Retrieving scientific abstracts iteratively: MedIER at TREC 2018 Precision Medicine Track

Jinghui Liu,[1] **Clair Kronk,**[2] **Wu-Chen Su,**[2] **Danny T.Y. Wu,**[2] **V.G.Vinod Vydiswaran**[3,1]
[1] School of Information, University of Michigan
[2] Department of Biomedical Informatics, University of Cincinnati
[3] Department of Learning Health Sciences, University of Michigan
ljinghui@umich.edu, {kronkcj, suwc, wutz}@ucmail.uc.edu, vgvinodv@umich.edu

Abstract

This paper describes the approach developed by the MedIER team – a collaboration between the University of Michigan and the University of Cincinnati – for the TREC 2018 Precision Medicine Track. We implement an iterative approach of document retrieval with modified queries, and combine the results by formulating re-ranking as a text classification task. We evaluate our proposed framework to retrieve biomedical research abstracts. Our experiments show that the iterative re-retrieval approach is effective in retrieving higher number of relevant scientific abstracts.

1 Introduction

As the precision medicine paradigm takes roots, advancements are being made on medical interventions tailored to an individual based on their genetic, environmental, or behavioral information. This is especially true in precision cancer treatments and research integrating genotypic and phenotypic information. However, it is often overwhelming for clinicians and biomedical researchers to find relevant articles and prior scientific breakthroughs directly relevant to specific individual characteristics. As more and more studies and clinical trials explore genetic variations among individuals, intelligent retrieval tools can help clinicians to identify relevant evidence from published literature that can contribute to clinical decision-making.

The TREC Precision Medicine Track aims to address this issue by enabling development of novel information retrieval techniques that incorporate genetic or person-specific information to efficiently locate relevant scientific abstracts and clinical trials. The track organizers defined a number of oncology-related cases ("topics"), where each case describes a specific type of cancer, a particular gene with a specific genetic variant, and patient demographic information. Two tasks were designed – one on retrieving scientific abstracts that can help clinicians find prior literature most relevant to the oncology case; and the other on retrieving relevant clinical trials that the patient might be eligible for. We participated in the scientific abstract task.

Topic id	Description
Topic 4	⟨disease⟩ melanoma ⟨/disease⟩ ⟨gene⟩ BRAF (K601E) ⟨/gene⟩ ⟨demographic⟩ 38-year-old male ⟨/demographic⟩
Topic 19	⟨disease⟩ melanoma ⟨/disease⟩ ⟨gene⟩ extensive tumor infiltrating lymphocytes ⟨/gene⟩ ⟨demographic⟩ 49-year-old male ⟨/demographic⟩

Table 1: Example topics in TREC 2018 Precision Medicine track

This paper presents the details of the system we developed as part of our participation, describes the submitted runs, and summarizes their performance.

2 Data

For TREC 2018 Precision Medicine track, fifty topics were defined based on synthetic cases created by oncologists at the University of Texas MD Anderson Cancer Center. Most of the fifty topics contained three parts: type of cancer, gene(s) with or without variants or locus, and basic demographic information (e.g., Topic 4 in Table 1). In six of the fifty topics, instead of the gene name, a description of a tumor-related condition or biomarker was provided (e.g. Topic 19 in Table 1).

The scientific abstracts corpus was obtained from two sources – MEDLINE abstracts and conference proceedings. The first source is the January 2017 snapshot of MEDLINE, which consists of 26.67 million documents with title, abstract, and article metadata including MeSH terms and publication type. The second source is the collection of conference proceedings from American Association for Cancer Research (AACR) and American Society of Clinical Oncology (ASCO), since these documents are more targeted towards cancer therapy and hence relevant to the track topics. This collection contains about seventy thousand documents.

3 Previous Strategies

In our previous strategies working on the TREC 2017 Precision Medicine task,[9] the MedIER team explored

the effectiveness of the query expansion and other strategies to leverage medical ontologies. The expansion strategy included four components: (a) identifying synonyms from the NCBI database [8], (b) pruning the expanded gene names based on inclusion information, provided by NCBI, on genes and associated PubMed articles, (c) finding common synonyms for diseases, based on MeSH terms [6], and (d) querying a MeSH identifier index to retrieve additional documents. In addition to these components, we also tried other approaches for the topic expansion, such as expanding gene variations with protein identifiers and enriching disease terms using UMLS [1] and SNOMED CT [4].

However, these strategies showed limited effect. For query expansion, we suspect that, despite being useful in cases like handling typographic variations, such as adding "K-ras" to "Kras", the expansion introduces noisy terms that affect the quality and ranking of retrieved documents by creating verbose queries that drift from the desired topic. Also, certain expansion terms, especially genes, do not appear in human-related context and are not helpful in improving the quality of retrieved documents.

