UWaterlooMDS at the TREC 2018 Common Core Track

MUSTAFA ABUALSAUD[1], GORDON V. CORMACK[1], NIMESH GHELANI[1], AMIRA GHENAI[1], MAURA R. GROSSMAN[1], SHAHIN RAHBARIASL[1], MARK D. SMUCKER[2], and HAOTIAN ZHANG[1],

[1] David R. Cheriton School of Computer Science, University of Waterloo

[2] Department of Management Sciences, University of Waterloo

This year we applied dynamic sampling (DS) [4] to create a sampled set of relevance judgments. One goal was to test the effectiveness and efficiency of this technique with a set of non-expert, secondary relevance assessors. We consider NIST assessors to be the experts and the primary assessors. Another goal was to make available to other researchers a sampled set of relevance judgments (prels) and thus allow the estimation of retrieval metrics that have the potential to be more robust than the standard NIST provided relevance judgments (qrels). In addition to creating the prels, we also submitted several runs based on our manual judging and the models produced by our HiCAL system [1, 6].

1 DYNAMIC SAMPLING

While we detail the steps of dynamic sampling (DS) in Algorithm 1, in this section, we first give a general overview of the process followed and then later give the details.

1.1 Overview

Dynamic sampling (DS) [4] is a technique that creates a stratified sample of relevance judgments for test collection construction. DS is performed for each topic in a test collection. In our implementation of DS, we start by creating a "zeroth" stratum that consists of judged documents found from a mixture of continuous active learning (CAL) [2, 3] and interactive searching and judging (ISJ). Because these documents are found without any sampling, they are all given an inclusion probability of 1.0. The authors performed all judging themselves.

With a set of relevant and non-relevant documents forming our zeroth stratum, we use these documents, plus a random selection of unjudged documents assumed to be non-relevant, to train a classifier. We then rank all documents, that are not yet a member of a stratum, using the classifier. To rank the documents, we first divide each document into non-overlapping *paragraphs*. We then rank all paragraphs and select B unique documents with the highest scoring paragraphs. These B documents form the next stratum. From the stratum, we then use simple random sampling to sample n documents for judging. Each of these documents will have an inclusion probability of n/B. When we judge documents for relevance, we select the paragraph with the highest probability of relevance as per our classifier and judge this paragraph. We do not view the whole document. Past research has shown that judging paragraphs is more efficient than judging full documents, and the judgments have comparable quality [7, 8].

Both the stratum size B and the number of documents to sample n vary for the strata. Each stratum is larger than the previous one, and as relevant documents are found, the inclusion probability decreases. In addition to the differences between the strata, after each stratum is sampled and judged, the classifier is trained anew and thus should be better at finding relevant documents for formation of the next stratum.

1.2 Details

Algorithm 1 details the steps of our implementation of dynamic sampling. In Step 1, one of the authors used a live, human-in-the-loop, AutoTAR CAL implementation to assess in total 3648 documents over 50 topics, 1419 of

them are relevant (38.9%). This author spent a total of 11.1 hours judging documents (13.2 minutes per topic). The same CAL implementation was used for relevance assessments for the MRG_UWaterloo submission in TREC Common Core Track 2017 [5]. These assessments in the initial train set were used to build the classifier in Step 7.

Among the assessments for 50 topics, we found 20 topics had less than 10 documents judged as relevant. In order to provide a better prime model for dynamic sampling, we used interactive search and judging to augment the relevance assessments on those topics in Step 2. We conducted ad-hoc searches using the search model of the HiCAL system [1]. The interfaces of the search model remained the same but we replaced the back-end Indri search engine with Anserini[1]. Five authors used this search engine and tried to find at least 10 relevant documents on those 20 topics. The search engine returned 50 results by default. We reformulated queries as many times as we wanted. We allocated a maximum of 30 minutes of judging per each of these 20 topics.

We merged the relevance assessments from the AutoTAR CAL judgments and from interactive search and judging (ISJ). In some cases, the same document was judged by both the CAL process and the ISJ process. If a document was found to be relevant by either process, it was considered relevant. After merging these two sets of assessments, we had in total 4161 judgments for 50 topics, in which 1645 of them were relevant (39.5%). These assessed documents form an initial seed set (the zeroth stratum) and were not shown to assessors again in the dynamic sampling process.

After having the initial seed documents in steps 1 and 2, we started the dynamic sampling process. We made the judgments on the CAL model of HiCAL system through step 6 to 14. In each iteration of dynamic sampling, the system displayed the selected paragraphs to assessors. There was no option to view the full document. Each document was assessed only once and not shown to assessors again. The assessors judged 300 paragraphs for each topic and then the dynamic sampling process stopped. In Step 4, we set $N = 25$ in our experiment.

We randomized the 50 topics and assigned them to five authors. Three assessors judged 24 topics in the first pass. We found there existed a bug in our code. Therefore, we rejudged those 24 topics in the second pass and finished all 50 topics. There was no time limit for assessing documents for each topic. The assessments were finished within one week and averaged 33 minutes per topic. In total, we spent about one hour per topic to produce our relevance judgments.

