Query Expansion Study for Clinical Decision Support

Wenjie Zhuang

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

Weiguo Fan, Chair
Yang Cao
Bert Huang
Eli Tilevich

January 15, 2018
Blacksburg, Virginia

Keywords: Query Expansion, Information Retrieval, Doc2Vec, MeSH Term, Learning to Rank

Query Expansion Study for Clinical Decision Support

Wenjie Zhuang

(ABSTRACT)

Information retrieval is widely used for retrieving relevant information among a variety of data, such as text documents, images, audio and videos. Since the first medical batch retrieval system was developed in mid 1960s, significant research efforts have focused on applying information retrieval to medical data. However, despite the vast developments in medical information retrieval and accompanying technologies, the actual promise of this area remains unfulfilled due to properties of medical data and the huge volume of medical literature.

Specifically, the recall and precision of the selected dataset from the TREC clinical decision support track are low. The overriding objective of this thesis is to improve the performance of information retrieval techniques applied to biomedical text documents. We have focused on improving recall and precision among the top retrieved results. To that end, we have removed redundant words, and then expanded queries by adding MeSH terms in TREC CDS topics. We have also used other external data sources and domain knowledge to implement the expansion. In addition, we have also considered using the doc2vec model to optimize retrieval. Finally, we have applied learning to rank which sorts documents based on relevance and put relevant documents in front of irrelevant documents, so as to return the relevant retrieved data on the top. We have discovered that queries, expanded with external data sources and domain knowledge, perform better than applying the TREC topic information directly.

Query Expansion Study for Clinical Decision Support

Wenjie Zhuang

(GENERAL AUDIENCE ABSTRACT)

Information retrieval is widely used for retrieving relevant information among a variety of data. Since the first medical batch retrieval system was developed in mid 1960s, significant research efforts have focused on applying information retrieval to medical data. However the actual promise of this area remains unfulfilled due to certain properties of medical data and the sheer volume of medical literature. The overriding objective of this thesis is to improve the performance of information retrieval techniques applied to biomedical text documents. This thesis presents several ways to implement query expansion in order to make more efficient retrieval. Then this thesis discusses some approaches to put documents relevant to the queries at the top.

Acknowledgments

During my time as a graduate student at Virginia Tech, I received help from many people. They changed me a lot and made my life colorful. As graduation is coming, let me express my truehearted gratitude to these fantastic people. I wish all of them have a great future.

At the beginning, I want to appreciate my advisor Weiguo Fan. He led me into the area of information retrieval. Each discussion with him is heuristic and impressive. His profound opinions helped me realize the drawbacks of my research procedure and solve the issues in time. Professor Fan gave me a broad space and freedom to try new ideas. He is supportive and encourages me when I had difficulties, no matter for academic issues or otherwise.

I would like to appreciate the help from my committee members: Yang Cao, Bert Huang and Eli Tilevich. They gave me invaluable feedback on my thesis at various stages. Special thanks to professor Tilevich, he provided insightful suggestions on presentation and thesis writing.

Thanks to all faculties who ever taught me in Virginia Tech. They guided me to think questions in multiple aspects and dive deep. Especially thanks go to Ali R. Butt and Na Meng. Professor Butt and professor Meng enriched my understanding about system and software engineering. They also helped me when I was in dilemma. Besides, I desire to thank faculty in ARC (Advance Research Computing). Specially thank James McClure and Nathan Liles. They afforded me a chance to be a research assistant and worked with me to deal with problems together.

Graduate life with my companions in Digital Library Research Lab is joyful. I want to thank the following people: Long Xia, Yufeng Ma, Liuqing Li, Xuan Zhang, Ziqian Song, Yu Wang and Siyu Mi. Further more, I appreciate all my previous teammates, particular thanks to Yue Cheng and Luna Xu. Studying with them is wonderful experience.

I would like to thank all staff in Computer Science for their support and help, particular thanks to Sharon Kinder-Potter. She is always patient and nice when I had questions about graduate requirements.

Last but not least, I want to thank my friends around me for tolerating my defects and accompanying me in the long trip.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Information retrieval(IR) is the process of finding effective information with usually unstructured data relevant to some goals from a large collection of resources. For clinical applications, doctors are often absorbed in making more effective decisions and treatment for patient care. There are various clinical tasks, such as determining the most likely diagnosis of a patient based on known symptoms or making a better treatment plan given existing conditions.

Electronic medical records(EMRs) expand the potential for evidence-based medicine (EBM) improvement of clinical practice greatly. The clinical decision support (CDS) track of Text REtrieval Conference(TREC) investigates techniques aiming for connecting medical cases to biomedical literature relevant for patient care. Clinicians who are seeking critical information for effective treatment, would use TREC-CDS information retrieval systems.

But it's not easy to access to the biomedical literature articles which should be advantageous to obtain answers to the generic clinical questions due to the high volume. It's necessary to find efficient information retrieval ways with notes (2016), descriptions and summaries included in TREC topic sets to search clinical literature.

In TREC 2016 clinical decision support task, realistic electronic health records (EHRs) are used in the task rather than synthetic version of medical case reports. They are notes in TREC topics. The challenge is how to retrieve full-text biomedical articles that address the questions for a given EHR note. Each topic consists of a note, a description and a summary. Jainisha Sankhavara and Prasenjit Majumder [1] used topic modeling on summary, description and note. Better results on summary and description are obtained when compared to note. Hongyu Liu et al [2] expanded the original query with extra data shown in top 10 web pages. Note topics are also applied with KODA, the knowledge drive annotator and MeSH dictionary. They also adopted medical experts' advice in the manual process. Danchen's team [3] extracted medical concepts by MetaMap and used Wikipedia knowledge base to predict the patient diagnosis. They expanded the original query with the predicted

diagnosis. Kuang Lu and Hui Fang [4] defined topic shift of a query and estimated result cut-off/redundancy thresholds.

There are quite a few alternatives for IR libraries e.g. Lucene, Lemur/Indri, Xapian, Terrier, Sphinx with some main-stream options. All of them supply basic IR functionalities. In this thesis, Terrier[5] is chosen as search tool. Terrier is superior in terms of speed and memory. It supports lots of weighting models to assign scores to the retrieved documents. In addition, a collection of Java APIs are available.

1.1 Motivation

A document collection of clinical texts and a set of topics for clinical decision support track are available on TREC website. There are three kinds of topics. They are diagnosis topics, test topics and treatment topics. In each topic, it contains a summary and a description usually. But there are lots of clinical jargon and abbreviations, making the process of searching inefficient. We generate a smaller document collection with all evaluated relevant documents listed by TREC to see the retrieval performance when we apply summaries and descriptions as queries separately. The average performance about recall of top 1000 retrieved documents is low. Precision of top 10 retrieved documents is also small. We detect irrelevant documents are put in front of relevant documents. This thesis is aiming to solve such questions.

