Hybrid Crowd-Machine Methods as Alternatives to Pooling and Expert Judgments

Christopher G. Harris[1] and Padmini Srinivasan[2]

[1] Department of Computer Science, SUNY Oswego, Oswego, NY 13126
christopher.harris@oswego.edu
[2] Department of Computer Science, University of Iowa, Iowa City, IA 52242
padmini-srinivasan@uiowa.edu

Abstract. Pooling is a document sampling strategy commonly used to collect relevance judgments when multiple retrieval/ranking algorithms are involved. A fixed number of top ranking documents from each algorithm form a pool. Traditionally, expensive experts judge the pool of documents for relevance. We propose and test two hybrid algorithms as alternatives that reduce assessment costs and are effective. The machine part selects documents to judge from the full set of retrieved documents. The human part uses inexpensive crowd workers to make judgments. We present a clustered and a non-clustered approach for document selection and two experiments testing our algorithms. The first is designed to be statistically robust, controlling for variations across crowd workers, collections, domains and topics. The second is designed along natural lines and investigates more topics. Our results demonstrate high quality can be achieved and at low cost. Moreover, this can be done by judging far fewer documents than with pooling. Precision, recall, F-scores and LAM are very strong, indicating that our algorithms with crowd sourcing offer viable alternatives to collecting judgments via pooling with expert assessments.

Keywords: Pooling, Crowdsourcing, Retrieval evaluation, Relevance judgment.

1 Introduction

Collecting relevance judgments is crucial for Information Retrieval research. Batch mode algorithm evaluation requires that we know the correct answers, i.e., which documents in a collection are relevant to a query. A standard approach to obtain relevance judgments when multiple algorithms are involved is by a process called 'pooling' [24], first introduced in the 1992 TREC initiative [7]. In pooling a fixed number *N* of documents are taken from each algorithm's output to form a pool. Pooling combined with TREC assessor judgments has generated many valuable collections of relevance judgments. Recognizing the expense of assessor judgments the TREC Crowdsourcing track (since 2011) spearheaded research on alternative mechanisms for collecting relevance judgments [12]. Continuing in the same spirit we propose two hybrid crowd/machine approaches for collecting judgments. In each an algorithm (clustered or non-clustered) selects documents for judgment (in contrast to pooling) and crowd workers provide judgments.

A. Jaafar et al. (Eds.): AIRS 2014, LNCS 8870, pp. 60–72, 2014.

Our first goal is to test the effectiveness of our two approaches (clustered versus non-clustered) for collecting relevance judgments in a *statistically robust* manner addressing domain differences (News versus Biomedicine) and controlling for worker variations. Previous crowdsourcing work has not addressed the effect of domain. TREC Crowdsourcing has focused on news and web data. Perhaps it is easier for a crowd worker to judge relevance for a query related to news and general web information than for a query related to more technical chemical patents or in biomedicine. Work quality may differ due to differences in background, expertise, level of commitment to the task etc. We know that multiple judgments are needed [14], but the typical approach is to obtain just 3 judgments on a document – query pair. We address this goal with an ANOVA design experiment. The ANOVA power calculation for medium power specifies a minimum of 24 workers to judge for each algorithm–collection combination. Moreover, each worker must judge all topics in a given combination. For practical reasons (e.g., to avoid worker fatigue), we limit this experiment to three topics. *Note that this is still sufficient to make statistically valid conclusions*. As required by ANOVA design we ensure that the two domains are comparable in topics (prevalence of relevant information) and documents (e.g., word complexity, number of sentences). We study both main and interaction effects.

Our second goal is to conduct another experiment (with News) using typical settings seen in crowdsourcing papers. Also we use full documents and more topics (20). We use a majority vote from three judgments for each decision. We compare our approaches with each other and with pooling in efficiency, effectiveness and cost.

