ECNU-ICA team at TREC 2019 Precision Medicine Track

Qi Zheng[1], Yong Li[1], Jiaying Hu[1], Yi Xue[2], Yan Yang[1], and Liang He[1]

[1]Department of Computer Science, East China Normal University
[2]University of Nottingham Ningbo China

Abstract

In this paper, we describe our biomedical retrieval system used in TREC 2019 precision medicine track. Based on existing querying framework, we develop a multi-pass retrieval system to adaptively refine query template based on indexing feedback. After initial retrieval process, We further re-rank the retrieved documents using language model based re-ranker to get the final results.

1 Introductions

In the field of cancer treatment, precision medicine aims to provide better and more personized treatments for patients. However, due to rapid development of related research, it becomes essential to build automatic system for tackling with large database. The major goal of TREC 2019 precision medicine track is to provide useful precision medicine-related information to clinicians for treating cancer patients. Same as previous year, the precision medicine track is divided into two subtasks, namely biomedical articles and clinical trials. For the biomedical articles tasks, participants are challenged to retrieve article abstracts from MEDLINE/PubMed data dumps. While in the clincal trials subtask, participants are required to retrieve relevant clinical trials(from ClinicalTrials.gov) for which patients are eligible.

In this work, we present a multi-pass query system for effective documents retrieval. Our system mainly consists of two stages, documents retrieval pass and re-ranking pass. In documents retrieval pass, we build the initial query by composing templates from multiple sources and iteratively refine the query in a heuristic way. In the second pass, we use a pre-trained doc2vec model to further rerank the candidates documents retrieved in the first pass. We will detail our method in the following chapter.

2 Approach

2.1 Documents Pre-process

Before performing any kind of querying or indexing, we reformulate the original XML document dumps into JSON format by extracting data fields that we are insterested in. For PubMed documents data dumps, these include article title, abstract text, publication type, MeSH (Lipscomb, 2000) heading list, Medline key words; while for clinical trials data dumps, there are summary, gender, age from clinical trials data dump. We perform several kinds of pre-processing during extraction stage, including mapping age into MeSH age group and tokenizing words inside documents.

2.2 Query Generation

The major goal of this stage is to generate a proper query so that we can cover as many candidates documents as possible, while still keeping the size of the candidate sets at a tractable level. The keywords used in query mainly come from two sources: the initial data in the topics provided from TREC official, and the additional information for query expansion based on hand-crafted rules.

We explore various ways for expanding query, including: (1) Synonyms for diseases and genes (2) Hypernyms for genes and diseases (3) Mapping age number into MeSH age groups (4) Additional keywords based on disease type (5) Detailed description of diseases and genes.

The synonyms, hypernyms and descriptions of diseases and genes can be automatically extracted by querying existing medical knowledge bases. In our case most of these information are extracted from UMLS (Bodenreider, 2004), and we use Lexigram API[1] for online querying.

Our retreival system is based on Elasticsearch (Divya and Goyal, 2013). Elasticsearch

[1]https://www.lexigram.io/

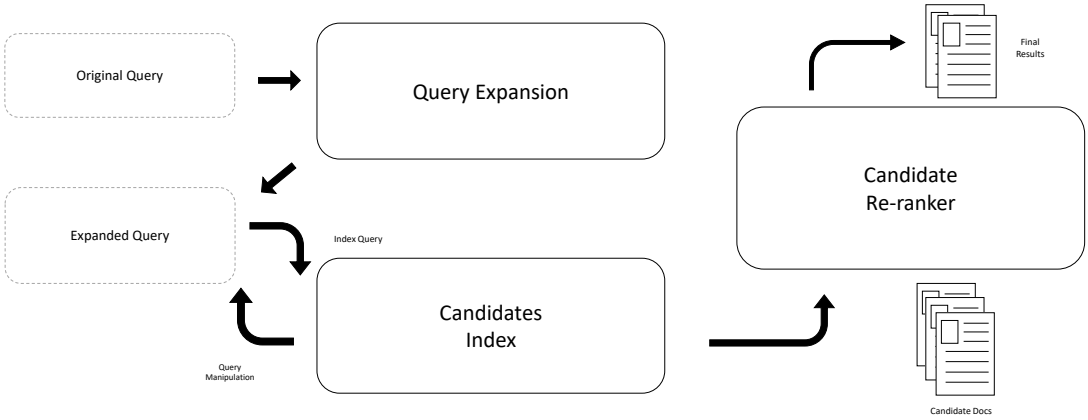

Figure 1: Illustration of our query process

uses BM25 (Robertson et al., 1995) as its main retrieval algorithm. In addition, Elasticsearch provides flexible ways to assign weights and boosting value to different queries and indexing fields. For example, in some cases diseases genes may play a more important role in confirming the solution for patients, hence we need to assign a higher weight to related type of keywords when indexing specific fields, such as abstract and documents title. We tweak these parameters based on provided data dumps and topics used in TREC 2018 precision medicine track.

