NII Table Linker at the NTCIR-15 Data Search Task: Re-ranking with Pre-trained Contextualized Embeddings, Data Content, Entity-centric, and Cluster-based Approaches

Phuc Nguyen
National Institute of Informatics, Japan

Kazutoshi Shinoda
The University of Tokyo, Japan

Taku Sakamoto
The University of Tokyo, Japan

Diana Andreea Petrescu
The École Polytechnique Fédérale de Lausanne, Switzerland

Hung Nghiep Tran
National Institute of Informatics, Japan

Atsuhiro Takasu
National Institute of Informatics, Japan

Akiko Aizawa
National Institute of Informatics, Japan

Hideaki Takeda
National Institute of Informatics, Japan

ABSTRACT

In the Open Data era, many datasets have been made available on the Web, leading to new challenges in organizing, managing, and searching for those resources. This paper introduces NII Table Linker, a dataset searching framework designed for the two English and Japanese sub-tasks of the NTCIR-15 Data Search Task (DST). In particular, we study the capacity of the standard information retrieval techniques on DST and introduce the four re-ranking models based on (1) pre-trained contextualized embeddings, (2) entity-centric, (3) data file content, and (4) cluster-based approach. On the English sub-task, the model using pre-trained contextualized embeddings achieves the 2nd in the primary metric (nDCG@10) and the 1st in the remaining metrics in the official evaluation. On the Japanese sub-task, our Japanese runs also achieve promising performance unofficial evaluation; the run of BM25 fine-tuned produces the 1st in the nERR metrics.

TEAM NAME

NII Table Linker

SUB-TASKS

Data Search Task (English and Japanese sub-tasks)

1 INTRODUCTION

Thank the open data and research reproducible movement, a large number of data resources have been made available on the Web. The number of open datasets has risen significantly, about 560%, from 500K in 2016 to 28M in March 2020[1]. The rise of datasets and data heterogeneity also leads to many problems in organizing, managing, and searching.

According to a survey on dataset search [2], the studies of human data interaction and dataset search behavior [1, 6, 7], we are in the early stage of dataset search. Most users find dataset to solve their tasks [7], such as process-oriented and goal-oriented tasks; some of them have just explored the dataset search [6].

This preliminary work focuses on ad-hoc data retrieval, specifically NTCIR 15 Data Search Tasks (DST) [9]. In this task, users provide a text query, and the system returns a list of datasets ranked with a relevance score. In this paper, we introduce the NII Table Linker framework for the English and Japanese sub-tasks of DST. Figure 1 depicts the overall framework.

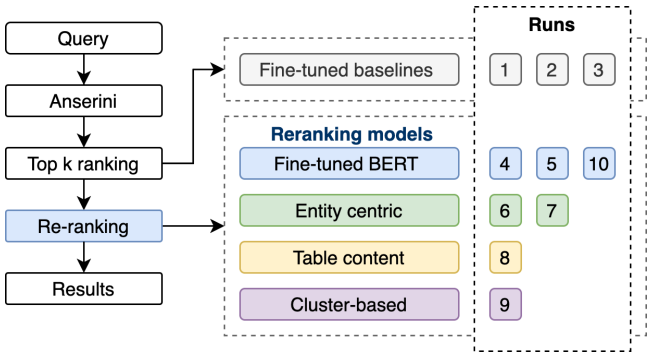

Figure 1: NII Table Linker framework

The framework consists of 10 runs for each English and Japanese sub-task. We conduct these runs with a standard procedure of ad-hoc data retrieval in run 1, 2, and 3; a sequential procedure in run 4, 5, and 10; and a parallel procedure re-ranking in run 6, 7, 8, and 9[1]. The details of each run are described as follows.

- Run 1, 2, 3: We study the capability of standard information retrieval methods on DST. Specifically, we conduct hyper-parameter fine-tuning on the shared-task baseline systems [9], e.g., BM25, BM25 with Probabilistic Relevance Framework (PRF) using Anserini toolkit [16].
- Run 4, 5, and 10: We introduce re-ranking approaches based on the pre-trained contextualized embeddings. In this study, we fine-tune the pre-trained BERT embeddings on the relevance prediction task.
- Run 6, 7: In these English runs, we extract special information from dataset descriptions such as name entities, noun

[1]The run 7, and 8 are difference between English and Japanese sub-task

phrases (E6), time, location (E7). That extracted information is used to create new indices for another searching branch. The re-rank models are combined by the baseline searching branch and an entity-centric branch. Due to the difference of language characteristic, we only prepare a run 6 for Japanese data (J6) using noun phrase information.