As highlights of our results: in our first experiment we find, for example, that the clustered approach achieves significantly better recall while the non-clustered one achieves better precision, F-score and LAM. In the second experiment, the non-clustered approach achieves F-score of 0.73 while the clustered approach is just short of 0.8. LAM (Logistic Average Misclassification rate) scores are around 0.05. Though not strictly comparable these results appear competitive with the best TREC 2012 results. When compared to pooling, our methods are more efficient and cost far less. Thus despite wide concerns about quality of work done by crowdsourcing (e.g., [11], [28]), our methods provide results of high quality at much lower cost. Moreover, our methods are scalable and easy to extend to other relevance assessment campaigns.

2 Related Research

In IR experiments research groups have obtained relevance judgments from many types of sources including students, librarians, subject specialists and TREC assessors. Using individuals external to the research group almost always requires payment usually in the form of money. Besides cost there is the impracticality of getting judgments for every document retrieved. Thus we see wide usage of sampling strategies such as pooling. In recent years crowdsourcing is being tested as a source for relevance judgments (and other kinds of decisions). Utilizing non-experts through crowdsourcing platforms has proven to be cost-effective (e.g.,[2],[11],[14]). However, two key challenges are the variability of assessors' judging abilities and the aggregation of noisy labels into a single consensus label. Even within the TREC/NIST framework, considerable variability across trained assessors still exists (e.g., [4],

[26]); this is shown to increase when non-experts are used [21]. To address noise, typically a majority vote is obtained across several judgments [18].

Several others have recognized the challenges of relevance assessment that are not met by pooling, and have introduced some new approaches. Soboroff et. al. examined some methods that could approximate pooling, including using a shallower pooling depth and identification of the duplicate documents normally removed by pooling, improving upon standard pooling methods [20]. We build on their methods, using the amount of duplication as an important input into our ranking algorithm. Sanderson and Joho sidestep pooling by using existing TREC judgments as inputs to a relevance feedback algorithm [23], which provides valuable information to identify which documents to investigate further. Carterette et. al. algorithmically determined the smallest test collection size to judge using a variation of MAP [3]. Their method uses ranked lists to choose documents for judgment, which is one aspect we use in our own methods. Yilmaz et. al. used stratified sampling with pooling at different k-depths in [30], providing good results, but the focus in their study was primarily on improving existing IR evaluation measures.

In 2011, the TREC Crowdsourcing track was begun to examine the crowd's ability to provide quality relevance assessments while addressing the above challenges [12]; this continued with the TRAT sub-track in 2012 [22] and 2013. A number of approaches using the crowd were examined. The top-performing BUPT-Wildcat [29] used an elaborate multi-stage prescreening process for crowd workers and an E-M Gaussian process to determine a consensus label among these workers. The complexity in their method, requiring the development, evaluation, and testing of prescreening tools suggests that it might have difficulty in scaling. Likewise, in 2012, Skierarchy, a top-performer requires an impressive but complex hierarchy of experts and crowd workers [16]; this approach too might have problems with increasing scale due to the requirement of more subject-matter experts, which are often in limited supply. In contrast we use a hybrid machine-human approach involving fusion of ranks across submitted lists of retrieved documents, optional text-feature based clustering, document batching and selection, and criteria to stop the relevance judgment process. The power of our approach is in its simplicity and in its effectiveness. The approach is an extension of our earlier work in TREC [8]; significant differences include our use of full submissions versus pooling and more refined document selection strategies. In addition, we present more extensive experiments in multiple domains compared to the TREC TRAT effort.

3 Algorithms

We propose two algorithms (clustered and non-clustered) to select documents for relevance judgments. Consider a typical TREC ad hoc retrieval task with a set of M runs submitted by competing systems for a topic T. Each run is an ordering of documents ranked by system estimated relevance to T. Both algorithms start with the union (U) of *all* documents in these M runs. This contrasts with pooling which only takes a limited number (e.g., 100) of the top ranked documents. Our advantage is that we need not use an artificial cutoff. We then calculate a score CW for each document

in *U* with respect to topic *T*. Documents in *U* are ranked by their *CW* scores. $CS_T(d)$ is a *simple count* of the number of submitted runs that retrieved document *d*. $CB_T(d)$ is the *Borda count* which takes into account document rank [6].