After examining the topics, we notice descriptions contain more details than summaries, which means more words in descriptions can be added into queries. In TREC 2016, notes are added into the set of topics. In addition, medical terms have various inflections and synonyms appearing in the collection of clinical texts. These can be found by WordNet. There are also some words non-related with medical issues. It may effect the performance of retrieval. We believe removing redundant words are necessary.

In another side, the interpretations of some drugs and symptoms are important. The performance of retrieval can be improved significantly after adding such interpretations. It's hard to generate these interpretations by natural language processing. NLP is kind of immature in applications like this. Therefore domain knowledge and external data sources, like wiki, are essential.

1.2 Technology Background

1.2.1 Tf-Idf

In information retrieval, tf-idf (term frequency-inverse document frequency) is a metric to reflect the importance of a word to a document in the collection or corpus. It can be

assigned to documents in the collection as a weighting factor for evaluating the relevance of a document to queries. It can also be applied on text classification [6].

One of the basic ranking functions is obtained by getting the sum of tf-idf for each query term. Numerous sophisticated ranking functions are developed under the variants of this simple model. Inverse document frequency (Idf) is usually computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. It reflects how common the term in the corpus. Tf-idf is derived by the product of tf and idf, in case the so common words are assigned high scores by only using tf, like "an".

Various models for generating weighting factors have been created. Some models developed under the probabilistic estimation, while some models relied on the vector space models [7, 8]. Stephen Robertson et al [9] designed a term weighting schema with the probabilistic model and gave a ranking function called Okapi BM25. Another widely used weighting method is called pivoted normalization weighting from the work of Amit Singhal, Chris Buckley, and Mandar Mitra [10].

1.2.2 Word Embedding

Query expansion is for improving information retrieval performance through supplementing an initial query with extra terms. The original query is expanded by adding or replacing words or phrases in the query so that the chances of matching words in relevant documents are increased. Expansion can be performed manually, automatically, or interactively.

Automatic query expansion has been suggested as early as 1960 by Maron and Kuhns [11]. Then it has successively activated and derived various technologies, such as vector feedback [12, 13] and comparative analysis for term distributions [14, 15].

As the volume of data increases, information retrieval communities has focused on the application of deep neural network on complex information retrieval problems. Word embedding representations are dense vector representations of words in low dimensional vector space, where the dimension of the vector space is distinctly lower than the size of the vocabulary of the target resources. Word embedding representations can be directly used as input for neural networks, which accurately capture the semantic relationship among words.

A plenty of techniques for developing word embeddings, like Latent Semantic Analysis (LSA) [16] and probabilistic LSA [7], have been in use for many years. Word2Vec [18] is one of the most popular word embedding models recently. Its output can be easily accepted by deep learning models. Word2vec is basically a computationally efficient predictive model for learning word embedding representations from raw text. It requires a large corpus for training. The purpose of Word2Vec is to locate semantically similar words in close proximity in the vector space. Words similarities are computed mathematically. They are assessed by cosine distance.

Doc2vec [26] modifies the algorithm of word2vec to unsupervised learning of continuous representations for larger blocks of text, such as entire documents. Each document can be represented as a dense vector, regardless of the document length. Doc2vec can be used in finding the similarities among documents, computed by cosine distance.

1.2.3 Learning to Rank

Learning to rank in information retrieval field is applied on constructing a ranking model with training data according to the degrees of relevance, preference or importance, so that new items can be sorted by the generated ranking model. Usually rank top retrieved results after initial weighting models, for example, Okapi BM25 and DLH13 weighting model. There are multiple existing approaches, such as pointwise, pairwise, and listwise approaches. Pointwise [19] approaches verifies a single document at a time and train a classifier to predict how relevant it is for the retrieval need. Pairwise [20, 21, 22] approaches checks a pair of documents at every turn. It focus on considering the relevant order between two documents. Listwise [23, 24] approaches directly take the entire list of documents and seek to get the optimal ordering for the list. There are mainly two ways for listwise approaches, direct optimization of IR measures and minimization of a loss function. Deep learning approaches [25] also emerged in the application of learning to rank.

1.3 Overview and Contributions

We select Terrier as the searching engine when we retrieved data and then compare several query expansion ways. To find similarities of documents to each query, we utilize doc2vec to transform documents into vectors. Similarities are considered to adjust ranking function supported by Terrier. Finally, we make use of learning to rank to put relevant documents retrieved on top locations.

In summary, this thesis makes the following contributions:

1. Apply internet resources to combine additional clinical information, such as domain knowledge, with the information of TREC topics identified by MeSH. This enhances the performance of information retrieval.
   (a) Improve the precision of top 10 retrieved results with TF_IDF model.
   (b) Improve the recall of top 1000 retrieved documents by TF_IDF model. Make a good preparation for learning to rank.
2. Make use of WordNet for generating synonyms of terms in queries.

   (a) Consider the linguistic factor for query expansion. Slightly increase the retrieval performance of some queries, but find it is not a good way in general.

   (b) Utilize word2vec to improve the accuracy of WordNet.

3. Exploit doc2vec for ranking adjustment.

   (a) Provide an idea of modifying ranking function of TF_IDF by doc2vec similarities. It raises information retrieval performance a bit.

   (b) Explore the application of doc2vec on training learning to rank model. According to the overall performance of learning to rank model, we have discovered that it is ineffective to catch the relationship between a query and a document by doc2vec.

Chapter 2

System Design & Methodology

We create a dataset for all 90 queries from 2014 to 2016. The original data on TREC clinical decision support track is more than 1 million, in order to find the effective ways to improve retrieval performance, we generate a smaller dataset containing all evaluated documents from TREC 2014 to 2016. It contains 12294 documents. The number of evaluated relevant documents is generally smaller than 600. In many cases it's not beyond 100. Experiment results of the smaller dataset will provide hints for future work on the whole dataset.

We focus on addressing IR query processing efficiency by query expansion. The initial queries are basically generated by summaries in TREC topics. However there are two sets of TREC topics in 2015 due to the existence of two tasks. This thesis takes advantage of the set of topics for task B. The process of choosing initial queries will be discussed in the next chapter. Beyond that, we implement expansion with internet. It is so called web augmentation. After that, we convert words in queries to different parts of speech(POS) and add symptoms with WordNet. Terrier is chosen as search engine. The TF_IDF model of Terrier is utilized. The compared results are shown in case study section.

Given each TREC topic, we notice there exists some redundant words in the topic. To selected terms from descriptions, we refer to an external source MeSH (Medical Subject Headings). MeSH is a comprehensive controlled vocabulary as a thesaurus that boosts searching. It provides an interface called MeSH on Demand to identify MeSH terms in the submitted text. We add MeSH terms from descriptions into original queries instead of all words of descriptions. The interpretation of jargon in the queries can also be found on MeSH mostly. For 2016 topic, we do the same process for notes. Then import other external internet data sources and domain knowledge to modify these queries further. We utilize the symptoms, drugs, diseases and other clinical information in the queries to find critical information through google. The returned online results are used to adjust queries, such as diagnosis, drug usages, potential causes, test and treatment. We compare the related retrieval results to the results of initial queries.