- Run 8: In the English sub-task (run E8), we focus on extracting table headers of tabular resources. We also use the parallel procedure in which incorporating the results of the baseline search and the data content search. In the Japanese sub-task, we build J7 and J8 with a different procedure with ad-hoc weighting. We use table header information in run J7 while not using this information in J8.
- Run 9: In this run, we perform dataset clustering to address the duplicated dataset problem in the corpus [1]. We also create new indices with a searching unit as cluster content. Then, the re-rank model is also based on the baseline search results and the cluster-based search results.

The remaining of this paper is structured as follows. Section 2 provides definitions on datasets, tasks, and queries of the dataset search. In Section 3, we analyze datasets and query characteristics in English and Japanese resources. This analysis uses the datasets and training, testing queries provided by the shared-task organizers. Section 4 describes the overall framework of NII Table Linker, and the detailed implementation of each run. Section 5 reports NII Table Linker results and discusses possible directions to improve the current dataset search. Finally, we conclude in Section 7 with the lesson learned from the shared-task and discuss future work.

2 DEFINITION

2.1 Dataset

A dataset is a collection of data files such as tabular data, structured data, images, videos, or machine learning models [2]. The search engines use metadata to describe semantic dataset annotations due to the large variety of data formats.

In the DST context, a dataset is a pair of metadata and data files taken from the US (English sub-task)[3] and Japanese (Japanese sub-task) governmental data portal[4]. There is one data file for each dataset in Japanese sub-tasks, while many data files might be available in the English dataset. In English datasets, the three most popular formats are documents e.g. PDF (50.88%), structured e.g. XML (35.05%), and tables e.g. CSV (4.27%). In Japanese datasets, the three most popular formats are tables, e.g., Excel (53.89%), CSV (42.44%), and documents, e.g., PDF (3.67%).

2.2 Dataset Search

According to the shared-task overview report [9], the relevance judgments were conducted on a sub-sampling (the pooled top ranking of the baseline system) of the total dataset in the portals. The remaining non-judged dataset was ignored from the relevance judgments. Therefore, we formalize the NTCIR 15 Data Search Task as a re-ranking problem. The re-ranking models are constructed on the top of the baseline systems in a sequential procedure (Section 4.2) or parallel procedure (Section 4.3.1).

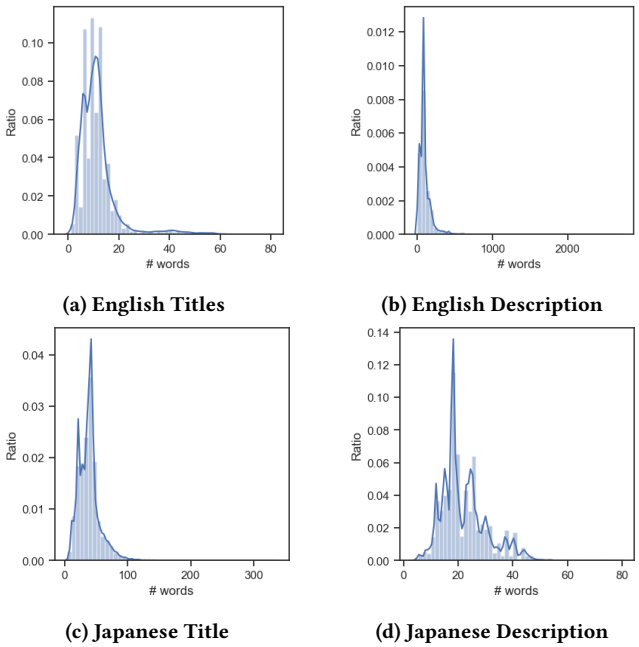

Figure 2: Word distribution of title and description in English and Japanese metadata resources

2.3 Queries

In general cases, a dataset search query could be simple as a form of natural language text, keywords, or a combination with other criteria about publishers, geographical, temporal information. Moreover, data search queries could be a data file such as a table, figure, or data model, or a combination of multiple data files. In DST, the query is a text manually created by crowd-sourcing services. The brief analysis of data search queries is reported in Section 3.2

3 PROFILING NTCIR 15 DATA SEARCH TASK

This section describes our analysis of the English and Japanese corpus and user query characteristics.

