# KSU Systems at the NTCIR-15 Data Search Task

Taku Okamoto
Kyoto Sangyo University
i2086042@cc.kyoto-su.ac.jp

Hisashi Miyamori
Kyoto Sangyo University
miya@cc.kyoto-su.ac.jp

## ABSTRACT

In this paper, we describe the system and results of Team KSU for Data Search Task in NTCIR-15. The documents covered by this task consist of metadata extracted from the governmental statistical data and the body of the corresponding statistical data. The metadata is characterized by the fact that its document length is short, and the main body of statistical data is almost always composed of numbers, except for titles, headers, and comments. We newly developed the categorical search that narrows the set of documents to be retrieved by category in order to properly capture the scope of the problem domain intended by the givne query. In addition, to compensate for the short document length of metadata, we implemented a method of extracting the header information of the table from the main body of statistical data to augment documents to be searched. As a ranking model, we adopted BM25, which can be adjusted with few parameters to take into account term frequency and document length. We also decided to apply a BERT-based reranking method to understand how much contribution to this task can be expected from the pretrained language model. The evaluation in the official run showed that the combined method of category search and BM25 scored 0.448 in nDCG@10 in the Japanese subtask and 0.255 in nDCG@10 in the English subtask, where each showed the highest score on this measure among all the official runs.

## KEYWORDS

Text classification, categories, table understanding, header extraction, data search

## TEAM NAME

KSU

## SUBTASKS

Japanese subtask, English subtask

## 1 INTRODUCTION

Data retrieval refers to the search of publicly available data such as the open data provided by governments and other organizations, and is different from the conventional search of text documents written in natural language. Open data is one of the essential resources for people around the world to collaborate on global challenges, and data retrieval has attracted the attention of many researchers in recent years. NTCIR-15 Data Retrieval Task aims at establishing the fundamental technologies for the ad-hoc retrieval of governmental statistical data[2], and set up two subtasks targeting for ad-hoc retrievals of Japanese governmental statistical data (e-Stat) and for those of U.S. governmental statistical data (Data.gov).

In this paper, we describe the system and results of Team KSU for Data Search Task in NTCIR-15.

The documents covered by this task consist of metadata extracted from the governmental statistical data and the body of the corresponding statistical data. The metadata is characterized by the fact that its document length is short, and the main body of statistical data is almost always composed of numbers, except for titles, headers, and comments.

We newly developed the categorical search that narrows the set of documents to be retrieved by category in order to properly capture the scope of the problem domain intended by the givne query. In addition, to compensate for the short document length of metadata, we implemented a method of extracting the header information of the table from the main body of statistical data to augment documents to be searched. As a ranking model, we adopted BM25, which can be adjusted with few parameters to consider term frequency and document length. We also decided to apply a BERT-based reranking method to understand how much contribution to this task can be expected from the pretrained language model.

The evaluation in the official run showed that the combined method of category search and BM25 scored 0.448 in nDCG@10 in the Japanese subtask and 0.255 in nDCG@10 in the English subtask, where each showed the highest score on this measure among all the official runs.

## 2 DATA COLLECTION AND QUERY ANALYSIS

In order to clarify the guidelines for the construction of the system to be implemented, we first analyzed the properties of the data collection, which is the set of documents to be searched in this task. The documents to be searched in this task are composed of metadata and one or more corresponding statistical data in the data collection. Metadata is extracted from the introduction page of statistical data of e-Stat and Data.gov, respectively, and statistical data is the main body of the statistical data in a table format linked with metadata. Metadata is provided in JSON format, and contains variables such as id, url, title and description describing a brief summary of the statistical data, as well as URL, file format, and file name of the main body of statistical data.

In this paper, we focus on title and description variables in the metadata, because we believe that natural language descriptions such as summaries are more important than numerical values and URLs as search clues.

Table 1 shows the average and standard deviation of document lengths for metadata title and description, as well as the number of documents in the metadata.

**Table 1: Statistics on metadata in the data collection**

| subtask | document length | | number of documents |
|---|---|---|---|
| | average | stddev | |
| English | 101.93 | 81.19 | 46,615 |
| Japanese | 11.83 | 3.02 | 1,338,402 |

From Table 1, it can be seen that the document length of title and description is about 100 words in English and 10 words in Japanese. On the other hand, the document length of a newspaper article, which is often used in conventional ad hoc searches, is about 400 words in English[1] and 330 words in Japanese[4] . Therefore, the document length of metadata-only documents in this task, taken as a natural language description, is shorter than that of conventional ad hoc search.

Meanwhile, the main body of the statistical data is provided in various file formats, as shown in Table 2. The contents of the statistical data are expressed in table format, as shown in Figure 1, and consist of table titles, row headers, column headers, and table bodies. The main body of the table is basically described as numerical data, and only the table titles, row headers, and column headers are described in natural language.

