Crowdsourcing with a Crowd of One and Other TREC 2011 Crowdsourcing and Web Track Experiments

Mark D. Smucker

Department of Management Sciences
University of Waterloo

Abstract

Our submissions to the Crowdsourcing and Web tracks emphasized simplicity in either method, construction or both. Based on preliminary results, we found that if the number of relevance assessments is low, researchers may be better off "self-sourcing" the assessments, i.e. performing the relevance assessments themselves, rather than crowdsourcing the work. For the Crowdsourcing consensus task, we found that a simple weighted majority vote with iteratively refined workers' quality as measured by d' (d-prime) performed slightly above the median on the gold test set. Finally, we submitted easy to construct runs to the Web ad-hoc track which had P@10 scores above the median on a majority of the topics.

1 Introduction

This year we participated in both the Crowdsourcing and Web tracks. For the Crowdsourcing track, we participated in both the judging task (task 1) and the consensus task (task 2). For the Web track, we only participated in the ad-hoc retrieval task, although our runs were also evaluated for the diversity task. As our efforts in these tracks were unrelated, we discuss each track separately and begin first with the Crowdsourcing track.

2 Crowdsourcing Track

The Crowdsourcing track had two sub-tasks. Task 1 was called the "assessment" task and involved the study of relevance assessing. Task 2 was called the "consensus" task and involved the study of methods for determining the relevance assessment of a document given the relevance judgments of one or more assessors.

2.1 Task 1: Relevance Assessing

In this task, the Crowdsourcing track provided each group with a set of topic-document pairs to be judged. In our case, we had 20 topics and between 70 and 100 documents per topic assigned to us for a total of 1875 topic-document pairs. The track also provided between 10 and 35 "gold" judgments for each topic that had come from NIST assessors.

We decided to "self-source" the crowdsourcing task to ourselves given its modest size. We knew from past experience with relevance assessing, that qualified crowd sourced workers can average one document judged every 15 seconds [8], and the top 25% of workers judge at a rate of between 3.1 and 8.1 seconds per document. We have previously been involved in judging large numbers of documents [1, 7], and knew that we could rapidly judge documents. As such, we estimated that we would take between 2 and 4 hours to judge all 1875 documents.

We built a graphical user interface in C# similar to the command line based interface used by the University of Waterloo in previous TREC tracks [2]. The interface allows for single keystroke judging of relevance. The interface has two windows. One window shows the plain text version of the web page and the other window shows the jpeg rendered version. We maximized each window in a separate monitor. To allow maximal viewing with minimal scrolling, each monitor was rotated to be in portrait mode. The plain text version highlighted preselected keywords for each topic.

We did our best to begin judging a topic and not take a break until we were between topics. Sometimes we ran into difficulty with the jpeg display and would need to restart the system. We timed and logged each judgment.

Contrary to the track instructions, we did see and judge documents multiple times before recording a final judgment. For each document that we saw more

than once, we only recorded the final judgment and final amount of time. We needed to judge documents multiple times for a variety of reasons. First, our interface was not bug-free when we began the judging process. As such, we had to rejudge documents until we had the bugs removed from the system. Second, for topic 20812, "free email directory", we marked all but 1 document as non-relevant. Realizing that such a high rate of non-relevance was unlikely, we examined the gold documents and discovered that the NIST assessor had decided to judge documents that linked to email directories as relevant as opposed to following the NIST guidelines that require relevant pages to be the desired relevant page. As such, we rejudged the entire topic to include pages with links to apparently free email directories. Because we examined the gold documents for some topics, we did not submit judgments on the gold.

Excluding the time to build the interface, the time between topics, and the time for rejudged documents, we took an average of 4.7 seconds per document. We spent about one working day to build the interface, but we believe we would have spent at least this amount of time to build the infrastructure required to crowd-source the work.

Our self-sourced judging appears to have fared well compared to the crowd-sourced workers of the other teams. For the reported preliminary results, on consensus judged documents, we obtained a true positive rate (precision) of 90.5% and a false positive rate (1 - specificity) of 8.8%. The average true positive rate of the other teams was 77.5% and the false positive rate was 32.5%. While these rates may not hold out for the final reported results, the rates of the crowd-sourced workers appear consistent with our experience [8].

2.1.1 Observations

For some topics we found the plain text representation easier to judge than the jpeg rendered page. The jpeg page can have distracting images shown in it, while with the plain text these are hidden. Likewise, for some topics, it seemed as easy or easier to judge the rendered jpegs. We suspect there is an advantage to having both views available at once, even though we rarely looked at both views for a given document.

We found the topic descriptions to be very lacking. In many cases, without looking at the gold, it was hard to understand what was to be considered relevant.

We wish we had placed a "pause timing" button on our interface to allow us to take breaks.

Relevance assessments of web documents need more than a binary scale. There appears to be little high quality content compared to the large number of junky, but still relevant as per TREC standards, pages.

2.2 Task 2: Consensus

For the consensus task, we submitted two runs that differed in how we used the supplied gold judgments. UWatCS2Semi is a semi-supervised run while UWatCS2Unsup is nearly a fully-unsupervised run. UWatCS2Semi was our primary run and the only run for which we have preliminary results at this time.

The supplied input data had duplicate judgments in it, i.e. a given worker submitted more than one judgment for a given topic-document pair. We simply threw out duplicate judgments and retained the first judgment that we read into our program.

