SLWWW at the NTCIR-15 WWW-3 Task

Masaki Muraoka
Waseda University
muraokamasaki@suou.waseda.jp

Zhaohao Zeng
Waseda University
zhaohao@fuji.waseda.jp

Tetsuya Sakai
Waseda University
tetsuyasakai@acm.org

ABSTRACT

The SLWWW team participated in the English subtask of the NTCIR-15 We Want Web-3 (WWW-3) Task. This paper describes our approach and results from the WWW-3 task. We utilized learning to rank models which were trained on the MQ2007 and MQ2008 datasets.

TEAM NAME

SLWWW

SUBTASKS

English

1 INTRODUCTION

The SLWWW team participated in the English subtask of the NTCIR-15 We Want Web-3 (WWW-3) Task [6]. This report describes our approaches in the ad-hoc web task and discusses the official results. Four of our runs utilized learning to rank models from the Ranklib package [1], and the fifth run used a tuned BM25 from Anserini [9].

We aim to reproduce the target run of THUIR [10] in the WWW-2 English subtask **THUIR-E-CO-MAN-Base-2**, which was based on LambdaMART in Ranklib.

2 METHODOLOGY

The details of the learning to rank models are described in this section.

2.1 Dataset

We adopted the LETOR4.0 dataset [4] for training. This dataset consists of the Gov2 web page collection and queries from the MQ2007 and MQ2008 query sets. In addition, the dataset provides computed features for each query-document pair. The validation set used was the NTCIR-13 WWW English test collection. This collection contains 100 queries and 22,912 relevance assessments from the ClueWeb12-B13 corpus. The relevance assessments used are the updated qrels from NTCIR-14 CENTRE [5].

The test set of the NTCIR-15 WWW-3 contains 160 topics, 80 from WWW-2 and 80 new topics. We rerank the documents provided by the baseline. The results of WWW-2 are not used by the learning to rank models.

2.2 Feature Extraction

The top 1000 documents for each query in the validation and test sets were extracted using Anserini and preprocessed using NLTK. The preprocessing includes tokenization, stripping the document of punctuation and stopwords, and stemming. We then computed the seven of the eight features shown in Table 1 for four fields: the title, anchor text, URL and whole document. The only exception was inverse document frequency, in which the inverse document frequency of the whole document was reused four times, as we were unable to procure it for the other fields. This resulted in a total of 32 features for each document-query pair.

Table 1: Extracted features

feature name
tf (term frequency)
idf (inverse document frequency)
tf*idf
document length
BM25 score
LMIR.ABS
LMIR.JM
LMIR.DIR

With regards to the training set, LETOR4.0 provides 46 features for each relevance assessment, in which our 32 features are a subset of. We selected the relevant 32 features for training our models.

2.3 Model Selection

The Ranklib package implements eight learning to rank algorithms, as shown in Table 2. We trained each algorithm with the same features and the default parameters, with the training metric being NDCG@10. The default parameters for each model are provided in the Ranklib home page [1]. These eight models were then validated on the WWW test collection whose results are included in Table 3. From these eight algorithms, we selected three, being Rankboost, AdaRank and Coordinate Ascent, as they performed well on the validation set. In addition, we included LambdaMART as it is a reproduced run from last year despite it performing badly on the validation set.

Table 2: Learning to rank algorithms in Ranklib

MART (Multiple Additive Regression Trees)
RankNet
RankBoost
AdaRank
Coordinate Ascent
LambdaMART
ListNet
Random Forests

2.4 Run Breakdown

We submitted five runs. The first four runs were generated from the learning to rank models described in the previous section. Due to a mistake, all were labelled reproduced (REP) runs, however only

Table 3: Mean nDCG@10 scores over the WWW-1 topic set (n=100) based on the WWW-2 CENTRE qrels

Algorithm	nDCG@10
MART	0.3510
RankNet	0.3313
RankBoost	0.3746
AdaRank	0.3699
Coordinate Ascent	0.3470
LambdaMART	0.2763
ListNet	0.2865
Random Forests	0.3184

run 4 (LambdaMART) is a reproduced run. The other runs are new (NEW) runs trained in the same way. The final run is a tuned BM25 using Anserini.

SLWWW-E-CO-REP-1. Ranklib RankBoost.
RankBoost [2] is a boosting algorithm that works by combining many "weak" rankings which are only weakly correlated with the optimal ranking into a single accurate ranking. It maintains a distribution over pairs of documents which emphasizes the important pairs the weak learner should learn to order correctly.

SLWWW-E-CO-REP-2. Ranklib AdaRank.
AdaRank [8] is another boosting algorithm that is inspired by AdaBoost. It maintains a distribution over queries instead, where higher weights are assigned to queries that should be focused on. AdaRank is shown to significantly outperform RankBoost on multiple datasets [8].

SLWWW-E-CO-REP-3. Ranklib Coordinate Ascent.
Coordinate Ascent [3] is a linear feature-based model. It calculates the ranking score from a linear combination of the features.

