contributing scores with the following steps:

(1) Separate a query from competition's dataset into multiple words.
Example: "Births by month" -> "births", "by", "month".
(2) Take the Power Set of separated words, and construct new queries.
Example: 'births", "by", "month" -> "births", "by", "month", "births by", "births month", "month by", "births by month".
(3) With each queries constructed in the previous step, use BM25 to produce a ranked documents list. Then calculate produced list's nDCG@10 score using the true list.
Example: "births": 0.653, "births by": 0.653, "births month": 0.653, "births by month": 0.653
(4) Calculate contributing score of a word to the query by dividing mean nDCG@10 score of all queries that were generated in Step 2, which include that word, by max nDCG@10 score archived in the previous step.
Example: contributed score of "births" to the query "Births by month" is calculated by:

$$\frac{mean(0.653, 0.653, 0.653, 0.653)}{0.653} = 1 \tag{1}$$

(5) Repeat the above steps with all queries in the dataset, and calculate each word contributing score by taking the mean over all queries.

Figure 5 illustrates above steps using the query "Births by month".

We then used contributing scores calculate from above steps to train RFR model. In order to find best hyper parameters of the model, we ran Grid Search with 10-Fold Cross Validation. Candidate hyper parameters are listed in Table 1. Hyper parameters which give best validation score for English and Japanese are listed in Table 2 and Table 3.

Table 1: Parameters investigated in grid search

params	values
criterion	"mse", "mae"
maxDepth	3, 5, 10, 15, 20, 25, 30, 40, 50, 100
maxfeatures	"auto", "sqrt", "log2"
minSamplesSplit	3, 5, 10, 15, 20, 25, 30, 40, 50, 100
nEstimators	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100

Table 2: Best parameters of the grid search results in English

parameter	values
criterion	"mae"
maxDepth	30
maxfeatures	"sqrt"
minSamplesSplit	10
nEstimators	45

Table 3: Best parameters of the grid search results in Japanese

parameter	values
criterion	"mse"
maxDepth	15
maxfeatures	"sqrt"
minSamplesSplit	10
nEstimators	5

3.2 Learning To Rank Model

We constructed a learning-to-rank model which return a ranking score of every query-document pair. Documents are then sorted by their ranking score to provide a ranked list. Learning-to-rank is widely used technique, especially in information retrieval [3, 8] Microsoft's LightGBM framework was used to construct the learning-to-rank model [6].

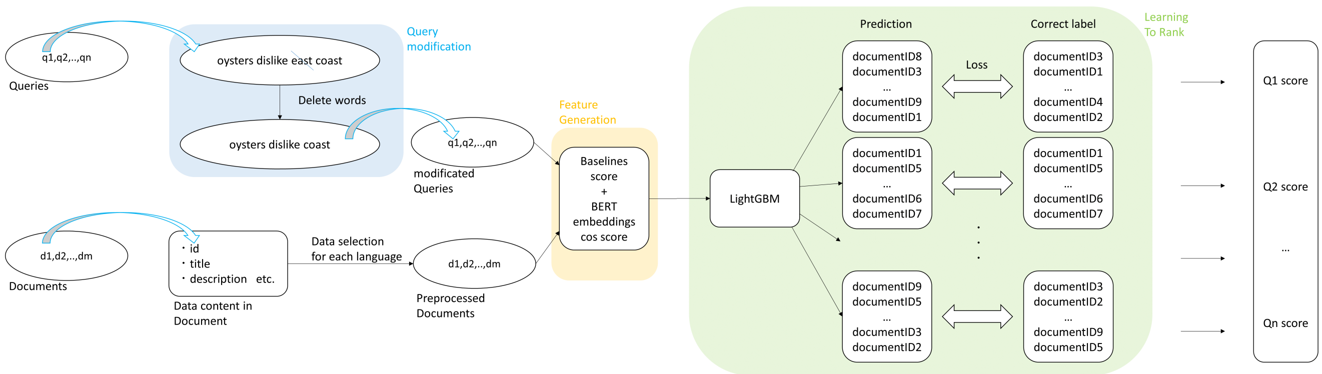

Figure 3: Our method flow

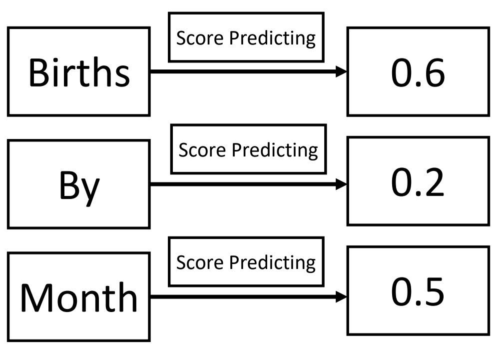

Figure 4: Example of Query Modification Method

3.2.1 Feature generation. In order to generate features for the learning to rank model, we used query-document score of multiple methods, including BERT embedding and baseline methods that were provided by the competition.