$$CW_T(d) = \propto CS_T(d) + (1-\propto)CB_T(d)$$

$$CB_T(d) = \sum_{i=1}^{M} N - r_{id}$$

Here r_{id} is d's rank in run *i*. *N* is a fixed number equal to or larger than the maximum number of documents that may be submitted in a run. For runs not retrieving *d*, rank is equal to *N*. The TREC campaigns generally allow a maximum of 1000 submitted documents per run per topic. In training runs using 10 separate topics we tested α from 0 to 1 in increments of 0.05. For each α we assessed the resulting ranking of *U* using RS_α.

$$RS_\alpha = \frac{\sum_{n=1}^{|U|} \mathrm{rel_T(n)} * \mathrm{CW_T(n)_\alpha}}{\sum_{n=1}^{|U|} \mathrm{rel_T(n)}}$$

Here $rel_T(n)$ is 1 if document *n* is relevant to topic *T* and 0 otherwise. RS_α is highest when all relevant documents occupy top ranks. Averaging RS_α across all ten training topics for each dataset the best results were at $\alpha = 0.8$. We therefore use this value in our experiments. At this point our two algorithms deviate as described next.

3.1 Algorithm 1 – Non-clustered Approach

Documents in *U* ranked by their CW_T score are partitioned into batches of equal size. Starting with the top ranking batch, crowd workers judge documents till a batch with no relevant documents is reached. Judgment then stops with all remaining documents marked as "not relevant". Again using our training set we determined that the best batch size is 20. For this training step and for later training steps we run the approach using the gold standard data to simulate crowd relevance assessment. This best-case scenario places a ceiling in the effectiveness of our algorithms.

3.2 Algorithm 2 – Clustered Approach

The motive is to involve text features in the document selection process. The well-known cluster hypothesis [13] indicates textually similar documents are more likely to be co-relevant (or co-non-relevant) to a given topic than dissimilar documents. We first cluster documents in *U* using k-means representing each document with a word-based unigram feature vector. We then rank documents in each cluster by *CW* and partition them into batches. Documents of the top-ranking batch of each cluster are automatically selected for judgment followed by the next ranked batch. If a batch

yields zero relevant documents then the remaining batches of its cluster are marked non relevant. Thus at least k batches are judged, one for each cluster. This is to accommodate documents retrieved by possibly distinct retrieval algorithms.

We establish k for k-means using standard approaches (e.g., [4], [16]). We evaluate k=5 through 20 in increments of 3 and calculate the variance in the number of relevant documents appearing in each cluster. Greater variance implies an increasing tendency for relevant documents to concentrate in fewer clusters, which is desirable. We then explore values of k one unit away on either side of the best value. As a result, we set k=11 for both collections in our experiments. Batch size remains 20 documents as in algorithm 1.

4 Datasets And Topics

4.1 Datasets and Documents

General Domain: News Dataset. This is the TREC-8 ad hoc task (TREC disks 4 and 5, less the Congressional Record). We use the document set corresponding to the TREC-8 ad hoc task [27] and topics 401-443. *Specialized Domain*: OHSUMED Dataset. This is the TREC-9 filtering track dataset, topics 1 – 43 [19].

Table 1. Domain characteristics. M: mean, sd: standard deviation

Text Statistic	OHSUMED		News	
	M	SD	M	SD
No. of sentences	6.97	1.04	6.99	0.04
No. of words	149.24	18.48	140.63	21.07
No. of complex words	32.04	3.96	24.90	10.63
% of complex words (>= 3 syllables)	22.19%	0.02	17.71%	0.03
Average words/sentence	21.13	1.39	20.12	0.48
Average syllables/word	1.76	0.09	1.68	0.11

OHSUMED has only titles, abstracts and metadata whereas the News documents are full text. Since length differences can bias results we use only the headline and first 7 sentences for News; for OHSUMED, we use the title and the abstract. These reduced News documents are shown to crowd workers and they are used when clustering (section 3.2). Collection features (after this step is completed) are provided in Table 1. Rows 3 to 6 of Table 1 illustrate differences intrinsic to the domains.