WordNet [27] is a lexical database for English which groups English words into sets of synonyms as synsets and keeps a number of relations among the synonym sets or their related members. It includes the lexical categories nouns, verbs, adjectives and adverbs. We obtain the synsets of words in original queries. Besides, we convert the properties of words, for example, transforming a noun to an adjective. We adjust queries by introducing relevant synonyms and adding converted word forms.

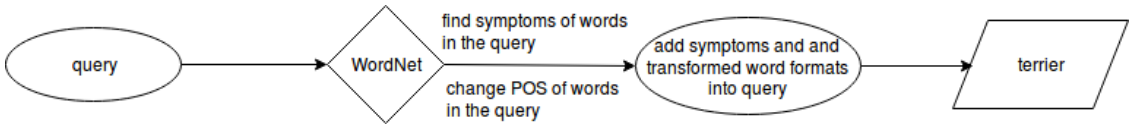

Figure 2.1: Flow of expansion with WordNet

The similarities among documents are also considered. We utilize doc2vec to rank the similarities of document to each query. Then modify retrieval results of Terrier with doc2vec model. Related retrieval results are contrasted.

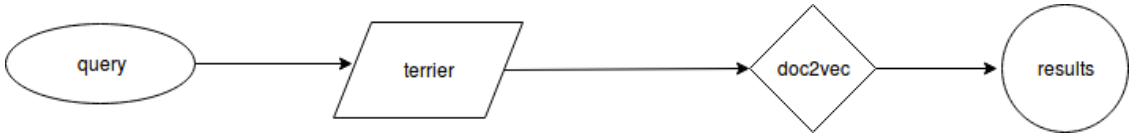

Figure 2.2: Application of doc2vec

Finally the retrieval performance of queries expanded with MeSH terms, domain knowledge and internet sources is better.

In order to ensure relevant documents returned first, learning to rank is introduced. According to the previous doc2vec model, we have documents and queries vectors. We apply these vectors to train and test learning to rank model. Beyond that, we report the performance analysis of learning to rank.

Chapter 3

Case Study & Experiments

This chapter is split into 2 parts. The first part is mainly to find the effective way to expand query. The second part utilizes learning to rank to put relevant retrieved results in front of irrelevant results.

3.1 Query and Ranking Function Modification

The sets of topics of TREC clinical decision support track in 2014 to 2016 are different, therefore we split the results into three parts according to years. We evaluate performance by recall@1000 and p@10. Recall@1000 is for recall in top 1000 retrieved results, while p@10 means precision in top 10 retrieved data.

As mentioned in the system design chapter, TF_IDF model is applied to compute the scores of documents for each query. TF is short for term frequency in a document, while IDF is an abbreviation of Inverse Document Frequency. They can be computed by the following equations

$$TF_{t,d} = \frac{Number\ of\ frequency\ of\ term\ t\ in\ a\ document\ d}{Number\ of\ frequency\ of\ term\ t\ in\ a\ document\ d + k},\ wherek = 1. \tag{3.1}$$

$$IDF_t = \ln \frac{Number\ of\ documents}{Number\ of\ documents\ contains\ term\ t}. \tag{3.2}$$

The weight of term t in document d is assigned as

$$TF\_IDF_{t,d} = TF_{t,d} * IDF_t. \tag{3.3}$$

In Terrier, TF is Robertson's TF and IDF is the standard Sparck Jones' IDF. The score of a document d for a query q is obtained by

$$score_{q,d} = \sum_{t \in q} TF\_IDF_{t,d}. \tag{3.4}$$

Our goal is to increase the scores of relevant documents for each query through query expansion. We followed the methodology mentioned in the previous chapter. In order to show the selection of initial queries, we mainly compare the performance of summaries and descriptions. For 2016, performance of notes is also compared. For 2015, the comparison process is a little complex. It will be discussed later. The results are in topic information section. The second part relies on the returned information from internet. The third part is WordNet. It modifies the queries with better performance after comparison in the second part. In this part, terms in the queries are convert to multiple possible word formats. Then add these word formats into original queries to generate new queries. Besides, we obtain the synonyms of terms in the original queries by WordNet and import them into original queries. We also combine symptoms and word formats with original queries. Finally we list the compared results. The last part is doc2vec, used for computing new scores of retrieved results from better performed queries, thus the relevant documents can get higher scores.

3.1.1 TREC CDS 2016

In TREC CDS 2016, a note from MIMIC-III(a freely accessible critical care database) is added into each topic. The note contains a patients medical history, current complaint, current possible diagnosis, tests and treatments. It is actual medical case report with jargon and acronyms. Thus there are three parts in each topic, a note, a description and a summary. In the description, less jargon is kept. The summary is a summarization of the description, as previous two years.

We first apply summaries as queries. Table 3.1 is the retrieval result.

Expansion with description and note

We first contrast the performance of summaries, descriptions and notes when they are selected as queries.

Figure 3.1 shows the comparison result about top 1000 recall performance. It's significant that the queries with summaries work best in most cases. We compute the average value of recall for each kind of retrieval results, the average performance of summaries is highest in table 3.2. But in some cases, like query 15, description performs best. In query 21, the performance of expansion with note is best. Results of precision about top 10 retrieval data in figure 3.2 are similar.

We consider using summaries as initial queries, then expand queries with words of descriptions or notes. In figure 3.3, we compare 4 results about top 1000 retrieval recall metric. It's easy to see the original queries with summaries work best in most cases. We compute the average recall for each kind of retrieval results in table 3.3. The average performance of summaries is most significant. But in some cases, like query 4, adding description performs

Table 3.1: 2016 search with summaries

query	recall@1000	recall@1000 percentage	p@10	p@10 percentage
1	45	0.352	5	0.5
2	33	0.971	4	0.4
3	49	0.322	1	0.1
4	6	0.333	0	0
5	44	0.449	3	0.3
6	73	0.518	4	0.4
7	38	0.494	1	0.1
8	346	0.405	10	1
9	88	0.715	4	0.4
10	11	0.579	1	0.1
11	128	0.349	4	0.4
12	54	0.486	9	0.9
13	71	0.467	7	0.7
14	42	0.350	2	0.2
15	44	0.571	5	0.5
16	55	0.299	6	0.6
17	129	0.729	5	0.5
18	24	0.353	0	0
19	101	0.455	5	0.5
20	266	0.404	9	0.9
21	20	0.274	0	0
22	3	0.375	0	0
23	56	0.514	5	0.5
24	364	0.469	9	0.9
25	174	0.795	2	0.2
26	76	0.673	5	0.5
27	10	0.769	1	0.1
28	52	0.245	1	0.1
29	104	0.881	1	0.1
30	18	0.450	6	0.6