3.1 English and Japanese Dataset

The English resources have 46,615 datasets, while there are 1,338,492 datasets in Japanese resources. The English resources only have 68.32% provider information and 62.99% category information, while the Japanese resources cover 100% of providers (担当機関) and category information (政府統計名). There are 17% of English datasets have tags, while there are no tags in Japanese datasets. The two resources have 100% cover of title, description, and at least one data file for each dataset.

Figure 2 depicts the distribution number of words in the title and description of English and Japanese datasets. The majority of English datasets have a title of fewer than 20 words and a description of fewer than 500 words. In Japanese, most datasets have titles less than

[2]Google Dataset: https://developers.google.com/search/docs/data-types/dataset
[3]US Government's open data: https://www.data.gov/
[4]Japanese Government Statistics: https://www.e-stat.go.jp

100 words, while the length of the description is less than 50 words. One interesting observation is that the description (mean=98.01 words) is longer than the title (mean=11.54 words) in an English dataset, while in Japanese data, the description (mean=22.07 words) is shorter than the title (mean=38.44 words) [5].

3.2 Query characteristics

In this session, we analyze the training and testing queries of English and Japanese sub-tasks.

Table 1: Statistics on query length

Sub-task	all	min	max	mean	median
English	192	2	11	4.74	4.50
Japanese	192	2	15	4.49	4.00

Table 1 shows the statistics of the query length of DST. Overall, there are 192 queries (96 training queries and 96 testing queries) for each sub-task. The English and Japanese statistics do not have much difference between the query length of English (mean = 4.74 words) and Japanese (mean = 4.49 words) sub-tasks. There is 10% of English queries and 15% Japanese queries that contain numerical text (including temporal information and numbers).

4 APPROACH

In this session, we describe our approaches to DST with the three different procedures as follows.

- In the first procedure (Section 4.1), we follow the standard retrieval pipe-line and conduct a fine-tuning on the standard retrieval methods, e.g., BM25 and BM25 with Probabilistic Relevance Framework (PRF).
- In the second procedure, we adopt a sequential one to build a re-ranking model with pre-trained contextualized embeddings on top of the standard approaches (Section 4.2).
- In Section 4.3.1, we use a parallel procedure for the entity-centric, table content, and cluster-based approaches. The indexing and searching of the standard models and these re-ranking models are processed separately and aggregated at top k weighted fusion.

4.1 Fine-tuned with standard methods

We build run 1, 2, and 3 based on the standard retrieval models with the Anserini toolkit [16]. Regarding the fine-tune setting, we use 5-fold cross-validation to find the optimized hyper-parameters. To evaluate the fine-tuned model performance, we use the combination metric as the sum of nDCG@20, nERR@20, and Q measure.

In this setting, we only perform fine-tuning on the two algorithms: BM25 and BM25+PRF, since the results of these algorithms give the best performances in the training data with the Anserini default hyper-parameters.

The fine-tuned results of each run are described as follows.

- Run 1: Fine-tuning the BM25 algorithm. In English sub-tasks, the optimized hyper-parameters are k1 = 0.83 and b = 0.28. In Japanese sub-task, the optimized hyper-parameters are k1 = 0.9 and b = 0.5.
- Run 2: Adopting the BM25+PRF algorithm with the Anserini default hyper-parameters. These are applied for the two sub-tasks of English, and Japanese as k1 = 0.9, b = 0.4, k1(PRF) = 0.9, b(PRF) = 0.4, number of new terms = 20, number of documents = 10, new term weight = 0.2
- Run 3: Fine-tuning the BM25+PRF algorithm. The optimized hyper-parameter for English are k1 = 0.83, b = 0.28, k1(PRF) = 1.5, b(PRF) = 0.35, new term weight = 0.05, number of new terms = 25, number documents = 10. In the Japanese sub-task, the optimized hyper-parameter are k1 = 0.9, b = 0.25, k1(PRF) = 1.5, b(PRF) = 0.45, new term weight = 0.1, number of new terms = 10, number documents = 10.

