TUA1 at the NTCIR-13 OpenLiveQ Task

Mengjia He
Tokushima University, Japan
c501647002@tokushima-u.ac.jp

Xin Kang
Tokushima University, Japan
Kang-xin@is.tokushima-u.ac.jp

Fuji Ren
Tokushima University, Japan
ren@ is.tokushima-u.ac.jp

ABSTRACT
Our group submitted the OpenLiveQ task of NTCIR-13. With th openliveq tool offered from the organizers, we compare a baseline result with three results ranked by RF (Random Forests). It shows the result ranked by RF in 1000 bags is closest to baseline, but still lower in average.

Team Name
TUA1

Subtasks
NTCIR-13 OpenLiveQ(Open Live Test for Question Retrieval)

Keywords
cQA; OpenLiveQ; Random Forests

1. INTRODUCTION
The OpenLiveQ Task, called Open Live Test for Question Retrieval for full, is a new task proposed in NTCIR-13. It's a ranking problem on Community Question Answering (cQA) services in which users can ask questions and get answers from other users. Similar with the search engine and QA system, the problems such as ambiguous and criteria of relevance can also be considered.

On this task, we use Random Forests Algorithm as the learning method for the ranking with the help of tools offered by the organizers. In this paper, we will describe our experiment on the offline phase, and discuss the results of the offline test. The online test result will be exhibited at the last simply.

2. OPENLIVEQ TASK
The OpenLiveQ task is defined as: from a query including some questions with their answers, rank these questions and return the list of them. The task consists of three phases [1]:

(1) Offline Training Phase

(2) Offline Test Phase

(3) Online Test Phase

2000 queries are collected from Yahoo! Chiebukuro, 1000 for training and 1000 for testing. Each query has the top 1000 questions from the current Yahoo! Chiebukuro search system of December 1-9 in 2016, with the total number of 1,967,274. All these questions include [2]:

- Query ID (a query by which the question was retrieved)
- Rank of the question in a Yahoo! Chiebukuro search result for the query of Query ID
- Question ID,
- Title of the question
- Snippet of the question in a search result
- Status of the question (accepting answers, accepting votes or solved)
- Last update time of the question
- Number of answers for the question
- Page view of the question
- Category of the question
- Body of the question
- Body of the best answer for the question

Some questions have clickthrough data with the total number of 440,163. This kind of data includes [2]:

- Query ID (a query by which the question was retrieved)
- Question ID
- Most frequent rank of the question in a Yahoo! Chiebukuro search result for the query of Query ID
- Clickthrough rate
- Fraction of male users among those who clicked on the question
- Fraction of female users among those clicked on the question
- Fraction of users under 10 years old among those who clicked on the question
- Fraction of users in their 10s among those who clicked on the question
- Fraction of users in their 20s among those who clicked on the question
- Fraction of users in their 30s among those who clicked on the question
- Fraction of users in their 40s among those who clicked on the question
- Fraction of users in their 50s among those who clicked on the question
- Fraction of users over 60 years old among those who clicked on the question

The task is limited to Japanese on the language scope, but the organizers provide a tool [3] for feature extraction so there is no need for Japanese NLP. The tool called "openliveq", is a python package using features such as TF-IDF and BM25, and learning to rank with RankLib. Features used by the tool are listed in Table 3 of [4].

3. EXPERIMENT
Random Forests [5], or random decision forests are an ensemble learning method for tasks like classification or regression. It can naturally be used to rank the importance of variables in