Table 3.2: 2016 average performance comparison among summary, description and note

metric	summary	description	note
recall	0.502	0.410	0.339
precision	0.383	0.197	0.170

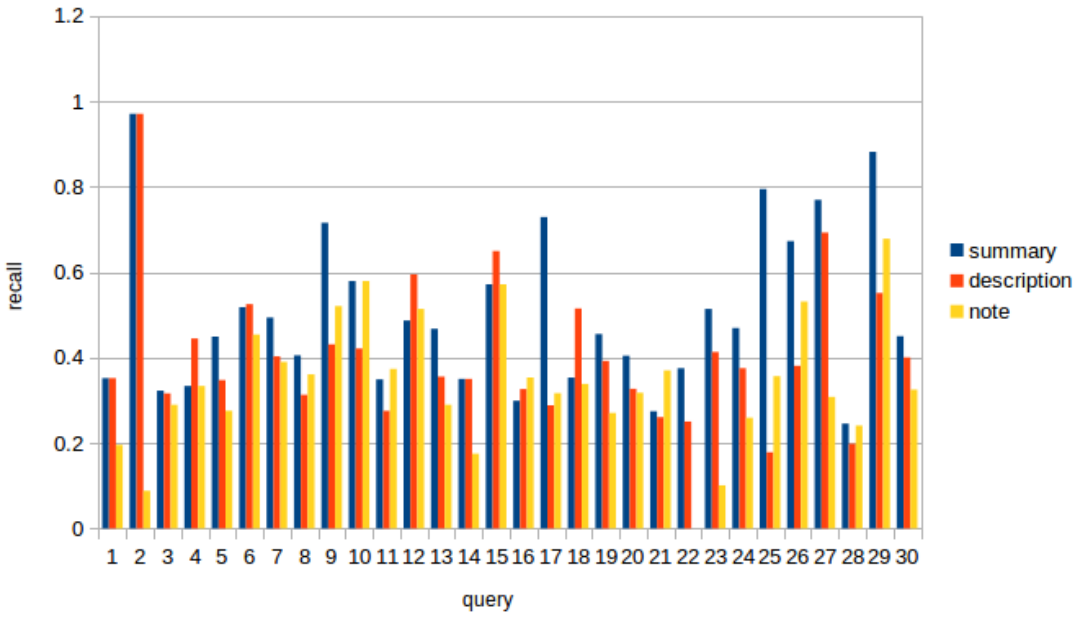

Figure 3.1: 2016 performance comparison among summary, description and note-recall

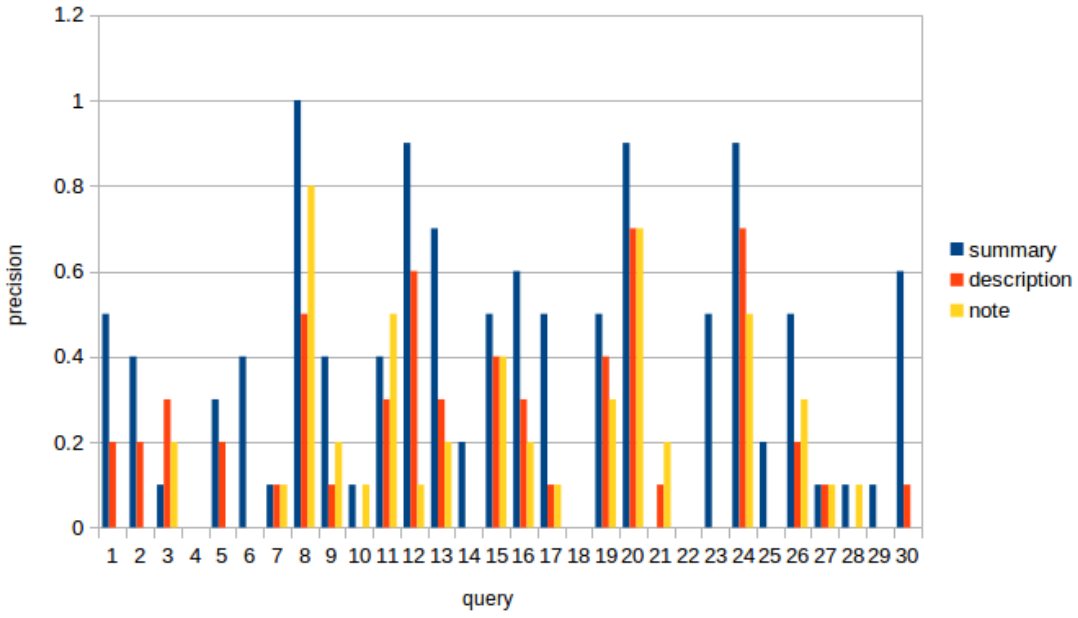

Figure 3.2: 2016 performance comparison among summary, description and note-precision

best. In query 10, the performance of expansion with note is best.

Table 3.3: 2016 average performance about expansion with data of topics

metric	summary	expansion with description	expansion with note	expansion with both
recall	0.502	0.416	0.365	0.349
precision	0.383	0.217	0.160	0.160

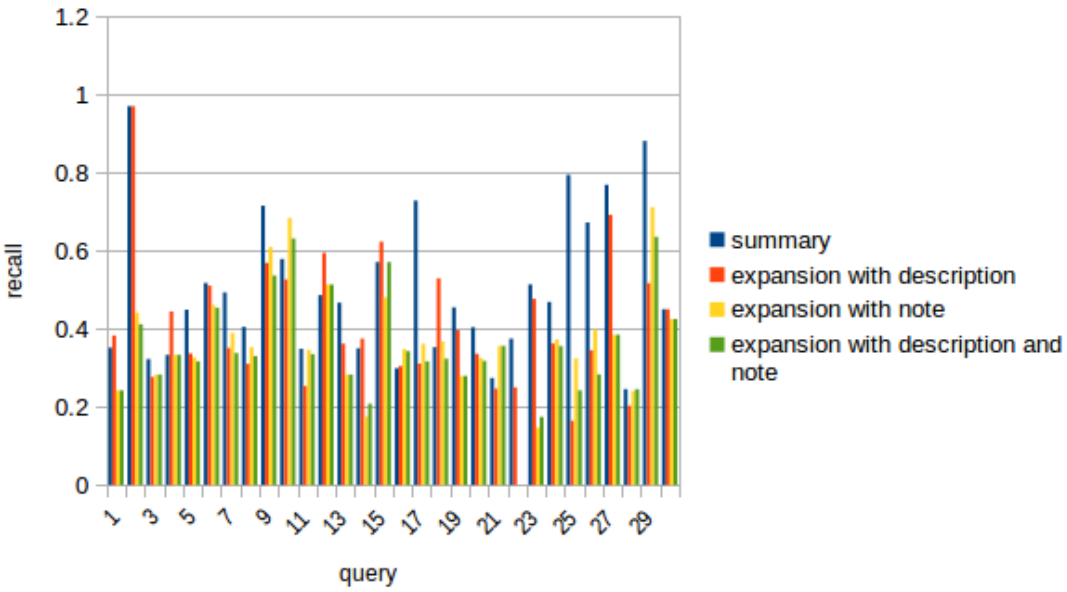

Figure 3.3: 2016 expansion with data of topics-recall

Precision of top 10 retrieval results is also considered. The simulation is similar in figure 3.4. More queries caught 0 relevant document after expansion.