4 Current System

The system we developed for this year's participation, summarized in Figure 1, was based on iterative cycles of query generation and document re-ranking. Different from our previous strategies that mainly focused on query expansion, the current system enabled us to explore a structured retrieval strategy based on minimal query modification, and the use of machine learning for re-ranking. Overall, the system is composed of three processing steps – (a) query generation and initial retrieval, (b) query modification and retrieval with re-ranking, and (c) iterative re-retrieval. The following subsections describe these steps in detail.

4.1 Indexing

For both MEDLINE abstracts and AACR/ASCO conference proceedings, we indexed the article ID, title, and abstract text using Apache Solr 7.4.0 with default settings for tokenization and stopword removal. This produces a combined index of 26,739,419 documents.

Since the Precision Medicine track focuses on individual (human) characteristics, biomedical papers focusing on animal models or non-clinical topics were considered less relevant. To identify papers belonging to these two categories, we ran a concept annotator system, called PubTator,[5] over MEDLINE abstracts. PubTator outputs the entity type, nomenclature, and taxonomy ID of entities (e.g. humans, animals, etc.) mentioned in the given input text. For each scientific abstract, the PubTator output fields were indexed into a separate, second index. The index enabled us to check for the presence or absence of the human taxonomy ID tags as a filter and ensure that retrieved documents are restricted to human studies.

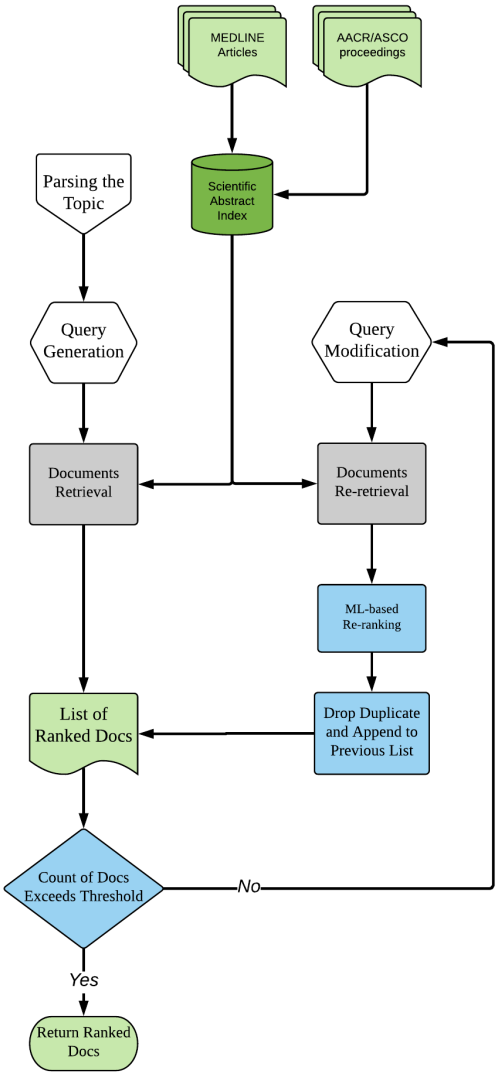

Figure 1: Schematic diagram of the current system

4.2 Topic parsing and query generation

In the current system, topics were analyzed and parsed into groups of terms before building queries over them. We focused on the gene and disease fields in a topic. We parsed the disease field into individual terms, and the gene field into genes and variants. For example, the disease field string "thyroid cancer" was split into "thyroid" and "cancer", and the gene field string "BRAF (K601E)" was parsed as "BRAF" as a gene name and "K601E" as a variant. For the six topics that included biomarkers instead of gene information, we ran a part-of-speech tagger to find nouns and adjectives, which were then treated as parsed "genes" and "variants" respectively. For example, for Topic 19 in Table 1, the parsed "genes" are "tumor" and "lymphocytes", and the parsed "variants" are "extensive" and "infiltrating". These parsed topics were then used to generate queries.