- **BERT Embedding:** We used BERT Embedding to transform queries and documents into vectors. After that, cosine similarity between query vector and document vector is calculated and used as one feature of our learning-to-rank model. There are many techniques for word or document embedding, such as word2vec [10, 11], fastText [1, 5], doc2vec [7], and glove [12]. However, in information retrieval and ranking, since contextual meaning is important, BERT was chosen. Recently, there are many studies which apply BERT in information retrieval [4, 15]. Pre-trained BERT models in each language were used. The Japanese version of BASE (normal version) of Kyoto Univer-sity is used for the Japanese subtask [4]. For morphological analysis, jumanpp [5] was used and BPE was performed by subword-nmt [6]. English version has the transformers using bert-base-uncased [13].

[4]http://nlp.ist.i.kyoto-u.ac.jp/DLcounter/lime.cgi?down=http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/JapaneseBertPretrainedModel/Japanese_L-12_H-768_A-12_E-30_BPE.zip&name=Japanese_L-12_H-768_A-12_E-30_BPE.zip
[5]https://github.com/ku-nlp/jumanpp
[6]https://github.com/rsennrich/subword-nmt

- **Baseline methods:** Table 4 lists all baseline methods that were provided by the competition and their descriptions. Each method was run by a tool called Anserini [14]. For every method, a ranking score between each pair of query-document is generated. We used those ranking scores as features of the learning-to-rank model.

3.2.2 Feature selection. We performed features selection in order to select best features for the learning-to-rank model through below steps:

(1) Take combinations of all 6 baseline methods
Example: (bm25), (bm25, qld), (bm25, qld, sdm), (qld, sdm), etc
(2) For every combination, use query-document ranking score of each baseline method as features of learning-to-rank model and perform 10-Fold Cross Validation. Grid search for hyper parameters was also performed to make sure the score is highest as possible. Candidate hyper parameters are listed in Table 5.
Example: For the combination (bm25, qld), ranking scores between each query-document calculated by both bm25 and qld are used as features.
(3) Calculate average validation nDCG@10 score of each combination. Select the combination that give best nDCG@10 score.

After performing 3 steps above, combination of sdm+qld and rm3+bm25 was recognized to give the best performance and was recruited.

Finally, we constructed four learning-to-rank models, which vary from used features, and whether query modification was performed or not. Our four models are listed in Table 6. Cosine similarity calculated between each query-document pair (both embedded by BERT) was also used as feature in two models. For each model, hyper parameters grid search is performed again in order to give the best model.

4 RESULTS

The results of our submissions of the task are shown in Tables 7 and 8. The results are sorted by nDCG@10. For both results, the combination method of query modification and BM25 scored the highest in both cases.

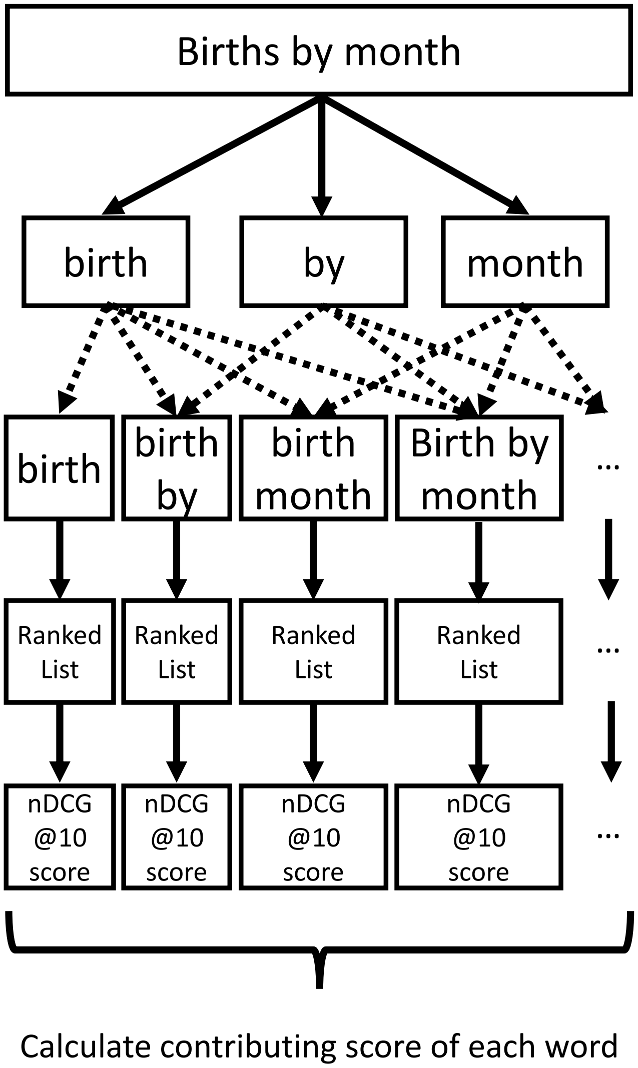

Figure 5: Example of word's contributing score calculation

Table 4: Baseline methods

name	caption
bm25	BM25 scoring model
bm25.accurate	BM25 scoring model
bm25prf	bm25PRF query expansion model
qld	query likelihood Dirichlet scoring model
rm3	RM3 query expansion model
sdm	Sequential Dependence Model