Overall summaries are applied as baselines.

Web Augmentation

We realize some words of the topics are not for medical issues, thus we remove these words according to medical corpus. Further, we utilize MeSH to identify keywords in the topics. We combine medical terms in the summaries with MeSH terms in the descriptions/notes to generate new queries. Since MeSH also shows the hierarchy of MeSH terms, such information is also taken into consideration.

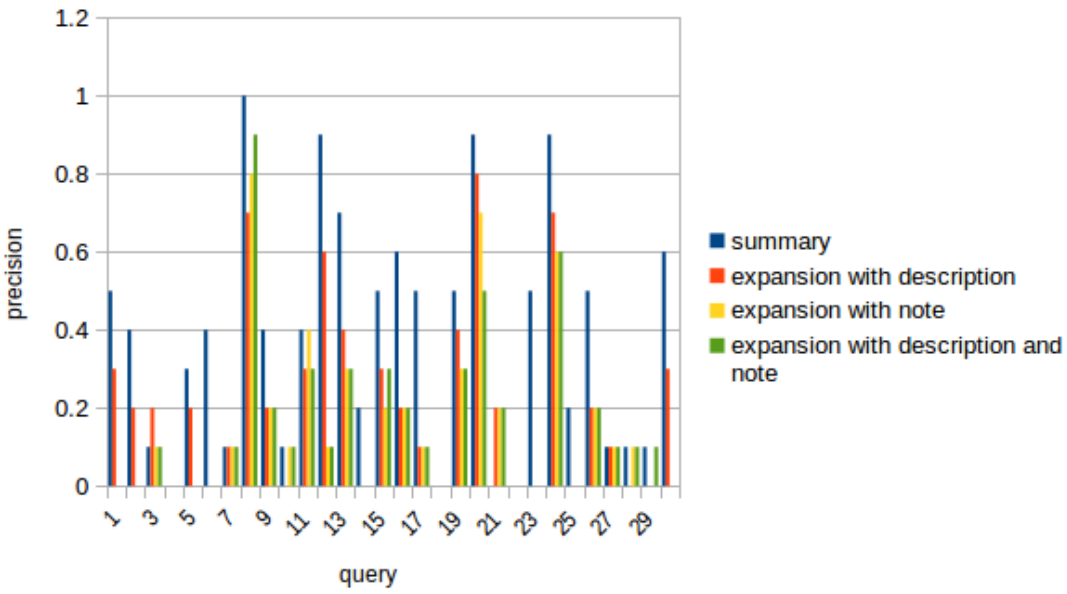

Figure 3.4: 2016 expansion with data of topics-precision

At the same time, we notice the retrieval performance could be improved if clinical analysis is imported. For example, in the query 1, there is a word "melena". After adding the casues of "melena", which is gastrointestinal bleeding, the retrieval performance is almost doubled. We make use of the keywords in queries to obtain domain knowledge through google, like symptoms explanations, drug usage, possible diseases, meaning of jargon and other information. We add these into queries.

Figure 3.5 shows comparison of recall between initial queries and query expansions. The improvement of performance is significant. Figure 3.6 shows comparison about precision. After expansion, there are more queries which get relevant data in top 10 retrieved results.

Table 3.4: 2016 average performance comparison between initial query and expansion with internet

metric	initial query	expansion with internet	improved
recall	0.502	0.854	70.1%
precision	0.383	0.483	26.1%

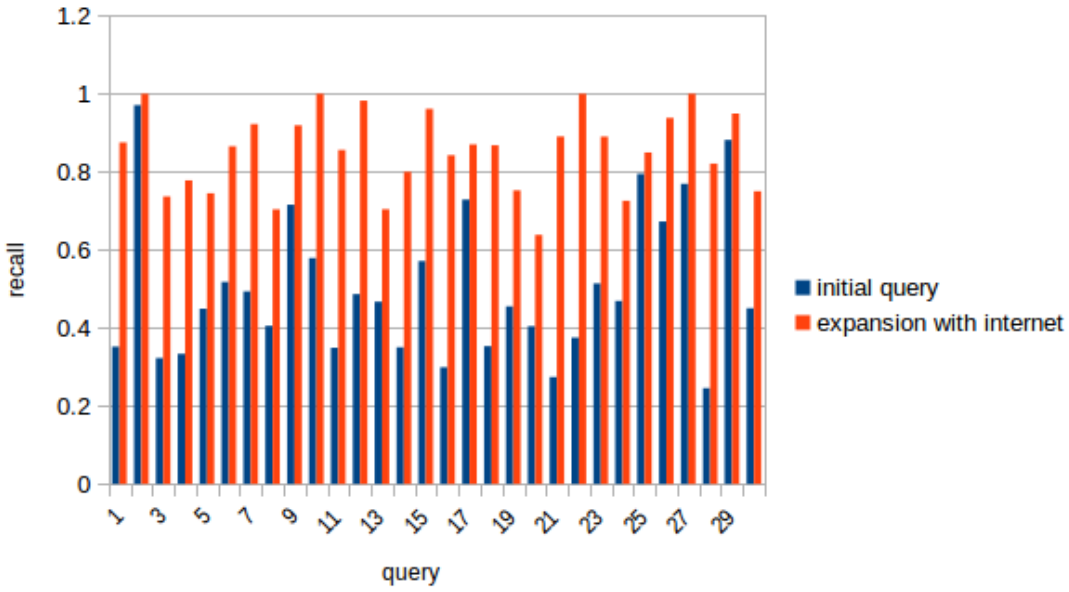

Figure 3.5: 2016 performance comparison between initial query and expansion with internet-recall