4.2 Topics, Runs, Gold Standard Data

To prevent topic differences from biasing results, we identified three comparable topics (in prevalence of relevant documents) from each collection. Intuitively prevalence, the percentage of submitted documents that is relevant, may indicate topic difficulty. Prevalence for News topics ranges from (0.03, 2.46) and for OHSUMED

(0.04, 4.62). We ranked the OHSUMED topics that fall in the overlapping region (0.04, 2.46) and divided them into 3 groups of roughly equal size. Randomly selecting one OHSUMED topic from each group we then identified 3 News topics that most closely matched in prevalence (see Table 2). For each selected topic all documents in submitted runs of past TREC participants are collected.

Table 2. Characteristics of selected OHSUMED topics

OHSUMED				News			
Topic ID	**No of Submitted Docs**	**No of Relevant Docs**	**Percentage Relevant**	**Topic ID**	**No of Submitted Docs**	**No of Relevant Docs**	**Percentage Relevant**
12	5291	7	0.132%	403	15636	21	0.134%
1	5784	44	0.761%	421	11090	83	0.748%
13	5841	77	1.318%	436	13940	180	1.291%

For News the TREC-8 ad hoc task obtained binary relevance assessments using pooling and TREC experts. For OHSUMED the TREC-9 filtering task provides assessments in one of three relevance states ("not relevant", "partially relevant", or "definitely relevant"). Following [2] we group "partially relevant" and "definitely relevant" documents as "relevant". It should be noted that OHSUMED relevance judgments were obtained in earlier studies [9, 10] and not via TREC expert assessors and pooling. We simulate pooling (selecting top 100 documents) with OHSUMED.

4.3 Participants

Crowd participants were from Amazon Mechanical Turk (MTurk). Participants were assigned randomly to either the non-clustered or clustered algorithm and were only permitted to participate once (as tracked by IP address and MTurk ID). They were compensated $0.20/batch of 20 documents assessed. For each algorithm–collection combination we used 24 participants; each participant evaluated all 3 topics for that collection. Participants not completing the full assessment of 3 topics had their assessments removed and the task given to other crowd participants. Judgments were collected via a web interface.

5 Results

Table 3 provides the means and standard deviations across our four metrics: recall, precision, F-score and LAM. Statistically significant differences are marked by * ($p < 0.05$) and ** ($p < 0.002$). Examining main effects we find that the non-clustered algorithm gives significantly superior precision, F-score and LAM compared to the clustered algorithm. However, the clustered algorithm is significantly superior in recall. In main effects we also find that the biomedical domain provides significantly superior results on all 4 measures compared to News. We discuss these surprising results later. Post-hoc analyses were conducted on all statistically significant pairings

between algorithm and domain for each measure. All pairs tested were also found to be statistically significant ($p < .05$) using Fisher's LSD post-hoc test. These results reject all main effects null hypotheses claiming no difference between algorithms or between domains on these measures. The two-way, algorithm × domain interaction results are similar in precision and F-score; combinations involving the non-clustered algorithm and either domain are significantly superior to combinations involving the clustered approach. For LAM this pattern is seen only for OHSUMED. For recall, combinations involving the clustered approach provide significantly superior results. We reject all but one of the two-way interaction null hypotheses.

Table 2. Means and standard deviations for the measures (n = 96)

Condition	Precision		Recall		F-score		LAM		N
	M	SD	M	SD	M	SD	M	SD	
Algorithm type									
Non-clustered	0.610*	0.089	0.543	0.102	0.551*	0.101	0.043*	0.007	48
Clustered	0.323	0.040	0.740*	0.055	0.430	0.044	0.062	0.004	48
Domain type									
News	0.451	0.137	0.584	0.137	0.449	0050	0.056	0.007	48
OHSUMED	0.482*	0.180	0.699*	0.087	0.532*	0.117	0.050*	0.013	48
Algorithm × Domain									
Non-cluster, News	0.572**	0.078	0.452	0.029	0.467*	0.049	0.049	0.005	24
Non-cluster, OHSUMED	0.648**	0.084	0.635	0.054	0.635**	0.062	0.037*	0.004	24
Cluster, News	0.331	0.043	0.717**	0.035	0.431	0.045	0.062	0.003	24
Cluster, OHSUMED	0.315	0.036	0.764**	0.062	0.430	0.044	0.062	0.005	24