Otherwise, the results were different for each subtask. For the Japanese subtask, the BERT method tends to score higher, while query modification + L2R and L2R tend to score lower. For the English subtask, despite the fact L2R alone is the second best method, query modification + L2R tends to have a much lower score.

As a result, the Japanese subtask ranked 3rd out of all teams and 4th in the order of submission and the English subtask ranked 4th out of all the teams and 5th in the order of submission. by the evaluation metric of nDCG@10. In terms of overall participation, The number of participating teams in the Japanese subtask was 3, with 17 submissions, while the number of participating teams in the English subtask was 5, with 37 submissions.

Table 5: Parameters used for grid search

params	values
num-leaves	6, 8, 12, 16
colsample-bytree	0.4, 0.7, 1
sabsample	0.4, 0.6, 1

5 DISCUSSION

The results show that the combination of query modification + BM25 gives high scores in both English and Japanese. However, the equality in terms of nDCG@10 scores compared to the ORGJ team's submission which used only BM25 (Table 7 and 8), indicates that the query modification method did not contribute to the improvement of the score. For nDCG@3 and nDCG@5, the scores have improved, but there is no significant difference.

In addition, because of the inclusion of the scores of BM25 as features, L2R can theoretically produce the same score as BM25. However, due to the low score of the query modification + L2R method, it is possible that the features used in L2R contain some unnecessary features for the search.

Also, since query modification + L2R + BERT scored higher than query modification + L2R for both languages, The contextual information that BERT has is also effective.

In summary, among the approaches we proposed, L2R and query modification were not effective, and we showed that BERT context might be effective for information retrieval.

6 CONCLUSIONS

In this paper, we described our information retrieval method in the NTCIR-15 Data Search Task. Our method is a complex method based on query modification and learning-to-rank. As a result, our English subtask submission ranked 4th out of all the teams and 5th in the order of submission, while Japanese subtask submission ranked 3rd out of all teams and 4th in the order of submission by the evaluation metric of nDCG@10. In addition we found that the score of query modification + BM25 is the highest among our submissions. However, in the discussion, we found that query modification does not have much influence on the score. And, we found that contextual information from BERT embeddings can have a positive influence on search.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP18H03243, JP18H03494, JP18H03244, JP17H00762, JP16H01756.

Table 6: List of our submitted runs

Run ID	features	with query modification
uhai-E-1, J-6	sdm+qld, rm3+bm25, BERT cosine similarity	yes
uhai-E-2, J-7	sdm+qld, rm3+bm25	yes
uhai-E-3, J-8	sdm+qld, rm3+bm25	no
uhai-E-4, J-9	sdm+qld, rm3+bm25, BERT cosine similarity	no
Run ID	method	with query modification
uhai-E-5, J-10	BM25	yes

Table 7: Evaluation all results of Japanese subtask

submit	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-measure
KSU-J-5	0.388	0.403	0.448	0.283	0.448	0.477	0.498
KSU-J-1	0.362	0.381	0.421	0.295	0.423	0.453	0.473
ORGJ-J-3	0.407	0.413	0.421	0.325	0.450	0.470	0.484
uhai-J-10(query modification + BM25)	0.403	0.406	0.415	0.312	0.447	0.466	0.484
ORGJ-J-2(ja-bm25)	0.402	0.405	0.415	0.328	0.447	0.467	0.483
ORGJ-J-6	0.379	0.386	0.406	0.321	0.423	0.447	0.464
ORGJ-J-1	0.382	0.396	0.405	0.308	0.426	0.452	0.464
ORGJ-J-7	0.380	0.386	0.401	0.323	0.430	0.452	0.471
ORGJ-J-4	0.365	0.377	0.400	0.318	0.409	0.433	0.452
uhai-J-9(L2R + BERT)	0.369	0.382	0.393	0.301	0.417	0.441	0.461
uhai-J-6(query modification + L2R + BERT)	0.369	0.375	0.389	0.293	0.418	0.439	0.455
ORGJ-J-5	0.362	0.363	0.377	0.288	0.415	0.434	0.452
ORGJ-J-8	0.357	0.363	0.373	0.289	0.404	0.425	0.437
uhai-J-7(query modification + L2R)	0.350	0.352	0.368	0.272	0.410	0.431	0.453
uhai-J-8(L2R)	0.346	0.350	0.362	0.278	0.392	0.414	0.432
KSU-J-3	0.114	0.117	0.119	0.045	0.136	0.145	0.151
KSU-J-7	0.114	0.117	0.119	0.045	0.136	0.145	0.151

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[2] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank Using Gradient Descent *(ICML '05)*. Association for Computing Machinery, New York, NY, USA, 89–96. https://doi.org/10.1145/1102351.1102363

[4] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 985–988.