For each topic, a boolean search query was constructed by including the terms from the parsed topic,

Types of specificity	Ex. ⟨disease⟩thyroid cancer⟨/disease⟩ ⟨gene⟩BRAF (V600R)⟨gene⟩
Strict match	+(thyroid cancer) +BRAF +V600R
Relaxed variant	+(thyroid cancer) +BRAF V600R
Relaxed phrase	+thyroid +cancer +BRAF V600R
Relaxed disease	+thyroid cancer +BRAF V600R
Lenient match	thyroid cancer BRAF V600R

Table 2: Example of the five types of query built from parsed topic

by specifying whether a term match should be "strict" (i.e. the term must match) or "relaxed" (i.e. the term may or may not match). The following scenario describes when a query needs modification and how it is done. Every time a query is used to retrieve a set of documents, the system will check whether it is "satisfied" with these documents and decide whether to return the set as the final output or to retrieve an additional set. If an additional set of documents needs to be retrieved, the previous query is modified accordingly. In the current system, the count of documents in the set is compared against an empirically selected threshold as a naïve measure of "satisfaction". Consequently, the query is modified mainly by relaxing the term specificity and making it lenient so that more documents can be retrieved. For example, given Topic 4 (see Table 1), the initial query that aims to strictly match all disease and gene terms in a boolean query format will be "+melanoma +BRAF +K601E". By relaxing the "variant" to retrieve more documents, the query will be modified as "+melanoma +BRAF K601E". The current system has five types of query built from the parsed topic, and they are summarized in Table 2.

In addition to the iterative cycles of query generation and modification, some optional terms are added to give higher scores to documents related to clinical intervention. Words including "treatment" and "patient" are appended to the query as selected terms. Since these terms are optional in the boolean query, we may still retrieve documents that do not mention these terms explicitly and that focus on animal experiment, which is less likely to affect the overall retrieval quality.

4.3 Document retrieval and re-ranking

With the set of query types described in Sec. 4.2, documents are retrieved in an iterative fashion as shown in Fig. 1. The initial query generated from the parsed topic is a strict match of all terms and phrases. This query is used to retrieve an initial set of documents, and its count is recorded and compared against a set threshold. In our current set, we set the threshold as 500. If the number of retrieved documents exceed the threshold, the system regards the initial query as being specific and informative enough for the processing of the topic to rely solely on the retrieval algorithm. Therefore, these initial documents are returned as the final ranked list. If the number of retrieved results is lower than the threshold, the system proceeds with the modified query and retrieves an additional set of documents. These documents are re-ranked and merged with the previous result list.

In the current implementation, we treat the re-ranking step as a text classification task. For training the re-ranking classifier, we use the top ten documents retrieved by the initial query for each topic as the training data, and the current set of retrieved documents as the test data. As described above, the initial query matches the given topic fields most strictly and would lead to the least number of retrieved documents. Further, human-specific studies would rank higher because of the additional terms specifically added to the query. Inspired by the relevance feedback approach, top ten results are regarded as highly relevant. For the test data, or the current set of documents retrieved with a lenient query, documents that already appear in the original set are removed. For the remaining documents, the abstract texts are converted into word matrices with tf-idf weighting. A one-class Support Vector Machine (SVM) model [3] is trained on the top ten documents and applied on the remaining documents. If a document is predicted by the classifier as a positive class, it's ranking score is boosted and merged with the overall result set, resulting in a re-ranked list of retrieved documents. This iterative step is repeated until sufficient number of documents have been retrieved. It should be noted that re-ranking is done using the original classifier model (i.e. the classifiers are not re-trained in every iteration).

4.4 Other considerations

In our initial analysis of the retrieved documents, we noticed that the runs using the PubTator index did not perform well, comparatively to just using the primary index. We suspect there are three main reasons for this – including incomplete PubTator database, limited range of taxonomy IDs, and PubTator's inherent deficiency with respect to documents in the corpus. The current version of the annotated PubTator database contains only 21,251,023 MEDLINE abstracts. Further, annotations are not available for any of the conference proceedings, leading to an incomplete index. Further, with respect to the taxonomy ID, we observed that the limited focus on human subjects neglects other relevant animal studies that should be retrieved as they may hold potential benefits / linkages to human studies. Finally, there were some relevant abstracts that did not mention any terms related to human, such as "patients" or "people", which results in the failure of using the taxonomy annotation to decide whether a study is clinically relevant to humans.

4.5 Similarity and Rank Fusion

To formulate different runs for the TREC task, the system is run with different configuration on retrieval models and rank processing. To compare the effec-

Run ID	infNDCG	P@10	R-prec
MedIER_sa11 (BM25)	0.5491	**0.6220**	0.3647
MedIER_sa12 (QL)	0.5329	0.5940	0.3484
MedIER_sa13 (Fusion of sa11 and sa12)	**0.5515**	0.6140	**0.3684**
MedIER_sa14 (BM25+re-ranking)	0.5449	0.6200	0.3642
MedIER_sa15 (QL+re-ranking)	0.5432	0.5960	0.3497
Best score	0.5621	0.7060	0.3684

Table 3: Average performance on the three evaluation metrics over 50 topics

tiveness and performance of the iterative retrieval approach, we designed runs with or without re-ranking using two different retrieval algorithms: BM25 [7] and Query Likelihood (QL) [10]. We used Reciprocal Rank Fusion to merge results from different queries. Reciprocal Rank Fusion (RRF) has been shown to be effective in improving ranking quality [2]. We combined the two lists of documents produced by the system using the two retrieval models without re-ranking. Sec. 5 describes the submitted runs in more detail.