4.2 Re-ranking with a sequential procedure: pre-trained contextualized embeddings

Inspired by a sequential re-ranking procedure with the BERT models [13], we adopt this procedure for the re-ranking with the pre-trained BERT models [3]. We use the BERT models as preliminary testing; however, we can also use other difference pre-trained models such as fasttext [11], RoBERTa [8].

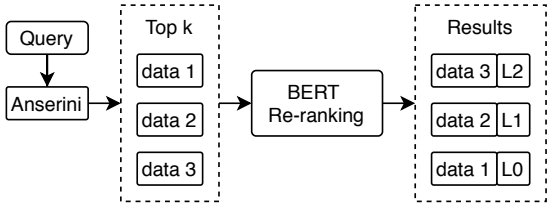

Figure 3: Re-ranking model with a sequential procedure

Figure 3 depicts the re-ranking model using the pre-trained contextualized embeddings. We use the pre-trained BERT base uncased model for the English sub-task [3]. For the Japanese sub-task, we use the Tohoku university pre-trained BERT base model trained on Japanese Wikipedia [14].

We fine-tune the pre-trained BERT models with the task of classification. Given a query and top k dataset ranking from the baseline systems, we create a list of samples as the pairs of [query, dataset information]. The predicting target is a relevance level of L0, L1, or L2. L2 is relevant, L0 is an irrelevance, and L1 is the relevance level between relevant and irrelevance. We re-ranking the top k dataset in terms of the predicted relevance levels to get the final answer. For example, in Figure 3, data 3 is ranked 3rd in the first result of the Anserini toolkit, then after the BERT re-ranking, this dataset is ranked 1st with a prediction of L2 (relevance).

Figure 4 depicts the fine-tuned BERT architecture. We stack a simple feed-forward neural network on the top of the BERT output. The BERT output dimension is 768, the hidden layer dimension is 32, and the output dimension is 3 (L0, L1, L2).

In this re-ranking models, we split 80% training data for fine-tuning and 20% for validation. The model performance is calculated on the validation set after finishing one epoch and stop the fine-tune when there is performance degradation. The detailed setting of each run is described as follows.

[5]To count Japanese words, we use Mecab-ipadic as the tokenizer

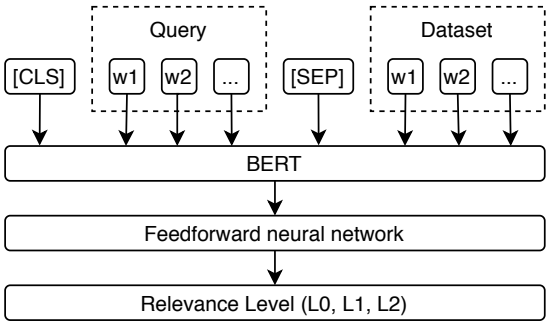

Figure 4: BERT fine-tune model for the relevance prediction

- Run 4: BM25+PRF with the Anserini default hyper-parametes + BERT (top 20). We use the top 20 ranking results from Run 2 for fine-tuning.
- Run 5: BM25+PRF with hyper-parameters fine-tuning + BERT (top 20). This run has a similar setting with Run 4, but we choose the top 20 ranking results from Run 3.
- Run 10: BM25+PRF with hyper-parameters fine-tuning + BERT (top 100). This run is similar to Run 5 but we select top 100 ranking results for fine-tuning.

We use the same hyper-parameters for run 4, run 5, and run 10 as learning rate = 5e-6, eps = 1e-8, epochs = 4, batch size = 32, max length token = 256.

4.3 Re-ranking with a parallel procedure

This section describes the parallel re-ranking procedure with entity-centric, data file content, and cluster-based approaches. Different from the sequential one, this procedure allows a weighted fusion on the final results. As a result, we can adjust how much contributions of other information from different re-ranking models.