We computed a document's consensus judgment as the weighted combination of all worker judgments for that document:

$$C(D) = \frac{\sum_i q_i^2 j_i}{\sum_i q_i^2} \quad (1)$$

where q_i is the *quality* of the ith worker with a judgment j_i for the document D. Judgments had a value of 0 for non-relevant and value of 1 for relevant. If the sum $\sum_i q_i^2$ was so small as to produce a floating point NaN value for $C(D)$, we set $C(D)$ to 0.5. We set a worker's quality to be the signal detection measure d' which equals:

$$d' = z(TPR) - z(FPR) \quad (2)$$

where the function z is the inverse of the normal distribution function and converts the TPR or FPR to a z score [5], and TPR is the true positive rate and FPR is the false positive rate. We measured a worker's TPR and FPR for the documents for which the worker had a judgment. For the UWatCS2Semi run, we used the gold judgment as truth if it existed, otherwise we used the computed consensus judgment. For the UWatCS2Unsup run, we used the computed consensus judgment in all cases.

We initialized all workers' quality to be 1. After computing consensus judgments for all documents, we recomputed worker quality with the new judgments. We repeated this process and selected the iteration with the highest accuracy as measured on the gold documents. The process reached the maximum accuracy after 3-4 iterations and showed little decrease in accuracy with additional iterations.

The UWatCS2Semi run appears to have done okay, scoring above the median but short of the maximum

on the held-out gold judgments assessment for key measures of accuracy, true positive rate, and false positive rate.

3 Web Track

For the web track, we submitted 6 runs oriented for evaluation in the ad-hoc task. All web track runs were evaluated for both the ad-hoc and diversity tasks. In general, our runs were designed to be examples of what can easily be done with existing resources.

All of our runs used Category A (English only) filtered to remove 70% of the spammiest material [3]. For each document, we converted it into a plaintext representation first using the Jericho HTML parser (`http://jericho.htmlparser.net/`). We included the HTML page's title and url in the content as well. We also included the Twente anchortext [4] as content.

We then indexed and did retrieval using Indri [9]. We stemmed words with the Krovetz stemmer. Each of the runs was as follows:

- UWatMDSdm: We used dependence models [6] on the full query.
- UWatMDSdmsr: We used dependence models on the query after stopword removal.
- UWatMDSsql: Query likelihood of full query.
- UWatMDSqlsr: Query likelihood of query after stopword removal.
- UWatMDSqlt: Query likelihood of full query. Documents' titles were given extra weight.
- UWatMDSqltsr: Query likelihood of query after stopword removal. Documents' titles were given extra weight.

We did not remove stopwords from the documents when we indexed them, and thus were interested in seeing the effect of querying with the raw queries or with stopwords removed from the queries.

For the ad-hoc evaluation, UWatMDSqlt scored an ERR@20 of 0.144, which placed our group third among all groups. For the diversity evaluation, UWatMDSqltsr scored an ERR-IA@20 of 0.494, which again placed our group third. Across measures, it appears that in general, the dependence model runs did better than the query likelihood with title emphasized runs, which did better than the plain query likelihood runs. There was little difference between retaining or removing stopwords, but there might be some small advantage to retaining stopwords in the queries.

For the ad-hoc evaluation, the UWatMDSdm run did as well or better than the median performance for P@10 for 42 of the 50 topics and had the best P@10 on 3 topics and the worst P@10 on 10 topics (on 12 topics the median P@10 equaled the minimum P@10). UWatMDSqlt and UWatMDSql did as well or better than the median on 39 and 38 of the 50 topics for P@10, respectively.

4 Conclusion

This year we participated in the Crowdsourcing and Web tracks. For the Crowdsourcing track, we submitted a "self-sourced" set of relevance assessments that fared well compared to traditional crowd-sourced workers. Our crowdsourcing consensus method was simple in nature but performed a little above the median on primary measures. Finally, our easy to construct web track runs appear to have achieved reasonable performance.

5 Acknowledgments

David Hu and Aiman L. Al-Harbi assisted with processing of the ClueWeb09 corpus. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the University of Waterloo. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) and Compute/Calcul Canada. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMass at TREC 2004: Novelty and HARD. In E. M. Voorhees and L. P. Buckland, editors, *The Thirteenth Text REtrieval Conference (TREC 2004)*. Department of Commerce, National Institute of Standards and Technology, 2004.

[2] G. V. Cormack and M. Mojdeh. Machine learning for information retrieval: TREC 2009 web, relevance feedback and legal tracks. In *TREC 2009*. NIST.

[3] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, pages 1–25, 2011. 10.1007/s10791-011-9162-z.

[4] D. Hiemstra and C. Hauff. Mirex: Mapreduce information retrieval experiments. Technical Report TR-CTIT-10-15, Centre for Telematics and Information Technology University of Twente, Enschede, April 2010.

[5] N. Macmillan and C. Creelman. *Detection theory: a user's guide*. Lawrence Erlbaum Associates, 2005.

[6] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *SIGIR'05*, pages 472–479. ACM, 2005.

[7] M. D. Smucker, C. L. A. Clarke, G. V. Cormack, and O. Vechtomova. University of Waterloo at TREC 2010: Legal interactive. In *The Nineteenth Text REtrieval Conference (TREC 2010)*, 2010.

[8] M. D. Smucker and C. Jethani. The crowd vs. the lab: A comparison of crowd-sourced and university laboratory participant behavior. In *Proceedings of the SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval*, July 2011.

[9] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language-model based search engine for complex queries (extended version). Technical Report IR-407, CIIR, CS Dept., U. of Mass. Amherst, 2005.