**Table 2: Distribution of file format of statistical data**

(a) Japanese subtask (e-Stat)

| file format | frequency |
|---|---:|
| xls | 686436 |
| csv | 568042 |
| pdf | 49124 |
| xlsx | 34794 |
| xlsm | 6 |

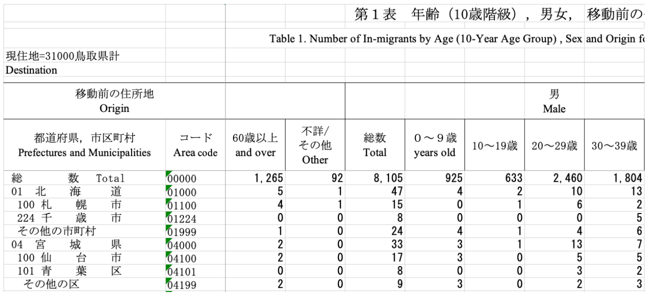
(b) English subtask (Data.gov)

| file format | frequency |
|---|---:|
| pdf | 47260 |
| x_gzip | 17099 |
| html | 9296 |
| xml | 4443 |
| csv | 2919 |
| plain | 2761 |
| none | 2542 |
| json | 1938 |
| rdf+xml | 1484 |
| octet-stream | 1430 |
| ms-excel | 753 |
| sheet | 568 |
| zip | 98 |
| ms-word | 88 |
| pgp-signature | 55 |
| document | 54 |
| x-octet-stream | 37 |
| x-zip-compressed | 15 |
| rss+xml | 11 |
| x-zip | 11 |
| x-sh | 7 |
| excel | 1 |
| pcap | 1 |
| javascript | 1 |
| jpeg | 1 |

Therefore, statistical data, with the exception of some items such as titles and headers, are basically described as numerical data, which makes it difficult to relate them directly to queries given as a set of words.

In this paper, we focus on the titles and descriptions in the metadata in the statistical data collections, and the titles, row headers and column headers in the statistical data, and use them as search clues.

Next, we considered the procedure for building the queries provided in this task.



第１表　年齢（10歳階級），男女，移動前の

Table 1. Number of In-migrants by Age (10-Year Age Group) , Sex and Origin fo

現住地=31000鳥取県計
Destination

| 移動前の住所地<br>Origin | | | | | | | 男<br>Male | |
|---|---|---|---|---|---|---|---|---|
| 都道府県，市区町村<br>Prefectures and Municipalities | コード<br>Area code | 60歳以上<br>and over | 不詳/<br>その他<br>Other | 総数<br>Total | ０～９歳<br>years old | 10～19歳 | 20～29歳 | 30～39歳 |
| 総　数　Total | 00000 | 1, 265 | 92 | 8, 105 | 925 | 633 | 2, 460 | 1, 804 |
| 01　北　海　道 | 01000 | 5 | 1 | 47 | 4 | 2 | 10 | 13 |
| 100 札　幌　市 | 01100 | 4 | 1 | 15 | 0 | 1 | 6 | 2 |
| 224 千　歳　市 | 01224 | 0 | 0 | 8 | 0 | 0 | 0 | 5 |
| その他の市町村 | 01999 | 1 | 0 | 24 | 4 | 1 | 4 | 6 |
| 04　宮　城　県 | 04000 | 2 | 0 | 33 | 3 | 1 | 13 | 7 |
| 100 仙　台　市 | 04100 | 2 | 0 | 17 | 3 | 0 | 5 | 5 |
| 101 青　葉　区 | 04101 | 0 | 0 | 8 | 0 | 0 | 3 | 2 |
| その他の区 | 04199 | 2 | 0 | 9 | 3 | 0 | 2 | 3 |

**Figure 1: Example of a typical table structure**

When constructing the queries, we first use the Japanese community QA service, Yahoo!Chiebukuro and get a question-and-answer pair that includes a link to the e-Stat, and then extract the questions indicating the information need for the data search to be addressed. The questions extracted here are called topics. From these topics, the queries are obtained manually through crowdsourcing. For English language, the topic is obtained by translating a Japanese topic into English, and the query is similarly acquired by crowdsourcing. Task participants are given the data collection and the queries and no topics are provided.

In this task, since the document length of the document to be retrieved is short and the topics are unknown, it is necessary to devise a way to better relate the query to the relevant documents in some way.

In this paper, we focused on the fact that all topics were obtained from the QA data of Yahoo Chiebukuro, including links to e-Stat, and thought it would be possible to narrow down the documents to be searched efficiently by using the categories used in Yahoo! Chiebukuro. Yahoo Chiebukuro uses ten categories as shown in Table 3(a), whereas Yahoo! Answers, an English community QA service, uses 23 categories as shown in Table 3(b). In this paper, we assigned these categories to all documents in the data collection in advance. By estimating the category for the given query, we considered that the association between the query and the relevant documents would be improved by narrowing down the set of documents to be retrieved to the set of documents corresponding to that category.

## 3 PROBLEM FORMULATION

In this section, we formulate the problems to be addressed in this task.

First, let the query set $Q$ and the document set to be retrieved $D$ be represented as:

$$Q = \{q_i\}, \quad D = \{d_j\} \tag{1}$$

In this task, the document to be retrieved $d_j$ consists of the metadata $m_j$ and the statistical data $t_j$.

$$d_j = m_j \cup t_j \tag{2}$$

For a given query $q_i$, the objective is to perform ranking using the appropriate function *ranking* from the set of documents $D$ to obtain the ranking result $R_i$. Here, $R_i$ is expressed as follows.

$$R_i = \{r_{i,k}\} = ranking(q_i, D) \tag{3}$$

**Table 3: Categories used in category search and the number of QA pairs collected**

(a) Obtained from Yahoo! Chiebukuro

| category | # of QA |
|---|---|
| internet | 145 |
| entertainment | 156 |
| technology | 140 |
| device | 143 |
| manners | 151 |
| health | 151 |
| parenting | 149 |
| relationships | 150 |
| chiebukuro | 146 |
| life | 163 |