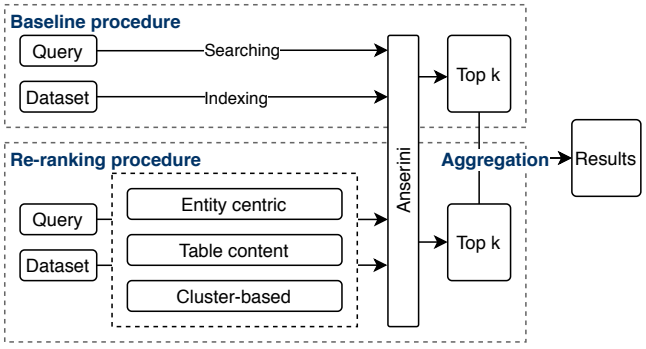

Figure 5: Re-ranking model with a parallel procedure

Figure 5 depicts the detailed procedure of these re-ranking models. In this procedure, we keep the top k ranking of the baseline system and then incorporate the other re-ranking models' top k ranking. We use the top k ranking of the BM25+PRF results (run 2) as the baseline procedure in the parallel procedure runs because this run gives the highest performance in the training data. To be simple, we take the average score of each item in the two top k ranking and sort the final ranking list in terms of the average scores. In general cases, we can train a weighted fusion function on the training data to control how much the re-ranking models contribute to the final result.

4.3.1 *Re-ranking with entity-centric approaches.*

English sub-task. Regarding the English sub-task, we use the pre-trained model on (OntoNote 5) of the SpaCy toolkit to extract entities from queries and dataset information [5].

Table 2: Percentage of entities, noun phrases, locations, and time information of English datasets

		Entities	Noun phrases	Location, Time
English	Datasets	99%	100%	82%
	Queries	44%	99%	17%

Table 2 reports the percentage of entities, noun phrases, and location, times in dataset content, and queries. Most datasets contain entity information with 99% datasets. The dataset titles and descriptions also contain rich information about location and time with 82% datasets. Although we have high-quality entity-described datasets, the availability of such information is relatively low in queries. 44% of queries contain entities, and 17% of them have locations and time information.

In the English run of E7, we extract entities (name entity recognition) and noun phrases with the SpaCy toolkit from data-title and description, then use this information to create new indices with the Anserini toolkit. In the E8 run, we only focus on extracting the types of location and time from queries, and datasets, e.g., LOC, TIME, and DATE of SpaCy entity types.

Japanese sub-task. Regrading entity-centric for the Japanese sub-task, due to the low performance of the entity extracting modules on the training data, we only focus on the noun-phrases of datasets. For run J6 of the Japanese dataset, we extracted noun phrases from metadata using a conventional Japanese dependency parser [6] with a simple POS-tag based filtering in the extraction. In the end, we only submitted the J6 run with noun phrases re-ranking model for the Japanese sub-task.

4.3.2 *Data file content.*

English sub-task. We first extract the headers in tabular resources since these headers carry important information about the schema of table data. Overall, we only extract 4.27% headers from tabular data. In the run E8, we concatenate all metadata content and the extracted table headers as a dataset content. We use this information to create new indices with Anserini for further searching.

Japanese sub-task. We build the two runs of 7 and 8 for the Japanese sub-task to study, whereas the table content does help in DST. The run of J7 and J8 compare the cases with and without the table content for the Japanese dataset. For the table content, we first extracted Japanese text in the top five lines of each CSV file (`EXCEL` files were converted into `CSV` while all `PDF` files were discarded) and then, identify all noun phrases in the same manner

[6] https://taku910.github.io/cabocha/

as J6. Considering the result's interpretability, we used word2vec representations only to identify the most similar index term for each query term and used a simple TF-IDF term weighting scheme for similarity calculation. Unlike other runs, the Anserini ranking was not used in these specific runs.

The performance values of runs J7 and J8 are moderate, and the comparison shows that exploiting table contents requires further consideration.

4.3.3 Re-ranking with cluster-based approach. According to the Google dataset search report, we have many duplicated datasets in the corpus [1]. We assume that duplicated datasets might have similar data content, such as title or description. So that, grouping these datasets could alleviate duplicated datasets, as a result, improve the dataset search performance.

In the English sub-task, we perform dataset clustering on the corpus with k-means[7] on the TF-IDF vectorizer of data-title and description on the English sub-task. We use the cluster results as samples; the sample content is the concatenations of all datasets' content in a cluster.

In the Japanese sub-task, we perform k-means clustering using latent topic distributions as features assigned to each entry learned by the neural topic model [10]. In the Japanese dataset, we observe that some words appear in multiple domains. We hypothesize that this nature of the dataset makes count-based clustering difficult. The neural topic model is expected to effectively obtain meaningful distributed representations of each entry in an unsupervised way because of its flexible and high modeling capacity. As our analysis of training data, we select the cluster results of k-means (k = 100) to index with the Anserini toolkit.