6 Analysis

The relative performance of the two algorithms is consistent across collections. The non-clustered algorithm provides better precision, F-score, and LAM while the clustered algorithm provides better recall. The former relies on the weighted score *(CW)* while the latter also exploits textual features via document clustering. The improvement in recall is consistent with the cluster hypothesis. However, this is at the expense of precision; non-relevant documents are also attracted towards relevant documents through similarity. In the combined F-score, the simpler non-clustered algorithm wins over clustering.

Another aspect that might have caused lower precision for the clustered approach is that at least 1 batch (the top ranking one) is judged from each cluster (see Section 3.2). Given 11 clusters/topic and 20 documents/batch we have a minimum of 220 judgments. Retrospectively we feel precision might improve if we are more selective in clusters to judge. We chose to judge at least 1 batch/cluster to capture different subthemes of relevance and because unique retrieval strategies may retrieve distinctive relevant documents. We address this in Section 7.1.

Surprisingly the performance for OHSUMED is better than for News on all measures. We expected familiarity with the domain to favour general news stories and not biomedicine. A possible explanation is that with News we had to limit our document to the first 7 sentences of text in order to possible length-based bias across domains. It may be that the text necessary for relevance judgments appears outside of these initial 7 sentences for News. OHSUMED have focused abstracts which may have enabled more accurate judgments. We address this in the next experiment.

Finally though not strictly comparable, our best LAM score for News (0.049) is better than the best scores obtained in TREC (also for News). Importantly, this is achieved while maintaining reasonable scores for the other three measures.

6.1 Comparing the Algorithms and Pooling: Efficiency and Effectiveness

We compare the algorithms with each other and with pooling in efficiency balanced against effectiveness. Pooling starts with the union of top N (typically set to 100 in TREC) ranked documents from each run. In contrast our algorithms start with the union of all retrieved documents submitted by all runs (this is typically set to 1000 in TREC runs). Thus we run the risk of judging a large number of documents; most are likely to be non-relevant.

Pooling for the 3 OHSUMED topics with N=100 would have judged 50% to 62% of the submitted documents; for News 6.7 to 16%. In contrast the clustered approach judged 8 to 14% for OHSUMED and 5 to 9.4% for News. The non-clustered approach judged 1% to 5% for OHSUMED and 0.6% to 3% for News. The savings in our methods are clear. Overall, our percentages are quite reassuring given that we start with the full set of submitted documents. The clustered approach is less efficient than the non-clustered again because at least 1 batch is judged from each of the 11 clusters (220 documents). The non-clustered approach has no such minimum.

Efficiency is only interesting if the strategies are effective at finding the relevant documents. Although the means presented earlier indicate effectiveness, we can look at the results in more detail. On average across the topics, the non-clustered algorithm evaluated 3.1% of the submitted OHSUMED documents, to find 68.0% of the relevant documents; the clustered algorithm evaluated 11.0% finding 77.5%. For News the non-clustered algorithm only evaluated 1.8% of the collection finding 46.8% of the relevant documents. The percentages were low for topics 421 and 436. With clustering 7.6% of the News submissions were evaluated, with 73.2% of the relevant documents found. While these are good results, we will strive to improve effectiveness of our strategies in future work.

Particularly noteworthy is our algorithm's success even when there are only a few relevant documents – for example, the non-clustered algorithm only evaluated 0.6% of the 15,636 documents retrieved for topic 403, but was able to find nearly all of the 21 relevant documents.