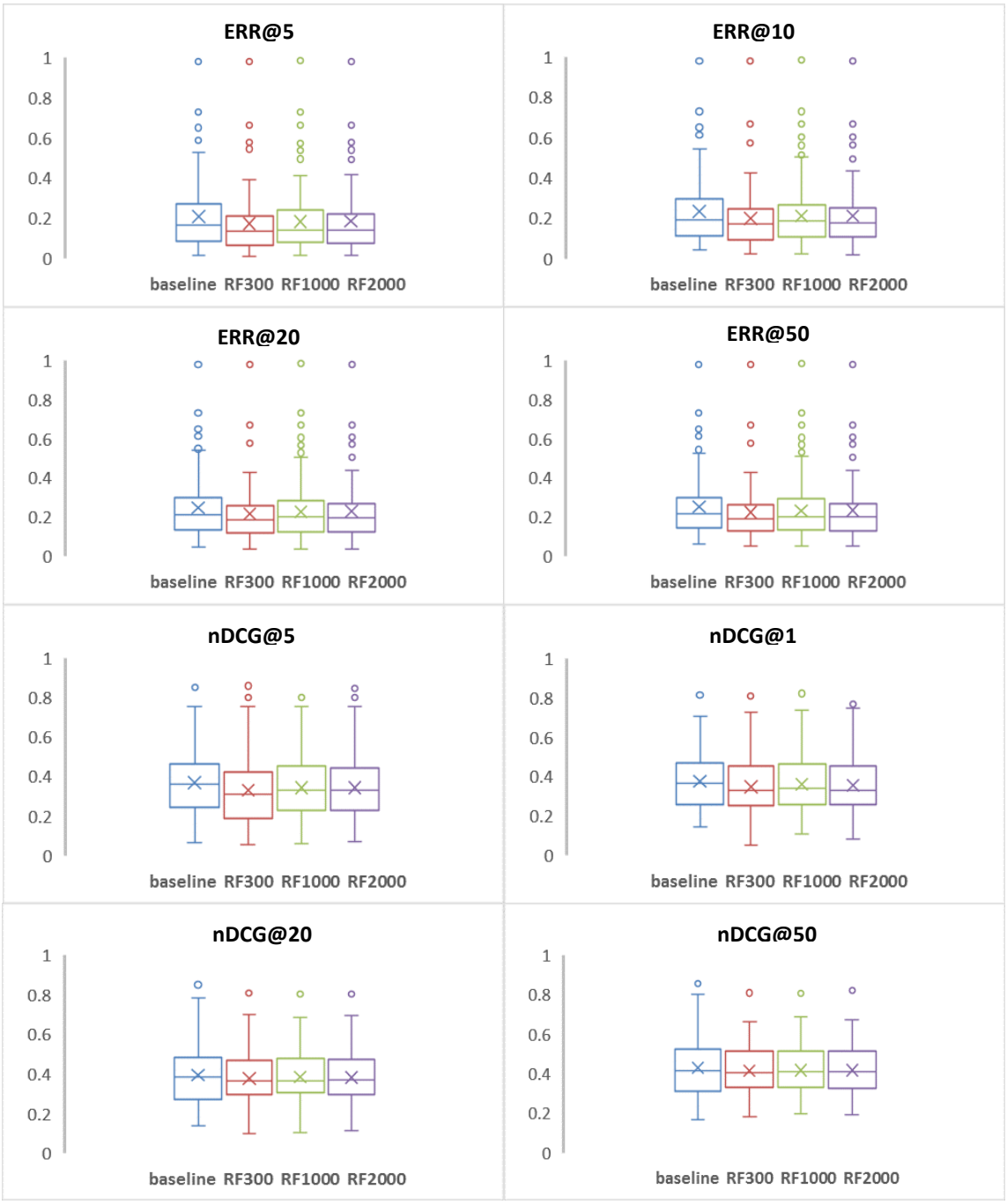

Figure 1. Rank evaluation box plots

classification or regression problem. Its learning speed is fast even the number of parameters is big, and the computing of the importance of features is considered to be necessary on this task.

We treat the result of the "openliveq" tool's first running as the baseline, and compare it with three results using Random Forests at 300 bags (RF300), 1000bags (RF1000) and 2000bags (RF2000) for learning to rank.

The evaluation metrics are nDCG (normalized discounted cumulative gain), ERR (expected reciprocal rank) and Q-measure.

The nDCG is one of the accuracy evaluation index for ranking problem. The nDCG@k score is calculated as

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

$$\text{nDCG@k} = \frac{DCG@k}{idealDCG@k}$$

where k is the number of ranking and rel_i means the relevance.

The ERR [6] is also an accuracy evaluation index on ranking problem with littler computing time than nDCG. It can be computed as

$$\mathrm{ERR} = \sum_{r=1}^{n} \frac{1}{r} \prod_{i=1}^{r-1} (1 - R_i) R_r$$

where n is the number of documents in the ranking and R_i is the probability that a document satisfies the user. The r is the position the users stops.

Q-measure [7] was proposed in NTCIR-4 and can be calculated as

$$\mathrm{Q-measure} = \frac{1}{R} \sum_{1 \le r \le L} isrel(r) \frac{cg(r) + count(r)}{cig(r) + r}$$

where the cg(r) is called as cumulative gain, and cig(r) is the cumulative gain while there's an ideal ranking result.

After submitting our four ranking results, we received the ERR@5, ERR@10, ERR@20, ERR@50, nDCG@5, nDCG@10, nDCG@20, nDCG@50 and Q- measure scores as the offline evaluation results.