2 RELEVANCE JUDGMENTS

The output of dynamic sampling is a set of sampled documents with inclusion probabilities. For each of these sampled documents, we have a relevance judgment. The judgments plus inclusion probabilities are called prels. The prels contains five fields: topic, assessed document id, stratum number, the inclusion probability of the assessed document, and relevance judgment of the document.

We provided our prels to NIST, who then had NIST assessors judge the same documents. With the NIST judgments and the inclusion probabilities, NIST was able to estimate the number of relevant documents in the collection for each topic.

As presented at TREC, at http://cormack.uwaterloo.ca/sample/, we provide our relevance judgments along with a new evaluation program, DynEval, written by Gordon Cormack that can use either traditional trec_eval qrels or xinfAP irels. In addition to our judgments, we also provide irels that combine our work with the judging done by NIST.

3 SUBMITTED RUNS

We submitted four runs to the Common CORE track 2018 for evaluation. All these runs were manual runs. As per agreement with NIST, none of these runs were part of the pooling.

[1] https://github.com/castorini/Anserini

ALGORITHM 1: Dynamic sampling algorithm used in this experiment.

Step 1. Use CAL to discover and label initial training set;
Step 2. Use interactive search and judging to augment the initial training set for the topics on which Step 1 yielded fewer than 10 relevant documents;
Step 3. Set the initial batch size B to 1;
Step 4. Set the initial decay threshold T to hyper-parameter N;
Step 5. Set the initial number of assessments A to 0;
Step 6. Temporarily augment the training set by adding 100 random document from the collection, temporarily labeled "not relevant";
Step 7. Construct a classifier from the training set and score all the paragraphs in the collection;
Step 8. Remove the random documents added in step 6;
Step 9. From the documents not yet part of a sampled stratum, select B documents such that they contain the highest scoring paragraphs.;
Step 10. Draw $n = \lceil \frac{B \cdot N}{T} \rceil \leq B$ random documents from the Step 9 documents;
Step 11. Assess relevance of the n documents based on viewing the highest scoring paragraph for each document. Update $A = A + n$;
Step 12. Add the assessed documents to the training set;
Step 13. Increase B by $\lceil \frac{B}{10} \rceil$;
Step 14. If the number of assessed relevant documents $R \geq T$, double T;
Step 15. Repeat step 6 through 14 until $A = 300$ documents have been assessed.

For each topic, we built a final classifier using all the judgments from the initial training set (stratum 0) and the dynamic sampling iterations (stratum 1, 2, 3, ...). The classifier was built on document features and was used to score all the documents in the collection.

UWaterMDS_DS_A: This run is composed by all documents we judged as relevant, ordered by the final classifier. If we found fewer than 10 relevant documents, we then appended all documents we judged as non-relevant, ordered by the classifier.

UWaterMDS_DS_B: This run is composed by all documents we judged as relevant, in reverse order by the final classifier. If we found fewer than 10 relevant documents, we then appended all documents we judged as non-relevant, ordered by the classifier.

UWaterMDS_Rank: We use the final classifier to rank all documents and include the top 10,000 documents.

UWaterMDS_SEQ: The run is generated based on the order of our judgments. For the stratum 0, we first put the AutoTAR CAL judged relevant documents and then the ISJ judged relevant documents, ordered by time of discovery. For the remaining strata, if a stratum had all documents sampled (inclusion probability 1.0), we put all judged relevant documents (no non-relevant ones) from that stratum by the order discovered. For the strata with documents having inclusion probabilities smaller than 1.0, we put all documents (including non-relevant and unjudged documents), ordered by the classifier.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (Grants CRDPJ 468812-14, RGPIN-2017-04239, and RGPIN-2014-03642), and in part by the University of Waterloo.

REFERENCES

[1] Mustafa Abualsaud, Nimesh Ghelani, Haotian Zhang, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. A System for Efficient High-Recall Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 1317–1320.

[2] Gordon V Cormack and Maura R Grossman. 2014. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 153–162.

[3] Gordon V. Cormack and Maura R. Grossman. 2015. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. *CoRR* abs/1504.06868 (2015). http://arxiv.org/abs/1504.06868

[4] Gordon V. Cormack and Maura R. Grossman. 2018. Beyond Pooling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 1169–1172. DOI: http://dx.doi.org/10.1145/3209978.3210119

[5] Maura R. Grossman and Gordon V. Cormack. 2017. MRG_UWaterloo and WaterlooCormack Participation in the TREC 2017 Common Core Track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*.

[6] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Angshuman Ghosh, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2017. UWaterlooMDS at the TREC 2017 Common Core Track. TREC.

[7] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. Effective User Interaction for High-Recall Retrieval: Less is More. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 187–196. DOI: http://dx.doi.org/10.1145/3269206.3271796

[8] Haotian Zhang, Gordon V. Cormack, Maura R. Grossman, and Mark D. Smucker. 2018. Evaluating Sentence-Level Relevance Feedback for High-Recall Information Retrieval. *arXiv preprint arXiv:1803.08988* (2018).