6.2 Comparing Methods on Cost

Relevance judgments costs are important given budgetary constraints. The 2007 TREC Legal track overview document is one in which TREC relevance assessment costs are indicated. Also they note that human assessors evaluate on average 20 documents per hour. Their relevance assessment cost was estimated at $150 an hour, or $7.50 per document [25]. A total of 9442 judgments would have been made with pooling setting $N = 100$ for OHSUMED and 4758 judgments were made for News. This is assuming a single judgment for a query document pair. This gives a total of $70,815 and $35,685 for OHSUMED and News respectively. Admittedly assessment for TREC Legal would have been amongst the costliest. However, even if we were to reduce the TREC cost drastically to $1/judgment, the TREC pooling process for the 3 topics in OHSUMED and News would be $9,442 and $4,758 respectively. In comparison, our cost was $0.22/batch of 20 documents, including Amazon Mechanical Turk overhead fees; this is slightly more than $0.01/document. The cost for all 24 participants to evaluate the same 3 News topics was $792.00 for the clustered algorithm and $179.52 for the non-clustered algorithm. For OHSUMED these costs are $496 and $143 respectively. Using only 3 crowd workers and the majority decision, as discussed in [1, 2], we can reduce these costs further by 87.5%.

6.3 Detecting Potentially Relevant Documents

Another aspect of our approach is that since we start with all submitted documents there is the possibility of discovering relevant documents missed by the pooling process. Limiting our attention to News and only those document – query pairs that were judged by at least 12 participants we find that there are 9 potentially relevant documents that were not included in the TREC pool and so were not judged and assumed non relevant. There were also four documents that our participants thought were relevant that were declared non relevant by TREC assessors and 10 documents in the reverse direction. These numbers may appear to be minor and yet they could in a different context make an appreciable difference.

7 Experiment 2: More Topics, Natural Design

We present results from a second experiment with the News dataset. We use only 3 judges for a document–query pair, a typical number in crowd sourcing tasks relying on a majority vote. We selected 17 additional topics randomly from the News dataset and added these to our 3 topics from experiment 1 (topics 403, 421, and 436). Also we use the full text of News items rather than just the first 7 sentences plus headline. This full text is used during clustering and is also shown to the participant judging the document. Similar to the earlier experiment, each participant is expected to evaluate 3 randomly assigned (out of 20) topics. If this was not done for any reason, then a substitute participant was solicited.

The mean precision, recall, F-score, and LAM scores are strong: 0.8826, 0.6282, 0.7270, and 0.0499 respectively for the non-clustered algorithm, and 0.8174, 0.7645,

0.7861 and 0.0524 respectively for the clustered algorithm. Wilcoxon Signed-Rank tests found that the non-clustered approach is better than the clustered approach in precision and in LAM (both at $p< 0.0005$). The clustered algorithm is better than the non-clustered one in recall and F-score (both a $p < 0.0005$). With the exception of F-score, these results are consistent with our findings in the first experiment, where we had used a larger number of participants for statistical robustness but fewer topics. The F-score results indicate that the difference in recall between the two algorithms more than adequately makes up for the difference in precision in this second experiment. In general, scores obtained in experiment 2 (20 topics) are higher than in experiment 1 (3 topics) with some differences being quite remarkable (e.g., in precision). Focusing only on the 3 News topics common to both experiments there are 19 of 24 measurements (2 algorithms × 4 measures × 3 topics) where experiment 2 gives better results and only 5 where experiment 1 is better. The key design difference is that experiment 2 uses the full text of the news items as opposed to just the first 7 sentences. It appears that this aspect makes a crucial difference.

7.1 Reducing the Number of Clusters Evaluated

In our clustered algorithm, at least one batch from each of 11 clusters is judged. This may be why this algorithm has lower precision compared to the non-clustered algorithm. If we can be more selective about clusters we might improve the performance overall. We explore a strategy based on the *CW* score. For each topic we rank the 11 clusters by their mean *CW* score. We remove clusters, one at a time, lowest to highest mean evaluating performance each time. Results are in Fig. 1.