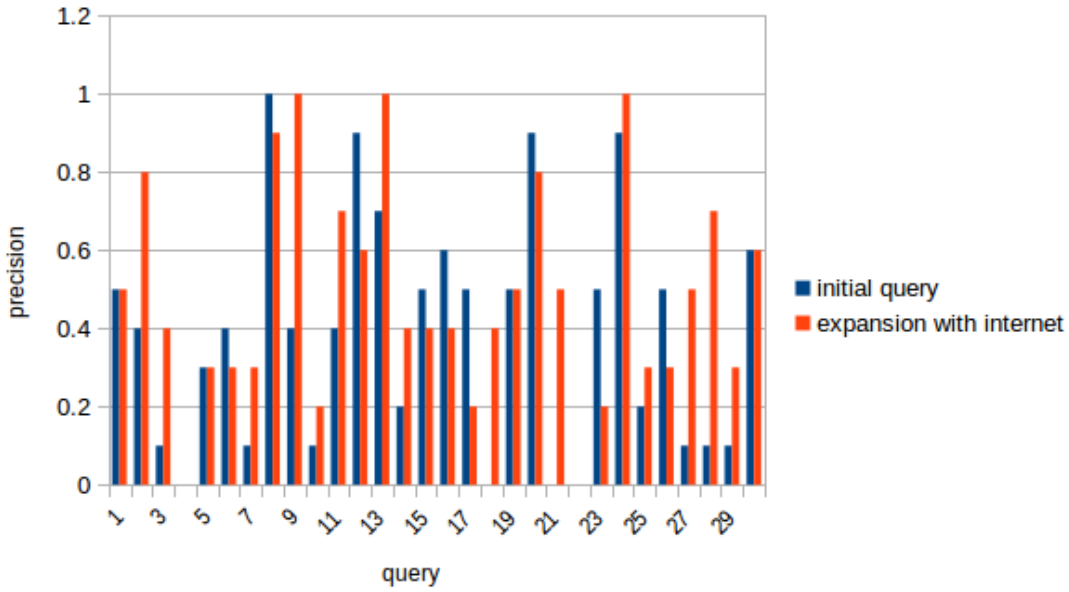

Figure 3.6: 2016 performance comparison between initial query and expansion with internet-precision

WordNet

We utilize WordNet to transform words of queries. For example, we convert "sclerosis" to "sclerotic", from noun to adjective. Besides, we obtain synonyms by WordNet. Since queries expanded by external online data sources and domain knowledge perform much better than initial queries, we consider adding more word formats and synonyms of words in the expanded queries.

We collect retrieval results when add more word formats and synonyms respectively or add formats and synonyms one time. We first compute recall metric among top 1000 retrieved results. In figure 3.8, it's easy to see results of just importing more formats of words are almost highest. We discover WordNet selected more synonyms than necessary due to polysemy. Word2vec is chosen to solve this problem. Synonyms which are less similar to original words based on word2vec model are removed. After modification, we can find the performance is improved and near to the performance of just adding multiple word formats generally.

Figure 3.7 and 3.8 show that queries expanded by external online data sources and domain knowledge perform best.

About precision among top 10 retrieved documents, the comparison process is similar to previous step. When we import synonyms from WordNet directly or import all synonyms and different word formats together, there is no relevant document in the retrieved results for query 2. After adjustment with word2vec, the problem is solved. Moreover, the average performance of adding word formats and more similar synonyms is slightly better than only adding multiple word formats.

Table 3.5: 2016 expansion average performance with WordNet part 1

metric	initial query	expansion with internet	expansion with word formats
recall	0.502	0.854	0.808
precision	0.383	0.483	0.443

Table 3.6: 2016 expansion average performance with WordNet part 2

metric	with word formats	with synonyms	with synonyms and word formats	with more similar synonyms and word formats
recall	0.808	0.750	0.742	0.809
precision	0.443	0.370	0.357	0.450

