Northeastern University Runs at the TREC13 Crowdsourcing Track

Maryam Bashir, Jesse Anderton, Virgil Pavlu, Javed A. Aslam
College of Computer and Information Science
Northeastern University, Boston, USA
{maryam,jesse,vip,jaa}@ccs.neu.edu

February 5, 2014

Abstract

The goal of the TREC 2013 Crowdsourcing Track was to evaluate approaches to crowdsourcing high quality relevance judgments for web pages and search topics. This paper describes our submission to Crowdsourcing track. Participants of this track were required to assess documents judged on a six-point scale. Our approach is based on collecting a linear number of preference judgements, and combining these into nominal grades using a modified version of QuickSort algorithm.

1 Introduction

We have participated in Crowdsourcing Track of TREC 2013 which required collecting relevance judgments for web pages and search topics. This year's Crowdsourcing Track offered two entry levels for participation, basic and standard. Basic task required relevance assessment of approximately 3,500 documents (a subset of NIST pool, 10 topics), whereas standard task required relevance assessment of approximately 20,000 documents (entire NIST pool, 50 topics). We have participated in basic task of the track. Instead of the usual nominal judgments ``how relevant is this web page?'', our assessing system used *preference judgments* ``which one of these two pages is more relevant?''. Evaluation of adhoc documents works better with preferences for following reasons:

- **reliability.** Traditionally, assessors are asked to give absolute relevance grades to each document with respect to some topic. However, studies have shown that assessors can give more reliable judgments if they are asked which of a pair of documents they prefer, i.e. "is document A better than document B?" [2].
- **consistency**. A much larger agreement among assessors is observed from preference judgments, most likely because assessors do not have to guess or interpret the given grade scale as they have to do on nominal judgments.
- **training.** Another advantage of using preferences is that many popular learning-to-rank algorithms such as RankBoost [3] and RankNet [1] are trained on preferences (we are not using here such learning algorithm). When preferences are not available, which is often the case, these algorithms need to infer preferences from the absolute relevance judgments collected from assessors; some information is lost during this process, leading to many ties between documents. This suggests that preferences can be used to improve the training of learning-to-rank algorithms that use such a pairwise approach.

The use of preference judgments on document pairs, as opposed to absolute judgments on documents, for IR evaluations creates new challenges. There are $\binom{n}{2}$ unique pairs of documents for a list of n documents, which means that the number of judgments we need to collect increases to $O(n^2)$. Since collecting judgments is costly (even with Mechanical Turks), we need a mechanism for collecting these preferences efficiently.

In the first stage we are approaching the assessing problem as a sorting-the-documents task. Our goal is to minimize the number of judgments needed to sort all documents and find true differences in the performance of retrieval systems. We have implement a QuickSort-like algorithm, using preference judgments (comparisons), so that web documents can be organized into grades following the preferences between each document and preselected special "pivot" documents. Such an algorithm reduces in general the number of judgments needed to fully order a list, as the rate of growth in the number of comparisons is $O(n \lg n)$, considerably slower than the $O(n^2)$ growth rate for all comparisons. Our algorithm uses constant number of pivots (given by the grade scale, typically 0-4 integers), so the total number of comparison is reduced to $O(n)$ in number of documents assessed.

The rest of paper is organized as follows: Section 2 describes our sorting algorithm and collection of preference judgments on document pairs. Section 3 provides details on our interface design and experimental setup. Section 4 describes the results of experiments. The final section, Section 5, concludes the paper with a description of future work.

2 Methodology

In this Crowdsourcing track, participants are actually free to use or not use crowdsourcing techniques however they wish. As discussed in Section 1, preference judgments are easier for assessors as compared to nominal grades. We have used a modified version of QuickSort algorithm for sorting documents. This algorithm can be divided into following steps:

1. Select pivot documents such that each pivot belongs to a different relevance class and one pivot is selected from each relevance class.

2. Sort pivot documents

3. For each document, search for the correct position between the sorted pivots.

Each of the above steps is explained in the following sections. Only one assessor (graduate student) was used in these experiments.