(b) Obtained from Yahoo! Answers

| category | # of QA |
|---|---|
| art | 260 |
| business | 260 |
| cars | 260 |
| computer | 20 |
| dining | 89 |
| education | 58 |
| entertainment | 220 |
| family | 260 |
| food | 20 |
| games | 260 |
| health | 260 |
| home | 54 |
| local | 180 |
| news | 240 |
| pets | 240 |
| politics | 160 |
| pregnancy | 20 |
| science | 260 |
| social | 140 |
| society | 60 |
| sports | 260 |
| style | 40 |
| travel | 20 |

## 4 CATEGORY SEARCH

We developed a method of category search which narrows the set of documents to be retrieved into categories in order to capture the range of the problem domain intended by the user query properly. At the time of indexing, each document is assigned a category by using a pre-built text classifier, and a new set of documents with the categories is registered. At the time of searching, the category is estimated from the given query using the text classifier, and the result is ranked only on the set of documents belonging to the estimated category. The procedure of category search is shown in Figure 2.

First, let the category set $C$ be expressed as follows:

$$C = \{c_p\} \tag{4}$$

At the time of indexing, a category $c_p$ is assigned to each document $d_j$ as a label $l_j$ by the $text\_classifier$.

$$(d_j, l_j), \quad l_j = c_p = text\_classifier(d_j) \in C \tag{5}$$

The labeled set of documents to be retrieved $D'$ is indexed and registered. $D'$ is described as the following:

$$D' = \{(d_j, l_j)\} \tag{6}$$

When searching, the text classifier $text\_classifier$ is used to estimate the category $c_p$ for the given query $q_i$.

$$c_p = text\_classifier(q_i) \in C \tag{7}$$

The function $ranking$ executes ranking only for the document set $D_{c_p}$ belonging to the category $c_p$, and returns the search result $R_i$.

$$D_{c_p} = \{d_j | (d_j, l_j) \in D',\ l_j = c_p \in C\} \tag{8}$$

$$R_i = ranking(q_i, D_{c_p}) \tag{9}$$

If the search result $R_i$ satisfies $|R_i| < \theta$ with respect to the threshold $\theta$, the search result for the original set of documents $D$ is added.

$$R_i = ranking(q_i, D_{c_p}) \cup ranking(q_i, D) \tag{10}$$

As the category set $C$, we adopted 10 categories used in the community QA service Yahoo! Chiebukuro for Japanese subtask and 23 categories used in Yahoo! Answers for English subtask.

We also constructed a text classifier $text\_classifier()$ to classify the documents and queries into categories. We collected question-and-answer pairs for each category in Yahoo! Chiebukuro and Yahoo! Answers that contain URLs in their answers. Table 3 shows the number of collected question-and-answer pairs for each category. The number of question-and-answer pairs in each category was the average of 149.4 with the standard deviation of 6.28 for Yahoo! Chiebukuro and the average of 158.0 with the standard deviation of 99.26 for Yahoo! Answers. Although the number of question-and-answer pairs was low in some categories, we were able to collect at least 20 questions per category.

In addition, the words in all part-of-speech in the collected question-and-answer pairs are transformed into term frequency vectors, and the classifier is trained in a three-layered multilayered perceptron for each subtask. The multilayer perceptron was used because it showed the best performance compared to several methods such as Naive Bayes. The parameters of the MLP were as follows; the number of units in the hidden layer being 100, the activation function softmax, the number of epochs 300, the error function being cross entropy, and the optimization being Adam. The constructed classifier is denoted by $text\_classifier_{all,tf,mlp}()$.

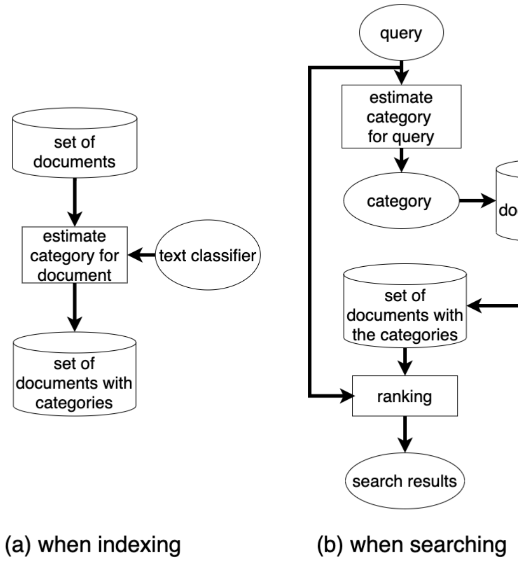
After applying the 10-fold cross validation, the correct answer rate of $text\_classifier_{all,tf,mlp}()$ was 0.48 for Japanese subtask and 0.66 for English subtask.

The threshold $\theta$ was set to $\theta = 1000$.

## 5 EXTRACTION OF TABLE HEADERS

To compensate for the short document length of the metadata, we implemented a method of extracting the header information of tables from the body of the statistical data and adding it to the document to be retrieved. The procedure is shown in Figure 3.