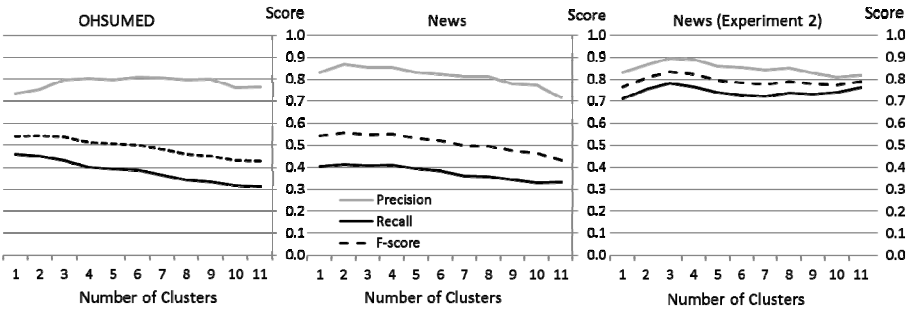

Fig. 1. Performance as the number of clusters increases for OHSUMED (left), news (center) in experiment 1 and for news in experiment 2 (right)

When all 11 clusters are evaluated, our F-score (middle line) is at its lowest. For News the best F-score is obtained using only the top 2 clusters while for OHSUMED it is with the top 3 clusters. These numbers are considerably smaller than 11. Experiment 2 yields similar results; the optimal number of clusters is approximately 3. These emphasize that it would be useful to select clusters to judge. In future research we will also explore functions of mean *CW* score as a cutoff.

8 Conclusion

We presented alternatives to pooling that use an algorithm to select documents for judgment and crowd workers to make the judgments. Our best algorithm is able to locate a majority of the relevant documents in two types of collections at a fraction of the cost of pooling. In both experiments we obtain LAM scores for News that are competitive with the best 2012 TREC Crowdsourcing results [22] (though the experiments are not strictly comparable). We find that contrary to some predictions [15] and our own expectations, results in OHSUMED (e.g., LAM is 0.037), a more challenging domain, are also strong. Overall we have demonstrated that our hybrid approach using rank fusion, optional clustering, document batching with intuitive stopping criteria, though simple in design is both effective and cost efficient. This backs up the earlier findings by Soboroff et. al. [23], Carterette et. al. [3], and Sanderson and Joho [20]; it builds on aspects of their methods with a clustering technique that is simple yet effective. We have presented results using statistically robust design considering carefully potential variations across crowd workers.

There are a number of ways in which we can improve our approach and extend this study. First we will further explore strategies for being selective in the clusters chosen for judgment. Second we would like to know if the relative ranking of the participating systems changes if we were to use just the judgments provided by our methods. This will parallel efforts such as [23]. Third we would like to conduct topic level analysis of our data. Some topics are likely more challenging for crowd workers than others. A follow up goal would be to see if we can predict which topics are likely to be challenging. A fourth direction is to analyze the crowd judgments to see the extent to which there is consensus. Our dataset is rich in that we have each query–document pair (as in experiment 1) judged by up to 48 individuals (24 workers/algorithm). This will offer insights into variations across workers. Last, we plan to look at stratified sampling techniques as discussed by Yilmaz et. al. in [30], and how system rankings coordinate with the full pool.

References

1. Alonso, O., Mizzaro, S.: Can We Get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In: Proc. SIGIR 2009 Workshop on the Future of IR Evaluation, pp. 15–16 (2009)
2. Alonso, O., Mizzaro, S.: Using Crowdsourcing for TREC Relevance Assessment. Information Processing & Management 48(6), 1053–1066 (2012)
3. Carterette, B., Allan, J., Sitaraman, R.: Minimal Test Collections for Retrieval Evaluation. In: SIGIR 2006, pp. 268–275. ACM (2006)
4. Carterette, B., Soboroff, I.: The Effect of Assessor Error on IR System Evaluation. In: SIGIR 2010, pp. 539–546. ACM (2010)
5. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis, vol. 3. Wiley, New York (1973)
6. Emerson, P.: The Original BordaCount and Partial Voting. Social Choice and Welfare 40(2), 353–358 (2013)