2.1 Pivot Selection

For pivot selection, we want to sample a subset of documents such that it has at least one document from all possible classes of relevance for a particular topic. Our goal is to minimize the number of documents we need to examine in order to select documents from all possible relevance classes. We sampled documents for pivot selection in the following manner. First, we calculated a prior relevance score for each document using BM25. This produced an initial ranking of the documents for each topic. We sampled every fifth document from this rank list for pivot selection. In addition to every fifth document, we also sampled top 10 documents from the rank list. The motivation behind this strategy is that BM25 rank list has higher ratio of relevant documents at top of list as compared to bottom of list. For pivot selection, our goal is not to select the most relevant documents, but to select at least one document from all relevance classes for a particular topic (If we only sample documents from top of the list, then there is a high chance that our sample has no document from some relevance class whose documents had very few topic keywords and were ranked very low in the BM25 rank list). Selecting every fifth document from BM25 rank list decreases chances of missing an entire relevance class.

These sampled documents were shown to an assessor who examined all these sampled documents. The assessor selected all the documents form this sample that were even slightly relevant to the topic. These selected documents were then re-examined for selection of small number of pivots. These pivots are selected in a manner such that each pivot is from a different relevance class, for a given topic.

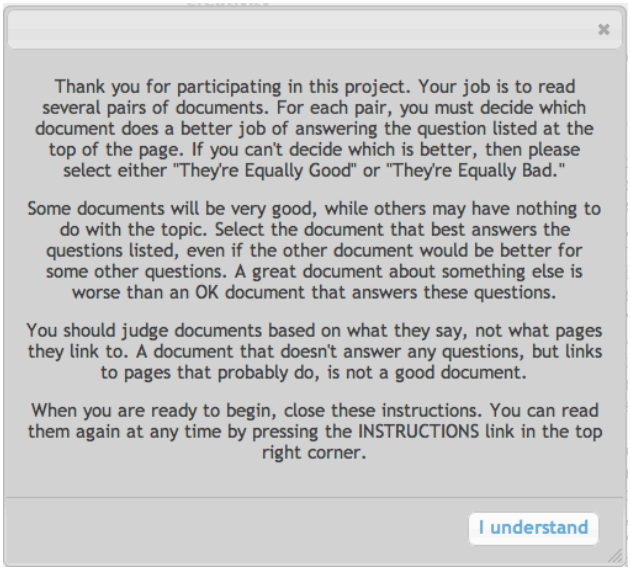

Figure 1: Instructions for preference judgements for assessors

2.2 Linear Search for Each Document

Once pivot documents are selected for a topic, they are sorted based on their relevance to the topic. Each of the remaining documents is compared with pivot documents to find its "correct spot". We start by comparing each document to the least relevant pivot document. If the document is assessed as more relevant than the pivot document then it is compared to the next least relevant pivot document. If the document is assessed as less relevant than a pivot document, we stop comparisons for that document. One would think binary search is the most effective search among sorted pivots, but in our case a weighted binary search essentially reduces a linear search: our prior assumption about relevance of Information Retrieval text collections is that the ratio of non-relevant documents, then little-relevance, and so on, is high enough compared to the next grade, such that the distribution over grades dictates a linear search as the most efficient. All the non-relevant documents will be compared only once to the least relevant pivot. Since the number of pivots is fixed, the overall assessment takes $O(n)$ comparisons.

2.3 Grades from Preference

After comparing all documents with pivot documents, we sorted the documents using preference judgements. The documents are partitioned into $n+1$ relevance classes for n pivots. Each topic-document pair needs to be judged on a six-point scale for this Crowdsourcing Track. Each of the pivots were manually examined by an assessor for assigning a relevance grade.

3 Experiments

In this section we describe our experimental design for the collection of preference judgements.