We use Anserini to index all clusters. The query will be searched on the cluster indices in the searching step and return the most relevant clusters. We then use the information of clusters, datasets in these clusters to generate the cluster dataset ranking. The final ranking results are derived from the aggregation of the baseline ranking and the cluster dataset ranking.

5 RESULTS

Table 3 reports the overall results of NII Table Linker by nDCG, nERR, and Q-measure metrics on English and Japanese sub-tasks. Due to an issue in the submission procedure, our Japanese results were not selected in the official evaluation pool. The reported performances of our Japanese runs were evaluated on the unofficial evaluation. So that, directly comparing our Japanese runs could downgrade the performance of these runs.

In English sub-task results, the fine tunes of BM25 (E1), BM25+PRF (E2) and the fine-tuning of BM25+PRF (E3) do not help to improve searching performance. The run of ORGE-E-2 using the BM25 with Anserini default hyper-parameters gave 0.248 in the primary metric (nDCG@10) [9], while -7% performance degeneration of our fine-tune of BM25. The default BM25+PRF (E2) gives a better performance than the fine-tuned one (E3). This observation gives us an assumption that the BM25 algorithm with Anserini default hyper-parameters was selected as the primary baseline system. Since only the top ranking of the baseline system was selected for the relevance judgments, a small variance of the retrieval algorithm might affect the overall performance.

The sequential procedure with the BERT model (E4) achieves the best performance of our submissions. In this run, we build a BERT fine-tune on the top of BM25+PRF (default Anserini hyper-parameters), but we can get better performance by a fine tune on top of the BM25 algorithm with the above assumption. Our E4 result achieves the 2nd in the primary metric (nDCG@10) and the 1st in the remaining metrics in the official evaluation.

The high performance could be explained as our fine-tuned BERT model could learn user intentions from the relevance judgments in training data. So, applying this model to the testing data also yields robust and effective results. The BERT re-ranking boost 20% improvements in terms of the average metrics (nDCG, nERR, and Q-measure) in the E4 runs compared to the run E2 without using this re-ranking method.

The other re-ranking models of the parallel procedure do not give a good performance with DST evaluation metrics. The performance in the primary metric (nDCG@) decreased by about 9% with the re-ranking of entity-centric, 9% with the re-ranking of table content, and 2% with the cluster-based approach compared to a run 2 using the BM25+PRF with the Anserini default hyper-parameters. Although these re-ranking models do not improve the DST search performance, this information also useful when evaluated on different searching purposes (focus on entity-centric, table content, and location time information).

In Japanese sub-task results, the fine-tune of BM25 (J1) and a re-ranking with the BERT model (J10) gave the best performances in our Japanese submissions. Although our submissions were not selected in the evaluation pool of the official evaluation, the submission of J1 also achieves the best performance in the nERR metrics.

6 EXTRA ANALYSIS

In this section, we present some additional experiments and analyses regarding the embedding search methods based on our semantic query approach [15]. The extra evaluations were conducted by the organizer separately from the official evaluations at a later date.

The additional run E-EX-2 is a re-ranking of E2 using FastText embedding search. Specifically, for the FastText embedding search method in E-EX-2, we computed each query's embedding and each dataset by TF-IDF weighted averaging of FastText word embeddings [4], then got the top similar datasets for each query as measured by the cosine similarity between their embedding vectors. Note that the IDF statistics were computed using the set of all datasets as the corpus. The TF statistics were computed for each query and each dataset separately.

The additional run E-EX-6 is also a re-ranking of E2 but with a different embedding search method. Specifically, we directly computed the maximum pairwise matching from each word of each query to every word of each dataset as measured by the cosine similarity between word embedding vectors, and vice versa, the maximum pairwise matching from each word of each dataset to every word of each query, then got the top similar datasets for each query. Note that we also weighed the contribution of each word in the similarity by its TF-IDF statistic.