In order to deal with various formats of statistical data, preprocessing is performed on the statistical data. First, the statistical data are converted into images, and contour extraction is performed on the images to extract the regions corresponding to each cell (hereafter called cell regions). For each cell region, the vertical and horizontal division number, the character strings obtained by applying OCR to the cell region (hereafter called cell text), and the type of cell text (hereafter called the text type) showing whether it consists a string of only numeric characters, a string of characters including non-numeric characters, or an empty string. Here, the vertical and horizontal division number is defined as the division id to which the text belongs when the statistical table is divided into N divisions in the vertical and horizontal directions. In this case, we set N=6.


query
set of documents
estimate category for query
estimate category for document
text classifier
category
set of documents with categories
set of documents with the categories
set of documents with categories
ranking
search results

(a) when indexing (b) when searching

**Figure 2: Procedure of category search**

Next, we construct a cell classifier $cell\_classifier()$ that determines whether a cell is a header or not. To construct the classifier, we take the horizontal alignment of cells as a sequence, and determine whether each cell is a header or not as a sequence labeling problem[3]. For the features, we used the vertical and horizontal division numbers, cell text, and text types extracted in the preprocessing, as shown in Table 4, and conditional random fields (CRF) were used for training.

First, apart from the e-Stat statistical data provided by the organizers, we obtained 81 statistical data (with the header of the table containing both Japanese and English notation) from the same site, and created a dataset for a total of 352,624 cells, with labels indicating whether each cell belongs to the header or not. Of the dataset created, 70% of the randomly selected data set was used as training data and the remaining 30% was used as test data.

The identification results based on the test data are shown in Table 5. The F-measure for header identification was 0.839.

Using the constructed classifier, the set of strings in each cell $cell_s^{t_j}$ of the statistical data $t_j$ of document $d_j$ that are determined to be headers are extracted as the header $h_j$.

$$h_j = \{text(cell_s^{t_j}) | cell\_classifier(cell_s^{t_j}) = \text{'header'},\ cell_s^{t_j} \in t_j\} \quad (11)$$

Here, $text(x)$ is a function to get the string of $x$.

By concatenating the extracted header $h_j$ and the metadata $m_j$, we created the document $d_j^{m+h}$ that compensates for the short document length of the metadata.

$$d_j^{m+h} = m_j \cup h_j \quad (12)$$

$$D^{m+h} = \{d_j^{m+h}\} \quad (13)$$

The function $ranking()$ is used to rank the set of documents $D^{m+h}$ and return the result $R_i$. $R_i$ is described as follows:

$$R_i = ranking(q_i, D^{m+h}) \quad (14)$$

**Table 4: List of features used for CRF**

| feature | explanation |
|---|---|
| cell_text | text strings in the cell |
| vertical_number | vertical division number |
| horizontal_number | horizontal division numnber |
| text_type | text type in the cell (character, number, or blank) |

**Table 5: Results of cell classification with CRF**

| label | # of test data | correct | precision | recall | f-measure |
|---|---|---|---|---|---|
| header | 11953 | 9530 | 0.932 | 0.763 | 0.839 |
| not_header | 85060 | 82322 | 0.664 | 0.519 | 0.577 |

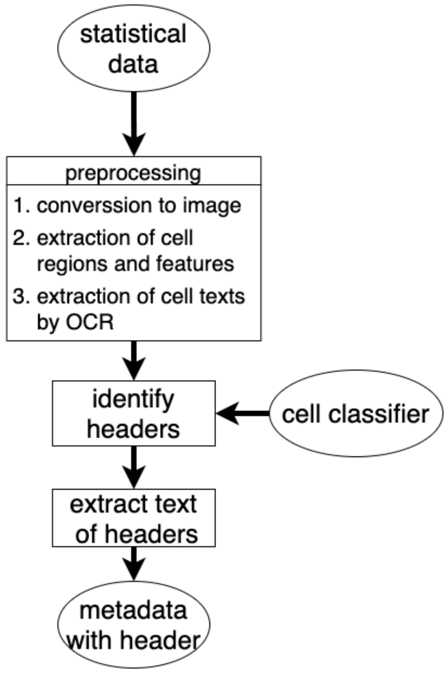


**Figure 3: Extraction of header information and metadata augmentation**

## 6 BERT-BASED RERANKING

We introduced the BERT-based reranking[6] as one of the comparison methods. Using the given query, a normal search is first performed for the inverted index, and the top ranked results are passed to the BERT-based reranking function for reranking.

$$R_i = ranking_{BM25}(q_i, D^{m+h}) \quad (15)$$

$$R_i' = reranking_{BERT}(q_i, R_i) \quad (16)$$

Specifically, inference by BERT is performed for each sentence of the candidate document, and the sentence level score is combined with the normal document score according to the following equation 17.

$$S_f = a \cdot S_{doc} + (1 - a) \cdot \Sigma_{i=1}^{n} w_i \cdot S_i \quad (17)$$

$S_{doc}$ is the document score before reranking, and $S_i$ is the top $i$ sentence score by BERT. $a$, $w_2$, and, $w_3$ are changed from 0 to 1 in

0.1 increments, and the reranking result by $S_f$ was evaluated by 5-fold cross valiation using the training data, to find the parameters that take the maximum MAP. As a result, $a = 0.9$, $w_2 = 0.1$, $w_3 = 0.0$ were found for Japanese subtask, and $a = 0.9$, $w_2 = 0.0$, and $w_3 = 0.1$ for English subtask.