2.3 Iterative Refinement

Due to the fact that in some topics, disease and gene have a stronger connection within each other while others don't, we need to dynamically change our query template rules during indexing. Based on simple heuristics, here we preform a dynamic refinement stage after getting the initial generated query. We first search the data dump with the initial query we get. After getting the results of the first query, we tweak the query template clause based on the number of documents we get. If the number of documents is below the lower-bound threshold t_{min}, we move some rules from "Must" clause into "Should" clause. In this way we can relax the restriction and cover a wider range of candidate documents. Similarly, if the number of documents retrieved is above the upper-bound threshold t_{max}, we move some of the rules in "Should" clause into "Must" clause to put a stronger constraint on our query so that we can filter out documents that are not specfic enough. We also explore tweaking fields weights and other querying weights during refinement, but we did not observe significant performance difference, so we keep most parameters unchanged during this stage.

2.4 Candidates Re-ranker

In re-ranking stage, we use an encoder to encode both candidates and query into a fixed length vector, and use their cosine score as similarity metrics:

$$score(\mathbf{d}, \mathbf{q}) = cosine\_sim(enc(\mathbf{doc}), enc(\mathbf{q}))$$

where **d** denotes candidate documents, **q** denotes query. We re-train a **doc2vec** model using the provided medical data dumps as our training corpus. We also experiment with deep pre-trained language model, including BERT (Devlin et al., 2018) (Vaswani et al., 2017), ELMo (Peters et al., 2018) and GPT (Radford et al., 2019). However, we didn't observe any significant increase in performance. This was largely due to the fact that the training corpus used by these language models doesn't include documents from medicine or any other professional fields. Due to time constraints, we did not fine tune these deep language model on TREC 2019 datasets. Instead we opt for a more simplified solution for re-ranking.

Figure 1 shows the complete query process of our system.

3 Experiment

We submitted nine runs in total, four for scientific abstracts and five for clinical trials. Table 1 and Table 2 show our final evaluation result. The detailed setting used in each runs are listed as follow:

cl_base: Baseline for clinical trials run, use default retrieval setting.

Run	infNDCG	Rprec	P@10
sa_base	0.4441	0.2608	0.5600
sa_base_rr	0.4427	0.2610	0.5600
sa_ft	0.4657	0.2712	0.5650
sa_ft_rr	0.4672	0.2718	0.5675

Table 1: Evaluation results on scientific abstracts.

Run	infNDCG	Rprec	P@10
cl_base	0.5180	0.3862	0.4842
cl_base_rr	0.5321	0.4001	0.5053
cl_ft	0.5089	0.3751	0.4711
cl_ft_rr	0.5351	0.3909	0.4842
cl_ft_agg	0.5355	0.3906	0.4868

Table 2: Evaluation results on clinical trials.

clinic_ft: Clinical trials baseline system with fine-tuned parameters for indexed retrieval system.

cl_ft_agg: Clinical trials baseline system with fine-tuned parameters for retrieval systems and additional synonyms and Hypernyms for query disease.

cl_ft_rr: Additional candidate re-ranker on top of baseline system with fine-tuned parameters for retrieval systems.

clinic_base_rr: Baseline systems with re-rankers op top.

sa_base: Baseline systems for scientific abstracts sub task.

sa_base_rr: Baseline systems for scientific abstracts with candidates re-rankers on top.

sa_ft: Baseline systems for scientific abstracts with fine-tuned parameters for retrieval systems.

sa_ft_rr: Baseline systems for scientific abstracrts with fine-tuned parameters and candidates re-rankers.

The evaluation results of scientific abstracts subtask are shown in Table 1. Baseline system with fine-tuned parameters and candidate re-ranker achieves the best results on all three metrics, indicating the effectiveness of language model based re-ranker.

The evaluation results of clinical trials subtask are shown in Table 2. Run **cl_ft_agg** scores the highest in infNDCG metric, while baseline systems with candidates re-ranker achieves the highest in Rprec and P@10 metrics. Though baseline system with fine tuned parameters performs worse in this subtask, we observe a significant performance boost with candidates re-ranker, which again proves the importance of introducing this stage. This is probabaly due to the fact that articles in clinical trials are relatively longer, which results in more accurate vector representation at documents level. Though we did not pre-train the deep language model using provided data dumps, recent works (Beltagy et al., 2019) show effectiveness of using scientific documents as pre-training corpus. These will be left for future work.

4 Conclusion

In this work we present a multi-pass retrieval system with candidates re-ranker. Experiments show solid performance of our system.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Manda Sai Divya and Shiv Kumar Goyal. 2013. Elasticsearch: An advanced and quick search technique to handle voluminous data. *Compusoft*, 2(6):171.

Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.