SLWWW-E-CO-REP-4. Ranklib LambdaMART.
LambdaMART [7] combines two ranking algorithms, LambdaRank and the boosted trees of MART.

SLWWW-CD-NEW-5. Anserini BM25 ($k = 1.1, b = 0.7$)
A BM25 run using Anserini, tuned on the WWW-2 test collection. A grid search was conducted with step size of 0.1 over the k and b parameters to maximize nDCG@10. Despite being marked *CD*, this only uses the content field of the topic due to a mistake.

3 RESULTS AND DISCUSSION

A summary of the runs are shown in Table 4. These metrics are the mean values over the 80 WWW-3 test topics.

3.1 Analysis

RankBoost, AdaRank and Coordinate Ascent yield reasonable metrics, but did not perform statistically significantly better than the baseline. LambdaMART performed the worst out of all five runs, and statistically significantly worse that the baseline for mean nDCG and mean Q. This could be expected given its poor performance in the validation set. Although we tried to reproduce THUIR's methodology in WWW-2, there were two differences that could contribute to a difference in evaluation. Firstly, we used the features provided by the LETOR4.0 dataset directly for training, whereas THUIR's approach was to compute the metrics manually. It is unclear if there were any differences between the features. Secondly, we were unable to obtain the idf for the title, anchor page and URL fields for the ClueWeb12-B13 corpus. We resorted to using the idf of the whole document for training and evaluation.

Table 4: Official results of our runs (Mean scores at cutoff 10 over the 80 WWW-3 topic set)

Run	nDCG	Q	ERR	iRBU
SLWWW-E-CO-REP-1	0.5189	0.5366	0.6397	0.8773
SLWWW-E-CO-REP-2	0.6227	0.6359	0.7345	0.9040
SLWWW-E-CO-REP-3	0.5719	0.5822	0.7061	0.8919
SLWWW-E-CO-REP-4	0.4465*	0.4531*	0.5804	0.8220
SLWWW-E-CD-NEW-5	0.6291	0.6445	0.7362	0.8981

*performs statistically significantly worse than baseline

The BM25 run performed the best among the five runs, which shows that a well tuned classical ranking algorithm can still be competitive with learning to rank models.

3.2 Poorly Performing Topics

In this section we discuss how well the models performed for individual topic. Table 5 shows the five worst performing (mean nERR) topics over all models submitted by all teams. The five worst performing topics for nDCG are similar in ranking as well.

Table 5: Hardest topics in terms of Average nERR over all WWW-3 runs

ID	Average nERR	Content
0132	0.1354	Friends
0169	0.1447	The origin of paper
0144	0.2715	Origin of Teacher's Day
0147	0.2872	bird nest effect
0153	0.3219	freeweblayouts

Of the 37 runs submitted, 17 had the topic *"Friends"* as the worst scoring topic for nERR. In addition, 36 of the 37 runs are *CO* runs, using only the content field. The description field for this topic reads *"Friends is a very famous TV series, you want to know in which year it first aired"*. Therefore it is likely that most rankers are unable to interpret the intent of the query correctly from the content field alone.

By digging into the qrels for the topic *"Friends"*, we discovered that out of the 18 relevant documents, only 6 of them contain the phrase "1994", which answers the description of this topic. The single document that was given a graded relevance score of 3 did not contain this phrase, while 6 of the 10 documents with a relevance score of 2 do contain this phrase. This might indicate some flaws with the way the relevant assessments were conducted, as we would expect highly scored documents to be more relevant to the topic. In

particular, a closer inspection of document with a relevance score of 3 showed that it had nothing to do with the TV show "Friends". None of the 7 documents with a relevance score of 1 contains "1994", but it is possible that this set of documents could contain marginally relevant information to the topic.

The next two topics, *"The origin of paper"* and *"Origin of Teacher's Day"* are the only two topics that contain the word "origin" in the content field. It is possible that rankers have hard time learning the meaning of this word.

3.3 Comparison of Rankers

The failure of our learning to rank models to outperform BM25 warrants some discussion. We examine a few topics that contribute to the difference in scores in two of our runs, the tuned BM25 (*SLWWW-E-CD-NEW-5*) and Ranklib AdaRank (*SLWWW-E-CO-REP-2*). Topics where one ranker performs substantially better than the other, resulting in the largest difference in nERR scores are shown in Tables 6 and 7.

Table 6: Topics where BM25 scored substantially better than AdaRank (nERR)

ID	Content	BM25	AdaRank	Diff
0160	akron beacon journal	0.9994	0.3034	0.6960
0132	Friends	0.5635	0.1602	0.4033

Table 7: Topics where AdaRank scored substantially better than BM25 (nERR)

ID	Content	BM25	AdaRank	Diff
0139	Rental contract	0.3173	0.7595	0.4422
0123	george washington university	0.3604	0.7865	0.4261

As BM25 scored better than AdaRank in mean ERR, it is not surprising that there are 42 out of the 80 topics where BM25 had a higher ERR score. There were two topics that had a large difference in scores, as seen in Table 6. The first topic, *"akron beacon journal"* had a high (0.9994) ERR for BM25 and a low (0.3034) ERR for AdaRank. From the description, *"You want to find the official website of Akron Beacon Journal"*, we can see that this topic has a navigational intent. However we are not able to conclude that BM25 performs better than AdaRank for navigational queries as the other navigational queries in the topicset (Topics 0102, 0112, 0153) did not produce a large difference in scores.