## 7 OFFICIAL RUNS

We constructed our official runs by adopting the various combinations of category search, table header information, and the ranking method. The list of submitted official runs is shown in Table 6.

In KSU-J-1 and KSU-E-2, the set of documents to be retrieved $D^{m+h}$ obtained by adding the table header $h_j$ to $m_j$ composed of title and description values of metadata, is narrowed down by the QA category estimated by the query $q_i$, to which the ranking with BM25 were applied.

$$D^{m+h} = \{d_j^{m+h}\} = \{m_j \cup h_j\} \tag{18}$$

$$\begin{aligned} D_{c_p}^{m+h} &= \{d_j | (d_j, l_j) \in D^{m+h}, \\ &\quad l_j = text\_classifier_{all,tf,mlp}(q_i) \in C\} \end{aligned} \tag{19}$$

$$R_i = ranking_{BM25}(q_i, D_{c_p}^{m+h}) \tag{20}$$

In KSU-J-3 and KSU-E-4, the set of documents to be retrieved $D^{m+h}$ is narrowed down by the QA category estimated by the query $q_i$, to which the ranking with BM25 followed by reranking with BERT were applied.

$$D^{m+h} = \{d_j^{m+h}\} = \{m_j \cup h_j\} \tag{21}$$

$$\begin{aligned} D_{c_p}^{m+h} &= \{d_j | (d_j, l_j) \in D^{m+h}, \\ &\quad l_j = text\_classifier_{all,tf,mlp}(q_i) \in C\} \end{aligned} \tag{22}$$

$$R_i = ranking_{BM25}(q_i, D_{c_p}^{m+h}) \tag{23}$$

$$R'_i = reranking_{BERT}(q_i, R_i) \tag{24}$$

In KSU-J-5 and KSU-E-6, the ranking with BM25 were applied to the set of documents to be retrieved $D^m$ obtained from $m_j$ composed of title and description values of metadata.

$$D^m = \{d_j^m\} = \{m_j\} \tag{25}$$

$$R_i = ranking_{BM25}(q_i, D^m) \tag{26}$$

In KSU-J-7 and KSU-E-8, the ranking with BM25 followed by reranking with BERT were applied to the document set to be retrieved $D^m$.

$$D^m = \{d_j^m\} = \{m_j\} \tag{27}$$

$$R_i = ranking_{BM25}(q_i, D^m) \tag{28}$$

$$R'_i = reranking_{BERT}(q_i, R_i) \tag{29}$$

## 8 RESULTS AND DISCUSSION: OFFICIAL RUNS

The evaluation in the official run showed that the combined method of category search and BM25 scored 0.448 in nDCG@10 in the Japanese subtask and 0.255 in NDCG@10 in the English subtask, where each showed the highest score on this measure among all the official runs.

First, regarding the methods using the category search, we investigated how appropriately the categories could be classified. The ratio of classifying into the appropriate category was about 20% for Japanese subtask and about 50% for English subtask.

In Japanese subtask, many queries and documents were misidentified, but they were often misidentified into the same category, and as a result, narrowing down the searched documents were considered to have worked effectively, leading to high scores. It is necessary to improve the performance of text classifier by using the distributed representation as feature vectors.

In English subtask, many cases were confirmed in which familiar topics such as daily life were misidentified into business categories. In addition, regarding the queries classified into the appropriate categories, the ratio of nDCG@10 in the ranking result by BM25 was less than the average of nDCG@10 in the entire test query was about 40%.

Next, regarding the method using table header extraction, we investigated what kind of header was extracted. When we checked the header information extracted for KSU-J-1, 80% of it was meaningless. In KSU-J-1, in order to deal with statistical tables with various extensions, the statistical tables are once converted into images and the character strings in the cells are extracted using OCR[5]. Although the header position information could be specified by identifying the cell type, it is probable that the appropriate header information could not be extracted because of the failure of the character recognition. It is necessary to make enhancement such as improving the accuracy of OCR.

Finally, we investigated the causes of low scores for reranking based on BERT. The reranking used in this paper is to rerank the ranking result of BM25. Therefore, the ranking result based on BM25 alone by ORGJ and the ranking result based on BERT were compared. As a result, there was almost no duplication in the ranking results of the top 1000 in both Japanese and English subtasks. There is a possibility that an appropriate candidate document group has not been obtained in the search by BM25 using Anserini. In the future, we plan to improve so that appropriate candidate documents can be obtained from the query and verify the effect of reranking based on BERT.

## 9 EXTRA RUNS

We sought to improve our approach in response to the results of official runs. Specifically, we aimed to further improve the performance of category search by improving the accuracy of text classification at the time of submitting official runs. In addition, since it was difficult to improve the accuracy of OCR recognition in header extraction, we aimed to improve the acquisition of header information by another method of estimating the header region.

### 9.1 Category Search

Since the feature vector used in the text classifier at the time of submitting official runs was a simple one of term frequency in all part-of-speech, the vector was revised so that the accuracy can be expected to improve in extra runs.

First, we considered three types of part-of-speech, which are noun only, noun or verb, and all part-of-speech, and three types of vectors which are word2vec, GloVe, and fastText.

We also looked into four training methods, which are SVM, logistic regression (LR), random forests, and multilayer perceptron.