7. Harman, D.K.: The First Text Retrieval Conference (TREC-1), Rockville, MD, USA, November 4-6 (1992); Information Processing & Management 29(4), 411–414 (1993)
8. Harris, C., Srinivasan, P.: Using Hybrid Methods for Relevance Assessment. In: TREC Crowd 2012, TREC Notebook Paper (2012)
9. Hersh, W., Buckley, C., Leone, T.J., Hickam, D.: OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In: SIGIR 1994, pp. 192–201. ACM (1994)
10. Hersh, W., Hickam, D.: Use of a Multi-Application Computer Workstation in a Clinical Setting. Bulletin of the Medical Library Association 82(4), 382 (1994)
11. Kazai, G., Milic-Frayling, N.: On the Evaluation of the Quality of Relevance Assessments Collected through Crowdsourcing. In: SIGIR 2009 Workshop on the Future of IR Evaluation, p. 21 (2009)
12. Lease, M., Kazai, G.: Overview of the TREC 2011 Crowdsourcing Track. TREC Notebook Paper (2011)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
14. McCreadie, R., Macdonald, C., Ounis, I.: Identifying Top News using Crowdsourcing. Information Retrieval, 1–31 (2013)
15. Meilă, M., Heckerman, D.: An Experimental Comparison of Several Clustering and Initialization Methods. In: Proc. of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 386–395. Morgan Kaufmann (1998)
16. Nallapati, R., Peerreddy, S., Singhal, P.: Skierarchy: Extending the Power of Crowdsourcing using a Hierarchy of Domain Experts, Crowd and Machine Learning. TREC Notebook Paper (2012)
17. Qi, H., Yang, M., He, X., Li, S.: Re-examination on Lam% in Spam Filtering. In: SIGIR 2010, pp. 757–758. ACM (2010)
18. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from Crowds. Journal of Machine Learning Research 99, 1297–1322 (2010)
19. Robertson, S.E., Hull, D.A.: The TREC-9 Filtering Track Final Report. In: Online Proc. of TREC (2000)
20. Sanderson, M., Joho, H.: Forming Test Collections with no System Pooling. In: SIGIR 2004, pp. 33–40. ACM (2004)
21. Smucker, M.D., Jethani, C.P.: Human Performance and Retrieval Precision Revisited. In: SIGIR 2010, pp. 595–602. ACM (2010)
22. Smucker, M.D., Kazai, G., Lease, M.: Overview of the TREC 2012 Crowdsourcing Track. TREC Notebook Paper (2012)
23. Soboroff, I., Nicholas, C., Cahan, P.: Ranking Retrieval Systems without Relevance Judgments. In: SIGIR 2001, pp. 66–73. ACM (2001)
24. Sparck Jones, K., van Rijsbergen, C.: Report on the Need for and Provision of an "Ideal" Information Retrieval Test Collection, British Library Research and Development Report 5266, Computer Laboratory, Univ. of Cambridge (1975)
25. Tomlinson, S., Oard, D.W., Baron, J.R., Thompson, P.: Overview of the TREC 2007 Legal Track. In: Online Proceedings of TREC (2007)
26. Voorhees, E.M.: Variations in relevance judgments and the measurement of retrieval effectiveness. Information Processing & Management 36(5), 697–716 (2000)
27. Voorhees, E.M., Harman, D.: Overview of the Fifth Text REtrieval Conference (TREC-5). TREC (97), 1–28 (1996)

28. Vuurens, J., de Vries, A.P., Eickhoff, C.: How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy. In: Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval, pp. 21–26 (2011)
29. Xia, T., Zhang, C., Li, T., Xie, J.: BUPT_WILDCAT at TREC Crowdsourcing Track. TREC Notebook Paper (2012)
30. Yilmaz, E., Kanoulas, E., Aslam, J.A.: A Simple and Efficient Sampling Method for Estimating AP and NDCG. In: SIGIR 2008, pp. 603–610. ACM (2008)