Relevance assessment for this topic was not entirely satisfactory. A quick look at a few documents with a relevance grade of 3 showed that they belonged to the Ohio State University or the City of Akron. The top ranked nonrelevant documents returned by AdaRank are shown in Table 8. These documents contain the phrase "akron beacon journal" multiple times but they belong to the domain "http://www.sportspyder.com/". These two documents mainly contributed to AdaRank scoring poorly for this topic, as the rest of the documents in the top 10 were graded relevant. While BM25 scored much better, a check of its first two ranked documents did not reflect the expected relevance. The top two documents returned belonged to Config.com and the American Friends Service Committee, but both received a relevance score of 3. While all four documents were not the official webpage of the Akron Beacon Journal, they contained links to it. Akron Beacon Journal is a newspaper and thus many documents, particularly news aggregation sites, will mention their name and link to their official website. Therefore, it is possible that the navigational intent of the query was not prioritized by either ranker, given that only the content field was used.

For the second query *"Friends"*, both scores fall below their mean ERR of about 0.73, given how poorly all rankers did for this particular topic. The top ranked nonrelevant documents returned by AdaRank are shown in Table 9, and they all return documents that contain the word "Friends" many times. This behavior is expected as AdaRank was trained with features including term frequency. The context of the query cannot be known from the content field alone. Interestingly, BM25 scores much higher than both AdaRank and the topic's average ERR of 0.1354. However, it seems that it is due to coincidence as inspection of the top 10 documents returned for both AdaRank and BM25, none of them contain any mention to the year "1994" when the TV show first aired.

The top documents returned by both rankers for the queries *"akron beacon journal"* and *"Friends"* were documents that matched the query phrase well. While this might be the expected behavior for BM25, we wonder if more complex features can be learned by learning to rank models. Alternatively, more rankers could incorporate both the content and description fields to capture the full intent of the query.

4 CONCLUSIONS

In this task we utilized learning to rank models and trained them on LETOR4.0. Unfortunately, we were unable to reproduce THUIR's LambdaMART results in WWW-2, but perhaps it was due to a difference in our methodology. Our study on the worst performing topics discussed how the content field alone was not sufficient in conveying the intent of the query, which in turn led to questions about the validity of the relevance assessments. Finally, analysis of topics which BM25 performed substantially better than AdaRank showed that the main difference was due to the differences in relevance grading.

REFERENCES

[1] Van Dang. 2012. *The Lemur Project-Wiki-RankLib.* Lemur Project. http://sourceforge.net/p/lemur/wiki/RankLib

[2] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. In *Proceedings of The Journal of Machine Learning Research.* 933–969.

[3] Donald Metzler and W. Bruce Croft. 2007. Linear feature-based models for information retrieval. In *Proceedings of Information Retrieval*, Vol. 10. 257–274.

[4] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. (2013). arXiv:arXiv:1306.2597

[5] Tetsuya Sakai, Nicola Ferro, Ian Soboroff, Zhahao Zeng, Peng Xiao, and Maria Maistro. 2019. Overview of the NTCIR-14 CENTRE Task. In *Proceedings of NTCIR-14.* 494–509. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/01-NTCIR14-OV-CENTRE-SakaiT.pdf

[6] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, and Ian Soboroff. 2020. Overview of the NTCIR-15 We Want Web with CENTRE (WWW-3) Task. In *Proceedings of NTCIR-15.* to appear.

[7] Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. In *Proceedings of Information Retrieval*, Vol. 13. 254–270.

Table 8: Top nonrelevant document returned by AdaRank for topic "akron beacon journal"

Rank	ID	Title
1	clueweb12-1312wb-14-19400	The Latest Cleveland Cavaliers News (Akron Beacon Journal) | SportSpyder
2	clueweb12-1310wb-87-17847	The Latest Cleveland Cavaliers News (Akron Beacon Journal: Cleveland Cavaliers) | SportSpyder

Table 9: Top nonrelevant document returned by AdaRank for topic "Friends"

Rank	ID	Title
1	clueweb12-0208wb-18-28489	Friends Myspace Graphics Friends Images Best Friends
2	clueweb12-0716wb-57-18215	Friends Myspace Graphics Friends Images Good Friends
3	clueweb12-0406wb-30-04332	Friends Myspace Graphics Friends Images

[8] Jun Xu and Hang Li. 2007. AdaRank: a boosting algorithm for information retrieval. In *Proceedings of SIGIR 2007*. 391–398.

[9] 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* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1253–1256. https://doi.org/10.1145/3077136.3080721

[10] Yukun Zheng, Zhumin Chu, Xiangsheng Li, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. THUIR at the NTCIR-14 WWW-2 Task. In *Proceedings of NTCIR-14*. 472–480.