**Table 6: List of official runs**

| RUN | category search | table header | ranking | text classifier for category search | | | table header extraction |
|---|---|---|---|---|---|---|---|
| | | | | POS | vector | training | |
| KSU-J-1 | ✓ | ✓ | BM25 | All | TF | MLP | OCR+CRF |
| KSU-J-3 | | ✓ | BERT reranking | | | | OCR+CRF |
| KSU-J-5 | ✓ | | BM25 | All | TF | MLP | |
| KSU-J-7 | | | BERT reranking | | | | |
| KSU-E-2 | ✓ | ✓ | BM25 | All | TF | MLP | OCR+CRF |
| KSU-E-4 | | ✓ | BERT reranking | | | | OCR+CRF |
| KSU-E-6 | ✓ | | BM25 | All | TF | MLP | |
| KSU-E-8 | | | BERT reranking | | | | |

**Table 7: Results of official runs**

| RUN | nDCG@10 |
|---|---|
| KSU-J-1 | 0.421 |
| KSU-J-3 | 0.119 |
| KSU-J-5 | 0.448 |
| KSU-J-7 | 0.119 |
| KSU-E-2 | 0.255 |
| KSU-E-4 | 0.052 |
| KSU-E-6 | 0.255 |
| KSU-E-8 | 0.039 |

When 10-fold cross-validations were performed for all combinations of the above elements, the highest classification accuracy of 0.69 was obtained for the combination of noun + verb, fasttext, SVM or LR in Japanese subtask. In the English subtask, the highest classification accuracy of 0.58 was achieved when all part-of-speech, fasttext, SVM or LR were combined.

## 9.2 Extraction of Table Headers

The biggest problem of header extraction is that it is difficult to improve the recognition accuracy of character strings from cell region images by OCR. Therefore, we decided to develop a method of extracting the region including the header from the statistical data without using OCR.

We considered a method of examining whether or not any characters are included in each cell of the statistical data and of extracting headers based on changes in the number of cells containing characters.

The extraction procedure is shown in Alrogithm 1. First, initialize the number of non-empty cells in the previous column *prev* with 0, the column header *hdr_col* with an empty list, and store the number of columns of the statistical data in *max_col*. Repeat the followings from the first column to the last column *max_col*. If the number of non-empty cells *curr* in the current column is greater than *prev*, append the non-empty cells in the current column to the column header *hdr_col*. Update *prev* with *curr* and move on to the next column. When the iteration is over, return the column header *hdr_col* as $h_j^{col}$.

For the row header, the similar process obtained by replacing the columns with rows in the algorithm 1 is performed to extract the row header $h_j^{row}$.

**Algorithm 1** Extracting column headers from statistical data

```
Input: statistical data $sd$
Output: column headers $hdr\_col$
  $prev = 0$
  $hdr\_col = []$
  $max\_col = sd.column.\mathrm{length}$
  for $i = 1, \dots, max\_col$ do
    $curr = sd.column[i].unempty\_cells.\mathrm{length}$
    if $curr > prev$ then
      $hdr\_col.\mathrm{append}(sd.column[i].unempty\_cells)$
    end if
    $prev = curr$
  end for
  return $hdr\_col$
```

Meanwhile, for English subtask, it is difficult to access the cells in the same way as in Japanese subtask, since most of the statistical data are provided in pdfs. Therefore, for English subtask, we used a module called PDFminer to extract all the text information in pdf. By extracting all the text information, we get information that is not directly useful for search, such as numerical data, but the statistical data in pdfs often include comments and explanations about statistical data, which may be useful for search.

## 9.3 Submitted Extra Runs

We constructed each extra runs by adopting the various combinations of the text classifier for category search and the way of extracting header information from the table. The list of submitted extra runs is shown in Table 8.

First, extra runs for Japanese subtask will be described.

In KSU-J-EX-1, the set of documents to be retrieved $D^{m+h^r+h^c}$ obtained by adding the table headers $h_j^{row}$ and $h_j^{col}$ to $m_j$ composed of title and description values of metadata, is narrowed down by the QA category estimated by the query $q_i$ with the text classifier trained by SVM, to which the ranking with BM25 were applied.

$$\begin{aligned} D^{m+h^r+h^c} &= \{d_j^{m+h^r+h^c}\} = \{m_j \cup h_j^{row} \cup h_j^{col}\} \quad (30) \\ D_{c_p}^{m+h^r+h^c} &= \{d_j | (d_j, l_j) \in D^{m+h^r+h^c}, \\ &\quad l_j = text\_classifier_{N+V,fasttext,SVM}(q_i) \in C\} \quad (31) \\ R_i &= ranking_{BM25}(q_i, D_{c_p}^{m+h^r+h^c}) \quad (32) \end{aligned}$$

KSU-J-EX-2 replaces the set of documents to be retrieved $D^{m+h^r+h^c}$ in KSU-J-EX-1 with the set of documents $D^{m+h^r}$.

$$D^{m+h^r} = \{d_j^{m+h^r}\} = \{m_j \cup h_j^{row}\} \quad (33)$$

KSU-J-EX-3 replaces the set of documents to be retrieved $D^{m+h^r+h^c}$ in KSU-J-EX-1 with the set of documents $D^m$.

KSU-J-EX-{6,7,8} replace the text classifier trained with SVM and used in KSU-J-EX-{1,2,3} with the one trained with logistic regression, $text\_classifier_{N+V,fastText,LR}()$, respectively.

Next, extra runs for English subtask will be described.