The ERR and nDCG evaluation results are shown in Fig.1. RF1000 shows better ranking quality than RF300 and RF2000, but still lower than the baseline. Among the three RF results in average, RF1000 also shows better scores than the other two, 0.014578 higher than RF300 and 0.000719 higher than RF2000 in nDCG@5, 0.012908 higher than RF300 and 0.006933 higher than RF2000 in nDCG@10, 0.005926 higher than RF300 and 0.001258 higher than RF2000 in nDCG@20, 0.002467 higher than RF300 and 0.001191 higher than RF2000 in nDCG@50, but RF's performance is no as well as the baseline, while 0.023746 lower in nDCG@5, 0.015303 lower in nDCG@10, 0.009698 lower in nDCG@20 and 0.01223 lower in nDCG@50. However, RF2000 shows best scores among the RF results in ERR evaluation in average that, 0.013994 higher than RF300 and 0.002465 higher than RF1000 in ERR@5, 0.011555 higher than RF300 and 0.000633 higher than RF1000 in ERR@10, 0.011215 higher than RF300 and 0.001502 higher than RF1000 in ERR@20, 0.011213 higher than RF300 and 0.00164 higher than RF1000 in ERR@50, but still no as well as the baseline that, 0.023356 lower in ERR@5, 0.021397 lower in ERR@10, 0.018434 lower in ERR@20 and 0.018843 lower in ERR@50.

In the Q-measure averaged evaluation results shown in Fig.2, RF1000 performs best among the RF results, 0.000197 better than RF300 and 0.000442 better than RF2000, but 0.009412 lower than the baseline.

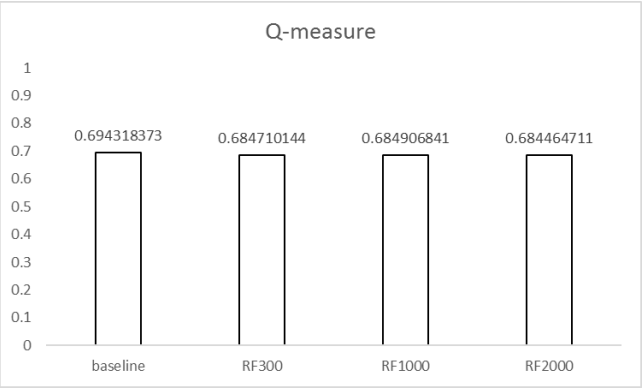

Figure 2. Q-measure in average

4. RESULTS AND CONCLUSIONS

In this paper, we described our work in the OpenLiveQ Task of the NTCIR-13. The OpenLiveQ Task is a ranking problem on Community Question Answering (cQA) services. We used Random Forests as the learning method for ranking and compared the results under 300 bags, 1000 bags and 2000 bags with the baseline. As a result of offline evaluation, Random Forests shows the best performance on 1000 bags, and the improvement from 300 bags to 2000 bags suggest that the performance of Random Forests will not improve much after 1000 bags. Compared with the baseline shows that Random Forests is still not enough to improve the performance on ranking, and other learning methods should be considered. Fig.3 shows the online credit from the Online Test Phase.

Our future work on OpenLiveQ will focus on some other learning methods trying.

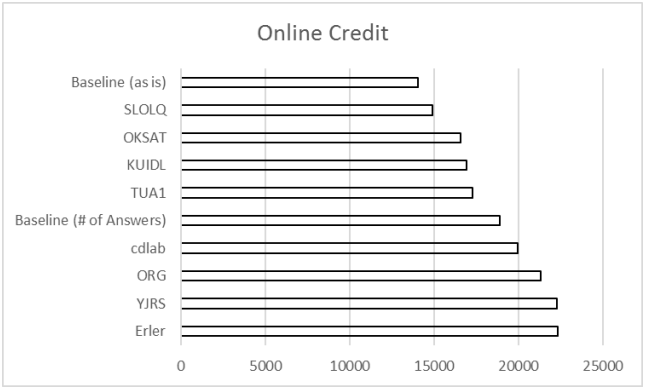

Figure 3. Ranking of the online credit

5. ACKNOWLEDGMENTS

This research has been partially supported by the Ministry of Education, Science, Sports and Culture of Japan, Grantin-Aid for Scientific Research(A), 15H01712.

6. REFERENCES

[1] Makoto P.Kato, Takehiro Yamamoto, Tomohiro Manabe, Akiomi Nishida and Sumio Fujita, Overview of the NTCIR-13 OpenLiveQ Task

[2] http://www.openliveq.net/?locale=en

[3] https://github.com/mpkato/openliveq

[4] Tao Qin, Tie-Yan Liu, Jun Xu, Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval, Information Retrieval, Volume 13, Issue 4, pp. 346-374, 2010

[5] Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)

[6] Olivier Chapelle, Donald Metlzer, Ya Zhang, Pierre Grinspan, Expected reciprocal rank for graded relevance, Proceedings of the 18th ACM conference on Information and knowledge management, pp. 621-630, 2009

[7] Sakai. T, Ranking the NTCIR Systems based on Multigrade Relevance. AIRS 2004 Proceedings (2004) 170-177