[7] We test k in [10, 100, 1000], and obtained the best performance when k=10

Table 3: Results of NII Table Linker by nDCG (@3, @5, @10), nERR (@3, @5, @10), and Q-measure on English and Japanese sub-tasks. The "E" prefix runs are our runs for English sub-task, while "J" prefix runs are Japanese sub-task. The best score of each task is in bold.

Runs	Name	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-measure
E1	BM25 [fine-tune]	0.201	0.211	0.231	0.228	0.221	0.239	0.257
E2	BM25+PRF	0.202	0.205	0.219	0.235	0.217	0.23	0.244
E3	BM25+PRF [fine-tune]	0.192	0.194	0.203	0.219	0.209	0.217	0.23
E4	E2+BERT (Top 20)	**0.233**	**0.237**	**0.248**	**0.251**	**0.251**	**0.264**	**0.278**
E5	E3+BERT (Top 20)	0.214	0.227	0.23	0.234	0.23	0.247	0.258
E6	Entity-Centric	0.157	0.168	0.193	0.212	0.157	0.171	0.191
E7	Date-Location	0.173	0.18	0.19	0.185	0.189	0.205	0.219
E8	Table-content	0.171	0.176	0.19	0.204	0.18	0.193	0.206
E9	Cluster-based	0.202	0.205	0.218	0.235	0.217	0.23	0.243
E10	R3+BERT (Top 100)	0.221	0.226	0.237	0.238	0.235	0.248	0.264
J1*	BM25 [fine-tune]	**0.400**	0.405	0.415	**0.446**	0.467	**0.483**	**0.478**
J2*	BM25+PRF	0.382	0.396	0.405	0.426	0.452	0.464	0.445
J3*	BM25+PRF [fine-tune]	0.396	0.407	0.412	0.445	0.468	0.480	0.463
J4*	J2+BERT (Top 20)	0.378	0.393	0.404	0.424	0.450	0.463	0.445
J5*	J3+BERT (Top 20)	0.389	0.403	0.411	0.438	0.463	0.475	0.463
J6*	Noun Phrase	0.313	0.321	0.335	0.345	0.364	0.381	0.311
J7*	NP+tfidf+w2v+adhoc weighting	0.335	0.342	0.363	0.368	0.386	0.402	0.368
J8*	NP+tfidf+w2v+tablehead+adhoc weighting	0.346	0.354	0.366	0.387	0.406	0.422	0.367
J9*	Cluster-based	0.376	0.385	0.400	0.433	0.458	0.473	0.444
J10*	R3+BERT (Top 100)	0.395	**0.409**	**0.416**	0.445	**0.471**	0.482	0.464

* Due to the duplicated submission issue, our Japanese runs were not selected in the official evaluation pool, so that these runs are non-comparable with the other participants run the overview paper [9]

The extra evaluation results are shown in Tab. 4. First, we notice that re-ranking strongly improves the results of a full-text search (E2 run) in all metrics, which is in agreement with our observation in Sec. 5. Second, the run E-EX-2 gets better results than other methods in most metrics, suggesting that re-ranking using embedding search is a promising approach. Finally, we see that the run E-EX-2 gets better results than E4 in most metrics but worse in nDCG@10, which suggests that each embedding search method may have different strengths and drawbacks; thus, there is some room for improvement.

We also note that the embedding search method in E-EX-2 is efficient because FastText sentence embedding is fast and can be computed for arbitrarily long sentences. Moreover, it can still be efficient when using other expensive sentence embedding methods because the embeddings of datasets are only computed once and used for any query.

7 CONCLUSIONS

This paper introduces NII Table Linker for the English and Japanese sub-tasks of the NTCIR 15 Data Search. In NII Table Linker, we study the capability of standard retrieval methods and introduce two re-ranking procedure: sequential with BERT fine-tuning and parallel with entity-centric, table content, and cluster-base approaches. Additionally, we also introduce embedding search (in the extra analysis sesion) for DST.

Overall, the fine-tuned methods on standard retrieval algorithms do not give improvements to searching performance. The re-ranking sequential procedure with the BERT models gives the best performance in all our submissions. Additionally, it also achieves the 2nd in the primary metric (nDCG@10) and the 1st in the remaining metrics in the official evaluation. Re-ranking in parallel procedures with entity-centric, table content, and cluster-based does not improve the overall search results. However, these runs could give better results when evaluated on different searching objectives such as entity search and table data retrieval. The re-ranking with embedding search also provides a promising performance in the extra evaluation setting.