In KSU-E-EX-4, the set of documents to be retrieved $D^{m+t}$ obtained by adding the entire statistical data $t_j$ to $m_j$ composed of title and description values of metadata, is narrowed down by the QA category estimated by the query $q_i$ with the text classifier trained by SVM, to which the ranking with BM25 were applied.

$$\begin{aligned} D^{m+t} &= \{d_j^{m+t}\} = \{m_j \cup t_j\} \quad (34) \\ D_{c_p}^{m+t} &= \{d_j | (d_j, l_j) \in D^{m+t}, \\ &\quad l_j = text\_classifier_{all,fasttext,SVM}(q_i) \in C\} \\ R_i &= ranking_{BM25}(q_i, D_{c_p}^{m+t}) \quad (35) \end{aligned}$$

KSU-E-EX-5 replaces the set of documents to be retrieved $D^{m+t}$ in KSU-E-EX-4 with the set of documents $D^m$.

KSU-E-EX-{9,10} replace the text classifier trained with SVM and used in KSU-E-EX-{4,5} with the one trained with logistic regression, $text\_classifier_{All,fastText,LR}()$, respectively.

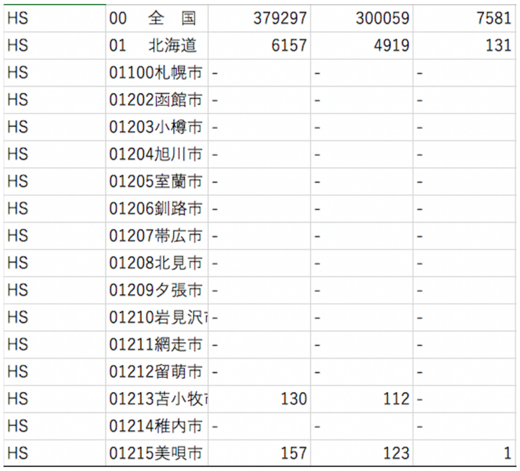
## 10 RESULTS AND DISCUSSION: EXTRA RUNS

The results of extra runs are shown in Table 9.

Some of the scores by the newly constructed methods in extra runs were found to be better than those by the methods used in the official runs and some otherwise, as shown in Table 9.

First, we analyze the Japanese subtask.

We investigated the effect of the improved text classifiers for category search on the extra runs. The values of nDCG@10 by KSU-J-EX-{3,8}, which is a combination of category search and BM25, were reduced by 0.06 and 0.07, respectively, compared with those by KSU-J-5. The accuracy of category classification by KSU-J-EX-3,8 and KSU-J-5 in category search was validated via 10-fold cross validation to increase from 0.23 to 0.68. KSU-J-EX-3,8 and KSU-J-5 have different part-of-speech for words used in vectors, vectorization methods, and training methods for classifiers, respectively. Although KSU-J-EX-3,8 allows more appropriate categories to be estimated, it is possible that the narrowed-down set of documents may not have been appropriate as the set of documents satisfying the information request as a result. In fact, the query "23-years-old, sleep time" was categorized into "life" in KSU-J-5 and "health" in KSU-J-EX-3,8, and most of the documents narrowed down in the "health" category were medical-related, failing to return the appropriate search results that matched the query.



| | | | | |
|---|---|---|---|---|
| HS | 00　全　国 | 379297 | 300059 | 7581 |
| HS | 01　北海道 | 6157 | 4919 | 131 |
| HS | 01100札幌市 | - | - | - |
| HS | 01202函館市 | - | - | - |
| HS | 01203小樽市 | - | - | - |
| HS | 01204旭川市 | - | - | - |
| HS | 01205室蘭市 | - | - | - |
| HS | 01206釧路市 | - | - | - |
| HS | 01207帯広市 | - | - | - |
| HS | 01208北見市 | - | - | - |
| HS | 01209夕張市 | - | - | - |
| HS | 01210岩見沢 | - | - | - |
| HS | 01211網走市 | - | - | - |
| HS | 01212留萌市 | - | - | - |
| HS | 01213苫小牧 | 130 | 112 | - |
| HS | 01214稚内市 | - | - | - |
| HS | 01215美唄市 | 157 | 123 | 1 |

**Figure 4: Example of not getting the appropriate row headers**

In addition, we investigated the impact of the improved extraction of header information of statistical data in extra runs. The values of nDCG@10 by KSU-J-EX-1,6, where the set of documents to be retrieved in KSU-J-EX-3,8 was augmented with row and column headers, were increased by 0.07 and 0.08, respectively, compared to those by KSU-J-EX-3,8. Similarly, the values of nDCG@10 by KSU-J-EX-2,7, where the set of documents to be retrieved of KSU-J-EX-3,8 was augmented by row headers only, were decreased by 0.08 and 0.07, respectively, compared to those by KSU-J-EX-3,8. In KSU-J-EX-1,6, the row and column headers seemed to be able to adequately augment the document set with metadata. Meanwhile, in KSU-J-EX-2,7, the augmentation by the row headers alone did not seem to lead to proper ranking. In fact, reviewing the contents of the row headers, we confirmed some examples in the statistical data that contain strings that are not directly useful for the search, such as in Figure 4. In this example, because the number of rows in the first and second columns of non-empty cells is the same, "HS" is extracted as a header, but a string of place names such as "全国 ("nationwide" in Japanese)" is not extracted as a header.

