Retrieving Scientific Abstracts using Medical Concepts and Learning to Rank: CincyMedIR at TREC 2020 Precision Medicine Track

Piyush Sahu[1], Hoang Vu [1], Danny T.Y. Wu, PhD, MSI [1]

[1]University of Cincinnati College of Medicine, Cincinnati, OH

Introduction

The CincyMedIR team led by Dr. Danny T.Y. Wu at the University of Cincinnati College of Medicine (UC CoM) participated in the Text Retrieval Conference 2020 Precision Medicine Track (TREC-PM). CincyMedIR only worked on the scientific abstracts this year with two main objectives: 1) to experiment learning to rank (LTR) models, a supervised machine-learning approach to adjust ranking based on the text features in the relevant documents, and 2) to develop a configurable pipeline for TREC-like tasks.

Method

CincyMedIR continued using ElasticSearch (ES) as the information retrieval (IR) platform. Through Python 3.6 and ES Application Programming Interfaces (APIs), an IR pipeline was developed and tested. Our team has been using the same set of tools for previous TREC tasks and gained working knowledge. In terms of the IR pipeline, first, all scientific abstracts were downloaded from the website and indexed using ES on Amazon Web Services (AWS). Second, the queries were formed by combining the disease and gene terms in the topic files (no treatment terms). No query expansion or modifications were done due to the limited improvement shown in our previous experiments. Third, the top 2,000 documents were retrieved for each query in TREC-PM 2020 and 2019 and combined with the annotated documents in the evaluation (qrel) files in 2019 and 2018. Fourth, these documents were parsed by Metamap to get their medical concepts; the terms and concepts were indexed for LTR. Then, eight LTR models were trained on 2018 documents and validated on the 2019 documents. Next, the top 1,000 relevant documents for each query were retrieved using BM25 and the best performed LTR model (random forest). In this step, queries may be modified to include the treatment terms. Lastly, five runs were generated for submission.

Results

Our team implemented a configurable pipeline with multiple steps to conduct the experiment for TREC-like tasks. The steps included index preparation, query preparation, metamap parsing, LTR model training and validation, search retrieval, and evaluation on qrels files. The pipeline was able to support the experiment this year in an efficient manner.

Table 1 shows that our submitted runs did not performed well. None of their Rprec, Precision at 10 (P_10) and infNDCG were above the median scores of all teams. The best performed run (dgt_trec_eval) did not use LTR nor re-rank methods at all but only added the treatment terms in the queries in a later step. We suspect that the poor performance resulted from the exclusion of treatment terms in the beginning of the process. Even though the re-rank methods considered the treatment terms, they did not improve the performance much. Therefore, we created another scenario by including the treatment terms in the beginning and generated

additional four runs. The re-ranking was changed from term-based to concept-based methods, which were proven to be effective in the last year's runs. The results show that one of the addition runs was able to achieve a reasonable performance with infNDCG slightly above the median score. However, this run did not use any LTR and re-ranking methods.

Table 1. Performance of Submitted and Additional Runs.

CincyMedIR	filename	Description	map	bpref	Rprec	P_10	infNDCG
Submitted Runs (N=5)	**dgt.trec_eval**	Q_DGT	0.2266	0.2491	0.2678	0.4452	0.3877
	28_t.trec_eval	Q_DG, LTR, RR_T	0.1495	0.1742	0.1822	0.3839	0.2721
	28dgt.trec_eval	Q_DG, LTR, RR_DGT	0.0302	0.0659	0.0664	0.0742	0.0900
	28.trec_eval	Q_DG, LTR.	0.0281	0.0646	0.0603	0.0710	0.0852
	20.trec_eval	Q_DG	0.0210	0.0483	0.0394	0.0516	0.0621
Additional Runs (N=4)	Not submitted	Q_DGT.	0.2859	0.2978	0.3124	0.4548	0.4338*
		Q_DGT, RR_CID	0.2830	0.3003	0.3133	0.4581	0.4221
		Q_DGT, RR_CT	0.2711	0.2851	0.2931	0.4516	0.4114
		Q_DGT, LTR	0.1782	0.1883	0.1823	0.3290	0.3137

Q_DG(T): queries were formed using disease, gene, (and treatment) in the XML.

LTR: Learning to Rank models trained on TREC-PM 2018 and validated on TREC-PM 2019

RR_(DGT|T|CID|CT): re-ranked methods; DGT) re-rank the results by moving a document up if it matches all disease, gene, and treatment terms; T) matches treatment terms; CID) matches concept ids; CT) matches concept terms.

* above the median score of all teams; Rprec (0.325), P_10 (0.464), and infNDCG (0.431).

Conclusion

CincyMedIR developed a python-based pipeline on ES to quickly respond to the TREC-PM tasks this year. However, the submitted and additional runs did not achieve high performance using LTR and medical concepts. It seems that retrieval with plain keywords and a classic ranking algorithm can generate a reasonable baseline. Due to the time constraint, we were not able to run more experiments to increase the performance from the baseline results. We will learn from other teams' techniques in the conference and continue to refine our results and the pipeline using the TREC-PM data in all four years between 2017 and 2020.

This is the last year of TREC-PM, which means a new medically related track will be created in TREC 2021. We thank the organizers of TREC-PM for their effort and look forward to participating in the new track next year.