In future work, we will study automatic techniques to improve and standardize datasets. This paper focuses on studying the capability of standard information retrieval and re-ranking methods on

Table 4: Extra evaluation results of NII Table Linker by nDCG (@3, @5, @10), nERR (@3, @5, @10), and Q-measure on the English sub-task. These results are measured by the organizer separately from the official evaluation results. The runs E-EX-1 and E-EX- are the two extra submissions using embedding search. The best score is in bold.

Runs	Name	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-measure
E1	BM25 [fine-tune]	0.197	0.206	0.223	0.219	0.236	0.253	0.217
E2	BM25+PRF	0.198	0.200	0.212	0.215	0.228	0.240	0.224
E3	BM25+PRF [fine-tune]	0.188	0.187	0.195	0.207	0.213	0.227	0.209
E4	E2+BERT (Top 20)	0.227	0.229	**0.239**	0.246	0.259	0.274	0.238
E5	E3+BERT (Top 20)	0.213	0.221	0.223	0.229	0.244	0.256	0.225
E6	Entity-Centric	0.148	0.16	0.180	0.151	0.164	0.184	0.195
E7	Date and Location	0.169	0.172	0.176	0.184	0.200	0.212	0.169
E8	Table Content	0.169	0.173	0.185	0.179	0.191	0.204	0.193
E9	Cluster-based	0.198	0.200	0.211	0.215	0.228	0.240	0.224
E10	R3+BERT (Top 100)	0.219	0.222	0.228	0.234	0.246	0.260	0.226
E-EX-2	E2+EmbSearch1	**0.227**	**0.236**	0.230	**0.255**	**0.271**	**0.277**	**0.239**
E-EX-6	E2+EmbSearch2	0.220	0.216	0.222	0.243	0.250	0.263	0.232

DST. It could help users find a corresponding dataset based on the available dataset metadata, description. However, many datasets do not have metadata, or the metadata is not completed or out of update [1]. Improve, and standardize metadata are a must in future data search directions.

Another direction is on delivering the question answering experiences to data search users. Our vision on NII Table Linker is a question answering system where it could answer the questions related to knowledge inside data files [8]. We first plan to match data files to knowledge graphs and further utilize and standardize the knowledge inside data files [12]. As a result, the data search queries could be answered with knowledge graph inferences.

ACKNOWLEDGMENTS

This work was supported by the Cross-ministerial Strategic Innovation Promotion Program (SIP) Second Phase, "Big-data and AI- enabled Cyberspace Technologies" by the New Energy and Industrial Technology Development Organization (NEDO).

We would like to thank Xanh Ho; a student of the Graduate University for Advanced Studies SOKENDAI, for making a preliminary evaluating dataset (about 100 queries on 10 datasets) on the English sub-task.

REFERENCES

[1] Omar Benjelloun, Shiyu Chen, and Natasha Noy. 2020. Google Dataset Search by the Numbers. *arXiv preprint arXiv:2006.06894* (2020).

[2] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *The VLDB Journal* 29, 1 (2020), 251–272.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8]Note that, most metadata currently was added by publishers which possibly could not cover all data file knowledge. [2]

[4] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[5] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[6] Emilia Kacprzak, Laura M Koesten, Luis-Daniel Ibáñez, Elena Simperl, and Jeni Tennison. 2017. A query log analysis of dataset search. In *International Conference on Web Engineering*. Springer, 429–436.

[7] Laura M Koesten, Emilia Kacprzak, Jenifer FA Tennison, and Elena Simperl. 2017. The Trials and Tribulations of Working with Structured Data: -a Study on Information Seeking Behaviour. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1277–1289.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

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

[10] Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering Discrete Latent Topics with Neural Variational Inference. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, 2410–2419.

[11] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[12] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. 2019. MTab: matching tabular data to knowledge graph using probability models. *arXiv preprint arXiv:1910.00246* (2019).

[13] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[14] Masatoshi Suzuki. 2019. Pretrained Japanese BERT models. https://github.com/cl-tohoku/bert-japanese.

[15] Hung Nghiep Tran and Atsuhiro Takasu. 2019. Exploring Scholarly Data by Semantic Query on Knowledge Graph Embedding Space. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 154–162.

[16] 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.