3.1 Interface Design

The interface we created to collect preference judgements had the following design. After accepting the assignment, assessor was shown the instructions in Figure 1. In these instructions, we explained that documents should be preferred strictly based on whether they provide information about the query, description,

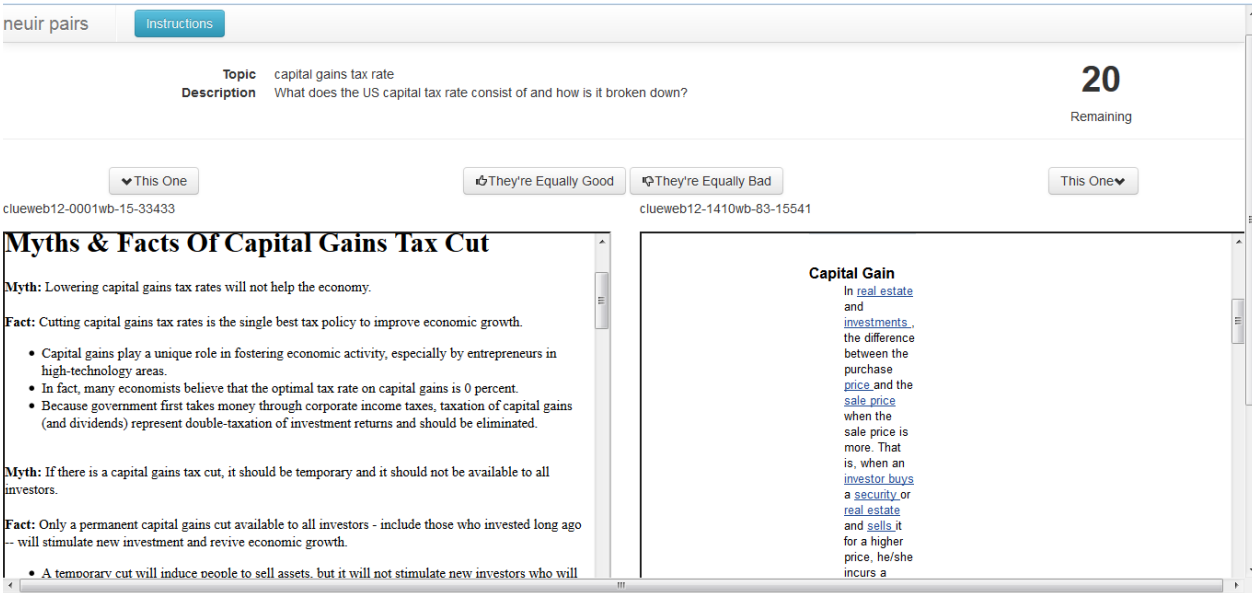

Figure 2: Preference pair selection interface with topic keywords and description

and narrative for a particular topic: that a well-written discussion of a related topic should not be preferred to a poorly-written document which is exactly on topic.

After dismissing the instructions, the assessor is shown the interface presented in Figure 2. The "Title" field of a TREC topic is displayed on top, along with its description and narrative. This information describes in detail what constitutes a relevant document for this query. Below the query information is a series of buttons, which allow assessor to record their preferences. Two documents are displayed side-by-side below the buttons. The leftmost and rightmost buttons are labeled "This One," with an arrow pointing to the left or right document, respectively. These buttons allow assessors to choose a winning document. Between these buttons are two buttons for recording ties, labeled "They're Equally Good" and "They're Equally Bad".

3.2 Data

In the basic task of Crowdsourcing track, there were a total of 3,470 documents from ClueWeb12 dataset. ClueWeb12 dataset consists of 733,019,372 English web pages, collected between February 10, 2012 and May 10, 2012 using web crawlers. 10 topics were provided by NIST assessors for this task. Participants of the Crowdsourcing Track were required to simulate the role of NIST assessors for the 10 topics.

4 Results

Four groups submitted a total of 11 runs to the Crowdsourcing track this year. Graded judgments of all runs were evaluated against the NIST qrels, using the GAP (Robertson et al [4]) measure. NIST did not judge as much of the pool as anticipated, and thus there are fewer documents judged than expected. A ranking of TREC 2013 Web track adhoc runs (34 adhoc runs) was induced using each of the submitted runs (crowd.qrels). Each crowd.qrel.ranking was compared to nist.qrel.ranking using AP-Correlation (Yilmaz et al [5]) and RMSE (root mean squared error). All groups were evaluated on the basic subset of 10 topics only. The evaluation results of our run submitted to TREC are given in Table 1. Table 1 compares our submitted run (NEUPivot1) to the median scores of the systems that participated in the TREC 2013 crowdsourcing task. Our methodology exhibits higher than the median GAP score for all but one topic (202) on submitted