[5] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3146–3154. http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[7] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[8] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. 3, 3 (March 2009), 225–331. https://doi.org/10.1561/1500000016

[9] Kato Makoto P., Ohshima Hiroaki, Liu Ying-Hsang, and Chen Hsin-Liang. 2020. Overview of the NTCIR-15 Data Search Task.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[11] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations.. In *HLT-NAACL*. 746–751.

[12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[14] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.

[15] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An Analysis of BERT in Document Ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1941–1944. https://doi.org/10.1145/3397271.3401325

Table 8: Evaluation all results of English subtask

submit	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-measure
KSU-E-2	0.204	0.231	0.255	0.238	0.229	0.257	0.276
KSU-E-6	0.204	0.231	0.255	0.238	0.229	0.257	0.276
NIITableLinker-E-4(R2+BERT)	0.233	0.237	0.248	0.251	0.251	0.264	0.278
ORGE-E-2	0.219	0.225	0.238	0.240	0.235	0.250	0.264
uhai-E-5(query modification + BM25)	0.219	0.225	0.238	0.240	0.235	0.250	0.264
NIITableLinker-E-10(R3+BERT+Top100)	0.221	0.226	0.237	0.238	0.235	0.248	0.264
STIS-E-2	0.230	0.228	0.237	0.217	0.248	0.255	0.264
ORGE-E-7	0.216	0.220	0.236	0.237	0.228	0.242	0.256
ORGE-E-8	0.224	0.230	0.233	0.238	0.244	0.255	0.264
NIITableLinker-E-1	0.201	0.211	0.231	0.228	0.221	0.239	0.257
NIITableLinker-E-5(R3+BERT)	0.214	0.227	0.230	0.234	0.230	0.247	0.258
uhai-E-3(L2R)	0.209	0.214	0.227	0.237	0.223	0.234	0.249
STIS-E-10	0.208	0.209	0.221	0.208	0.234	0.242	0.253
STIS-E-1	0.201	0.201	0.221	0.199	0.227	0.234	0.249
uhai-E-1(query modification + L2R + BERT)	0.200	0.209	0.219	0.232	0.213	0.225	0.239
NIITableLinker-E-2	0.202	0.205	0.219	0.235	0.217	0.230	0.244
ORGE-E-1	0.202	0.205	0.219	0.235	0.217	0.230	0.244
NIITableLinker-E-9	0.202	0.205	0.218	0.235	0.217	0.230	0.243
uhai-E-4(L2R + BERT)	0.198	0.197	0.216	0.223	0.209	0.218	0.234
ORGE-E-4	0.192	0.201	0.213	0.226	0.207	0.224	0.238
ORGE-E-5	0.195	0.202	0.213	0.230	0.201	0.215	0.228
STIS-E-3	0.189	0.195	0.211	0.202	0.202	0.214	0.226
ORGE-E-6	0.171	0.191	0.205	0.221	0.189	0.212	0.226
uhai-E-2(query modification + L2R)	0.173	0.178	0.203	0.213	0.184	0.194	0.213
NIITableLinker-E-3	0.192	0.194	0.203	0.219	0.209	0.217	0.230
STIS-E-6	0.165	0.175	0.197	0.187	0.182	0.194	0.211
NIITableLinker-E-6	0.157	0.168	0.193	0.212	0.157	0.171	0.191
STIS-E-4	0.172	0.171	0.192	0.185	0.190	0.199	0.212
NIITableLinker-E-7	0.173	0.180	0.190	0.185	0.189	0.205	0.219
NIITableLinker-E-8	0.171	0.176	0.190	0.204	0.180	0.193	0.206
ORGE-E-3	0.144	0.154	0.180	0.192	0.151	0.169	0.190
STIS-E-5	0.155	0.151	0.177	0.171	0.175	0.181	0.198
TIS-E-7	0.167	0.163	0.172	0.164	0.186	0.192	0.201
STIS-E-8	0.151	0.153	0.171	0.165	0.174	0.182	0.195
STIS-E-9	0.104	0.118	0.151	0.149	0.115	0.130	0.149
KSU-E-4	0.062	0.059	0.052	0.051	0.065	0.066	0.068
KSU-E-8	0.050	0.043	0.039	0.025	0.060	0.061	0.063