5 Submitted Runs

Five runs were submitted to explore whether the standalone iterative approach is effective and whether re-ranking can contribute to the performance of the system. These runs were:

- MedIER_sa11: This run is based on iterative looping using BM25 retrieval model to produce the ranked lists.
- MedIER_sa12: This run is based on iterative looping using Query Likelihood model to produce the ranked lists.
- MedIER_sa13: This run is the fusion of the previous two runs based on the iterative looping using either BM25 or QL retrieval model to generate the ranked lists.
- MedIER_sa14: This run is based on the iterative looping with machine learning re-ranking and BM25 retrieval model to produce ranked lists.
- MedIER_sa15: This run is based on the iterative looping with machine learning re-ranking and QL model to produce ranked lists.

6 Results

For the scientific abstracts task, three evaluation metrics are reported on the overall performance: the inferred NDCG (infNDCG), Precision@10 (P@10), and R-Precision (R-prec). Table 3 summarizes the performance of our submitted runs. We note that the top performances of the three metrics fall into two runs. The MEDIER_sa13 run obtains the highest score for both inferred NDCG and R-Precision measures, suggesting the effectiveness of the iterative looping approach using different similarity algorithms. Also, comparing runs with or without re-ranking, it can be seen that the success of the technique is correlated to the use of similarity algorithm. When using Query Likelihood model as the similarity function, re-ranking approach seems to be effective, increasing infNDCG from 0.533 to 0.543. For Precision@10, it can be seen that the run using BM25 algorithms based on the iterative approach (MedIER_sa11) obtains the highest score. Slight differences in Precision@10 between runs with or without re-ranking (MedIER_sa11 versus MedIER_sa14 and MedIER_sa12 versus MedIER_sa15) should be caused by cases where the top ten documents retrieved by the initial query for training machine learning algorithm are moderately different. This would happen when a topic contains such rare disease or gene descriptions that the initial query is extremely strict and retrieves less than ten documents. It does not seem to be the case for this year's topics. For R-Precision, the rank fusion based approach has the highest score on average and the use of re-ranking almost does not affect the performance measured by R-precision.

7 Conclusion

In this paper, we describe a system we developed for participating in the scientific abstracts task of the TREC 2018 Precision Medicine Track. The system is based on an iterative approach of query generation and document re-ranking, and includes three processing steps: i) query generation and initial retrieval, ii) query modification and retrieval with re-ranking, and iii) iterative re-retrieval. Our experiments reveal that both iterative re-retrieval and re-ranking can be useful in improving scientific abstract retrieval. The iterative looping approach was combined with different similarity algorithms, but produced consistent results measured by Precision@10 and R-precision. Re-ranking was found to be effective based on our analysis of inferred NDCG using certain similarity algorithms.

References

[1] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical termi-

nology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[2] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759. ACM, 2009.

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] Roger A Cote and Stanley Robboy. Progress in medical information management: Systematized nomenclature of medicine (snomed). *Jama*, 243(8):756–762, 1980.

[5] Ning Kang, Rogier Barendse, Zubair Afzal, Bharat Singh, Martijn J Schuemie, Erik M van Mulligen, and Jan A Kors. A concept annotation system for clinical records. *arXiv preprint arXiv:1012.1663*, 2010.

[6] Fernando Minguet, Lucienne Van Den Boogerd, Teresa M Salgado, Cassyano J Correr, and Fernando Fernandez-Llimos. Characterization of the medical subject headings thesaurus for pharmacy. *American Journal of Health-System Pharmacy*, 71(22):1965–1972, 2014.

[7] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[8] Stephen T Sherry, M-H Ward, M Kholodov, J Baker, Lon Phan, Elizabeth M Smigielski, and Karl Sirotkin. dbsnp: the ncbi database of genetic variation. *Nucleic acids research*, 29(1):308–311, 2001.

[9] Tong Yin, Danny TY Wu, and VG Vinod Vydiswaran. Retrieving documents based on gene name variations: Medier at trec 2017 precision medicine track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Text REtrieval Conference, TREC*, November 2017.

[10] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.