Next, we analyze the English subtask.

We investigated the effect of the improved text classifier for category search. The values of nDCG@10 by KSU-E-EX-5,10, a method combining category search and BM25, were reduced by 0.06 and 0.02, respectively, compared with those by KSU-E-6. The accuracy of category classification by KSU-E-EX-5,10 and KSU-E-6 in category search was validated via 10-fold cross validation to decrease from 0.53 to 0.42. KSU-E-EX-5,10 and KSU-E-6 differ in the part-of-speech of the words used for the vectors, the vectorization methods and the training methods of the classifier, respectively. A possible reason for the decrease in category classification accuracy is the bias in the number of data per category used for training. As shown in Table 3, the maximum and minimum number of data per

**Table 8: List of extra runs**

| RUN | category search | table header | ranking | text classifier for category search | | | table header extraction |
|---|---|---|---|---|---|---|---|
| | | | | POS | vector | training | |
| KSU-J-EX-1 | ✓ | ✓ | BM25 | N+V | fastText | SVM | ROW+COL |
| KSU-J-EX-2 | ✓ | ✓ | BM25 | N+V | fastText | SVM | ROW |
| KSU-J-EX-3 | ✓ | | BM25 | N+V | fastText | SVM | |
| KSU-J-EX-6 | ✓ | ✓ | BM25 | N+V | fastText | LogisticRegression | ROW+COL |
| KSU-J-EX-7 | ✓ | ✓ | BM25 | N+V | fastText | LogisticRegression | ROW |
| KSU-J-EX-8 | ✓ | | BM25 | N+V | fastText | LogisticRegression | |
| KSU-E-EX-4 | ✓ | ✓ | BM25 | All | fastText | SVM | All |
| KSU-E-EX-5 | ✓ | | BM25 | All | fastText | SVM | |
| KSU-E-EX-9 | ✓ | ✓ | BM25 | All | fasttext | LogisticRegression | All |
| KSU-E-EX-10 | ✓ | | BM25 | All | fastText | LogisticRegression | |

**Table 9: Results of extra runs**

| RUN | nDCG@10 |
|---|---|
| KSU-J-1 | 0.391 |
| KSU-J-3 | 0.110 |
| KSU-J-5 | 0.413 |
| KSU-J-7 | 0.110 |
| KSU-J-EX-1 | 0.426 |
| KSU-J-EX-2 | 0.276 |
| KSU-J-EX-3 | 0.353 |
| KSU-J-EX-6 | 0.426 |
| KSU-J-EX-7 | 0.276 |
| KSU-J-EX-8 | 0.342 |
| KSU-E-2 | 0.240 |
| KSU-E-4 | 0.051 |
| KSU-E-6 | 0.240 |
| KSU-E-8 | 0.038 |
| KSU-E-EX-4 | 0.042 |
| KSU-E-EX-5 | 0.181 |
| KSU-E-EX-9 | 0.043 |
| KSU-E-EX-10 | 0.216 |

category collected from Yahoo! Chiebukuro was 163 and 140, respectively, while the maximum and minimum number of data per category collected from Yahoo! Answers was 260 and 20. Therefore, the classifier using fastText may have learned the effects of these biases more faithfully than that using term frequency vectors, resulting in the decrease in classification accuracy. It is possible that correcting the bias in the number of data per category could improve the score.

We also investigated the impact of the improved extraction of header information of statistical data in extra runs. The values of nDCG@10 by KSU-E-EX-5,10, where the document set to be retrieved of KSU-E-EX-5,10 was augmented with the entire statistical data, were reduced by 0.14 and 0.17, respectively, compared to the values by KSU-E-EX-5,10. The rankings were considered to be negatively impacted by the inclusion of a lot of information that was not directly useful to the search, such as numerical data, because it was augmented by the statistical data as a whole. In fact, when we checked the contents of the documents, we found a lot of information that was not likely to be directly related to the query, such as numerical data and cautions in handling the documents. It is possible that the score could be improved by applying a technique that extracts only the headers.

## 11 CONCLUSION

In this paper, we describe the system and results of Team KSU for Data Search Task in NTCIR-15. We introduced the categorical search that narrows the set of documents to be retrieved by category, a method of extracting the header information of the table from the main body of statistical data to augment documents to be searched, and the ranking method of BM25 and of reranking based on BERT. The evaluation in the official run showed that the combined method of category search and BM25 scored 0.448 in nDCG@10 in the Japanese subtask and 0.255 in NDCG@10 in the English subtask, where each showed the highest score on this measure among all the official runs.

## REFERENCES

[1] Inches G, Carman M, and Crestani F. 2010. *Advances in Information Retrieval.* Statistics of online user-generated short documents. 649–652 pages.

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

[3] Yusuke Matsui and Hisashi Miyamori. 2014. Automatic recognition of statistical table data for the purpose of searching the evidence of trend information (in Japanese). , B4-5 pages.

[4] Mainichi Shimbun. 1996. CD-Mainichi Shimbun 1995 Data Collection, Nichigai Associates. http://www.nichigai.co.jp/sales/mainichi/mainichi-data.html

[5] Shuuhei Sowa and Hisashi Miyamori. 2017. Recognition and semantic interpretation method of heading hierarchy in statistical table with complicated structure (in Japanese). , B4-4 pages.

[6] Shengjin Wang Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. , 19–24 pages.