Topic	# Docs	GAP		τ_{AP} APCorr		RMSE	
		NEUPivot1	Median of TREC Runs	NEUPivot1	Median of TREC Runs	NEUPivot1	Median of TREC Runs
202	231	0.007	0.035714	-0.045	-0.01868796	0.157	0.3667390
214	305	0.797	0.629760	0.325	0.16270939	0.210	0.2100571
216	387	0.730	0.588345	0.449	0.14264343	0.090	0.2284822
221	368	0.708	0.535430	0.430	0.08049942	0.183	0.1972541
227	246	0.606	0.113211	0.642	-0.23982220	0.107	0.4655352
230	172	0.678	0.272572	0.569	-0.21109738	0.169	0.3311034
234	298	0.814	0.602812	0.231	0.12199660	0.320	0.2912807
243	342	0.598	0.254524	0.732	0.38503298	0.158	0.3668166
246	202	0.428	0.373818	0.048	0.22663554	0.210	0.3438483
250	207	0.474	0.093525	0.366	-0.18280615	0.107	0.2342819
All	2758	0.584	NA	0.461	NA	0.085	NA

Table 1: This table shows per-topic statistics and overall averages for the run NEUPivot1 and median score for 11 runs submitted to crowdsourcing track. The metrics GAP, ERR@20, AP-correlation and RMSE are listed for each topic. Note that for row all, (i) GAP is the mean gap over all 10 topics, (ii) APCorr and RMSE depend on the ranking of runs induced by the mean ERR20 for all the 10 topics.

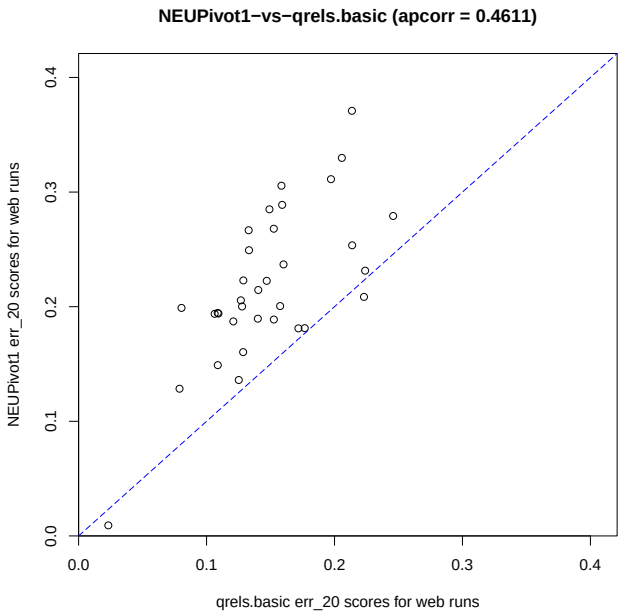

Figure 3: NEUPivot1-basic-ERR@20 vs qrels.basic-ERR@20. qrels.basic is the TREC 2013 web track qrels reduced to topics 202, 214, 216, 221, 227, 230, 234, 243, 246, and 250

runs. The median GAP score for topic 202 is 0.036. In our opinion, the reason for this poor median GAP for topic 202 is missing images from ClueWeb12 documents. The topic 202 ("USS Carl Vinson") is navigational topic and participants were asked to find homepage for this topic. The homepage had large missing image in the ClueWeb12 documents so no assessor could identify that web page as home page for this topic.

5 Conclusion and Future Work

In this paper we have described our work based on preference judgments for obtaining high quality multiple graded relevance judgements. We have used modified version of QuickSort for sorting documents using linear number of preference judgments. The choice of good pivots is essential, to good performance of our sorting algorithm so pivots should be selected by expert assessors. Once we select good pivots, the task of placing documents in right position among pivots is not complex and can be assigned to crowd workers. In future, we plan to use this methodology with crowd workers.

References

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.

[2] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there. In *ECIR*, pages 16–27, 2008.

[3] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, Dec. 2003.

[4] S. E. Robertson, E. Kanoulas, and E. Yilmaz. Extending average precision to graded relevance judgments. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 603–610, New York, NY, USA, 2010. ACM.

[5] E. Yilmaz, J. A. Aslam, and S. Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 587–594, New York, NY, USA, 2008. ACM.