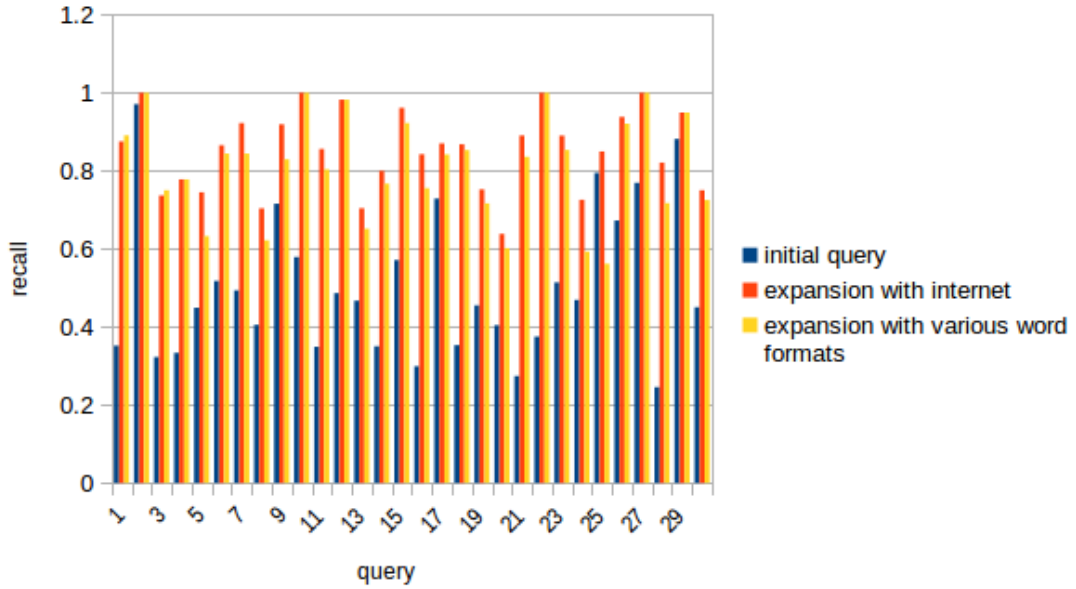

Figure 3.7: 2016 expansion performance with WordNet-recall part 1

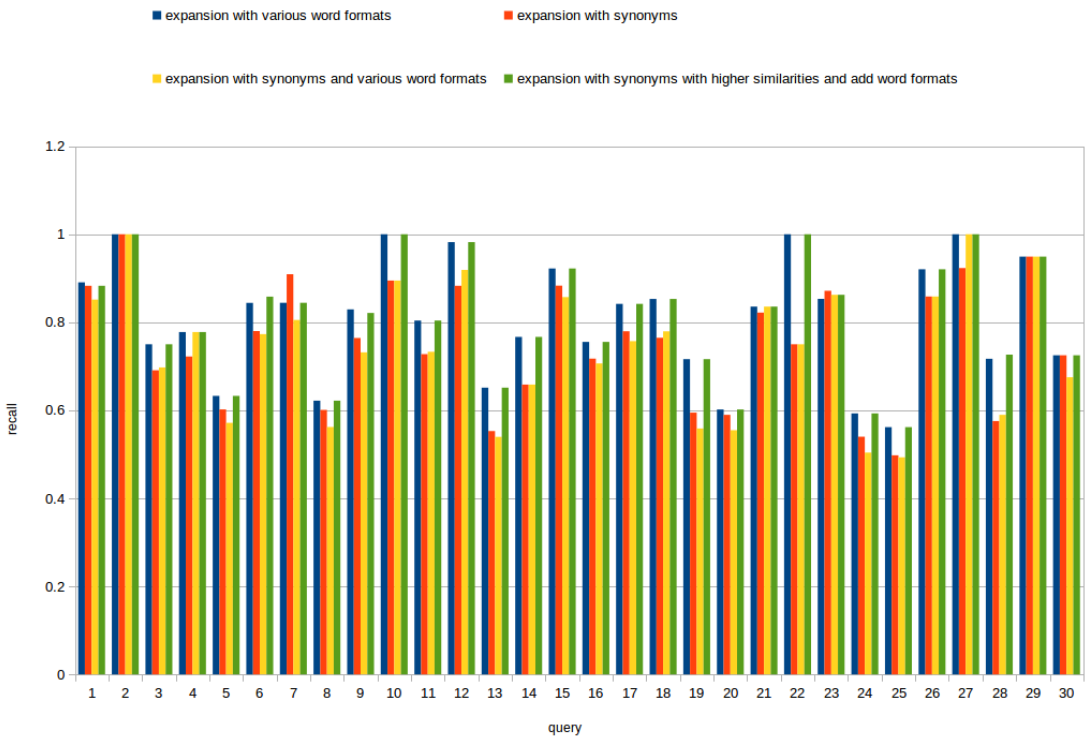

Figure 3.8: 2016 expansion performance with WordNet-recall part 2

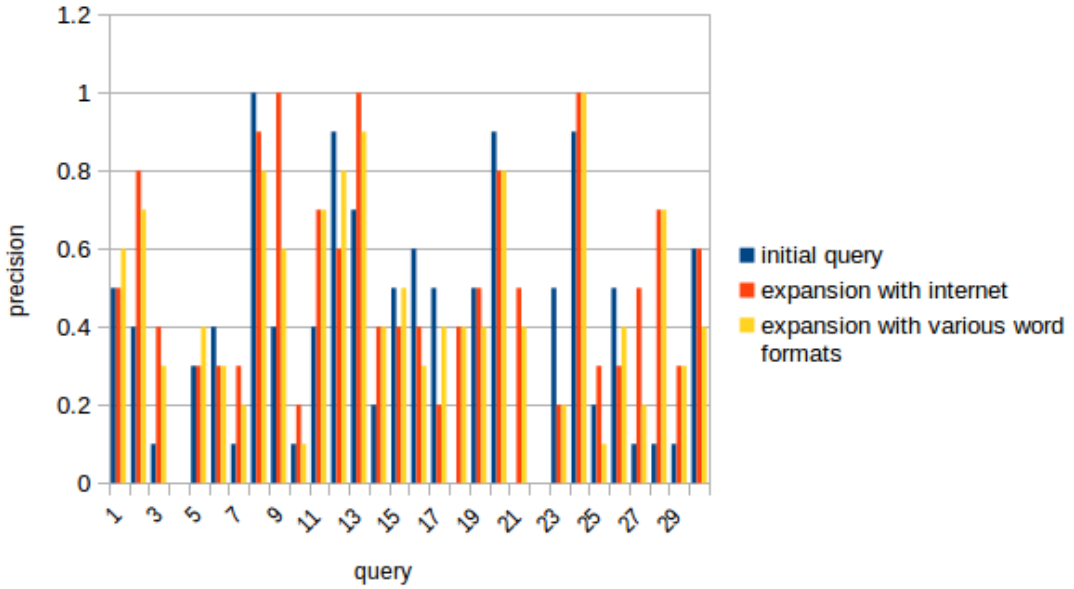

Figure 3.9: 2016 expansion performance comparison with WordNet-precision part 1

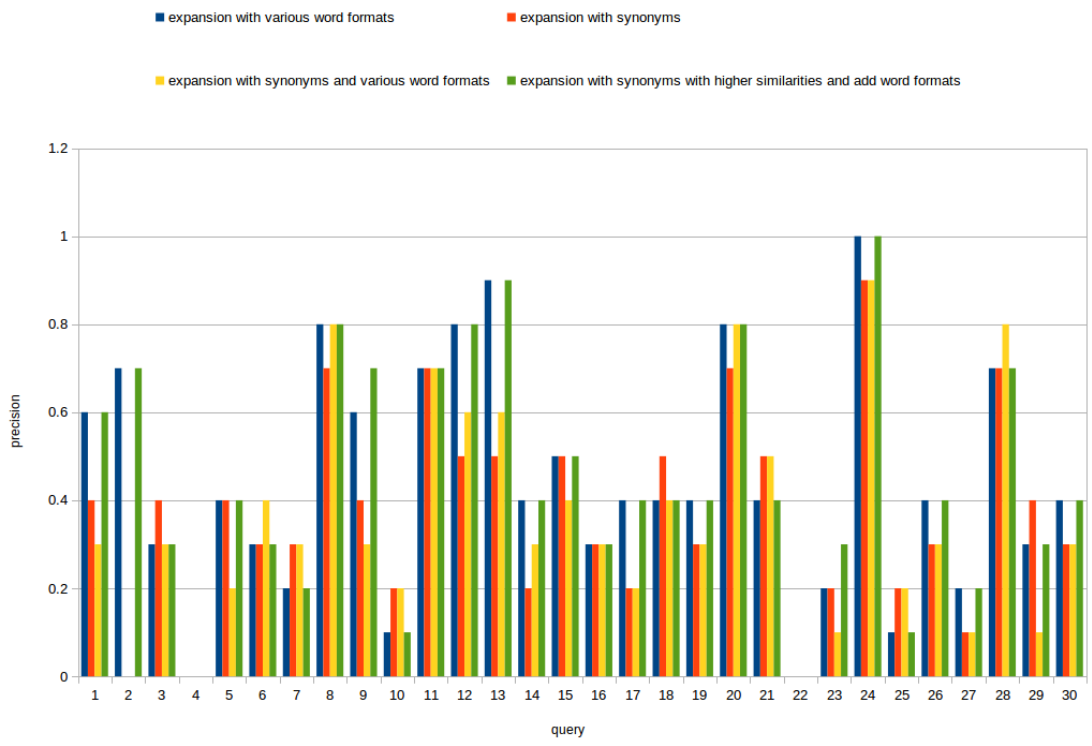

Figure 3.10: 2016 expansion performance comparison-precision with WordNet part 2

Doc2vec

From the previous section, queries expanded by external online data sources and domain knowledge perform best. Since we use Terrier TF_IDF model to rank the relevance of documents, we think about combining the document similarities to expanded queries obtained by doc2vec with TF_IDF weights to rerank the relevance of documents. The reranking equation is

$$new\_score = TF\_IDF\_score * doc2vec\_similarity. \tag{3.5}$$

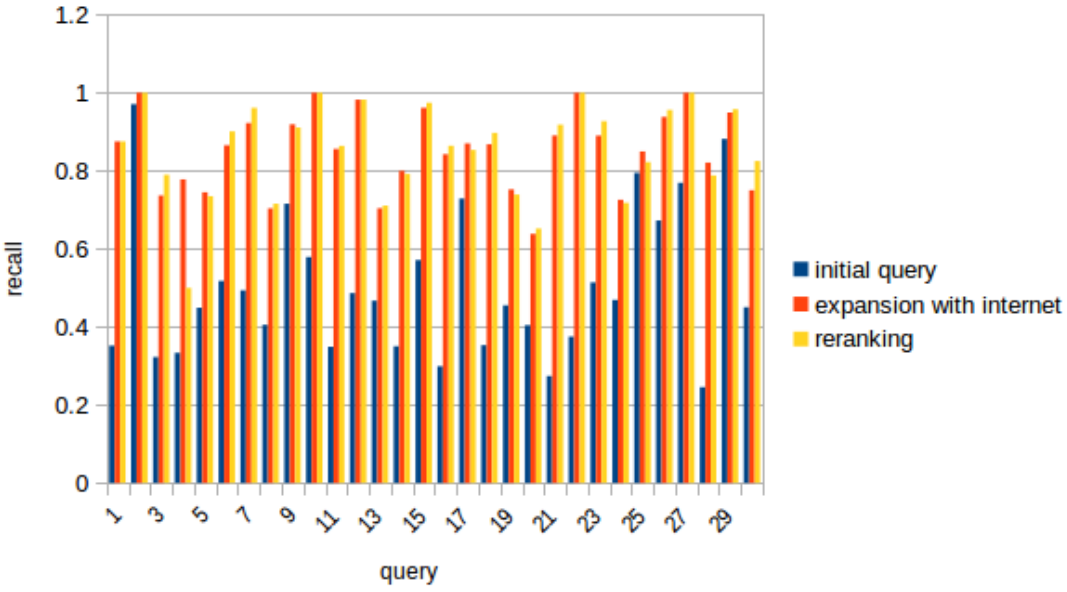

Figure 3.11: 2016 reranking comparison-recall

Table 3.7: 2016 reranking average performance

metric	initial query	expansion with internet	reranking	improved
recall	0.502	0.854	0.854	0
precision	0.383	0.483	0.483	0

In figure 3.11, we discern after reranking the results retrieved by TF_IDF model with expanded queries, the new ranking results increase recall rates in top 1000 documents slightly in a half of queries, while the recall of around 1/3 queries decrease slightly. On the other hand, the average precision performance keeps same after adjustment. It shows reranking make precision of some queries increased and some decreased in figure 3.12.

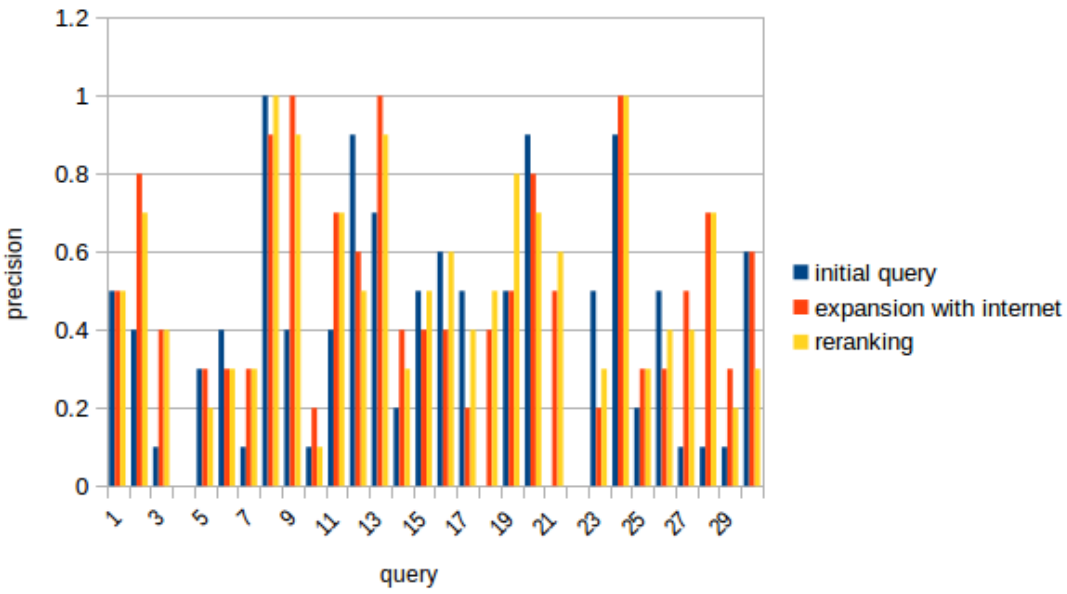

Figure 3.12: 2016 reranking comparison-precision

3.1.2 TREC CDS 2015

There are two TREC tasks in 2015, i.e., task A and task B. Different from 2016, there are two parts in each topic of task A, a description and a summary. Task A and task B are almost same, but task B contains the diagnosis results for the treatment topics and test topics. Other contents of task B are totally same as task A. In the description, there may be some jargon. The summary is a concise version of the description as previous.

We search summaries at first. Table 3.8 is the retrieval result.

Expansion with description

Since both descriptions and summaries are shown in all topics of task A and task B, we first contrast the performance of summaries and descriptions when they are selected as queries separately.

Figure 3.13 shows the comparison result about top 1000 recall performance. The queries with summaries work better than descriptions in 17 cases. Results of precision about top 10 retrieval data are shown in figure 3.14. The performance of summaries is significant. In the majority of queries, precision of summaries is higher than descriptions.

Nextwe combine summaries with descriptions as new queries. In figure 3.13, we can see the original queries with summaries work slightly better than descriptions and combinations. We

Table 3.8: 2015 search with summaries

query	recall@1000	recall@1000 percentage	p@10	p@10 percentage
1	39	0.188	3	0.3
2	8	0.267	2	0.2
3	93	0.260	7	0.7
4	190	0.607	6	0.6
5	18	0.667	2	0.2
6	23	0.307	4	0.4
7	47	0.313	5	0.5
8	123	0.961	9	0.9
9	25	0.532	4	0.4
10	38	0.469	2	0.2
11	30	0.135	5	0.5
12	7	0.226	1	0.1
13	67	0.390	7	0.7
14	16	0.372	6	0.6
15	113	0.309	3	0.3
16	273	0.399	8	0.8
17	238	0.702	10	1.0
18	6	0.0444	0	0
19	24	0.273	1	0.1
20	12	0.156	1	0.1
21	87	0.604	8	0.8
22	250	0.392	10	1.0
23	52	0.477	5	0.5
24	7	0.117	0	0
25	2	0.125	0	0
26	122	0.792	8	0.8
27	25	0.625	4	0.4
28	9	0.170	0	0
29	30	0.405	6	0.6
30	83	0.643	8	0.8

Table 3.9: 2015 average performance comparison among summary, description and combination

metric	summary	description	combination
recall	0.398	0.363	0.386
precision	0.450	0.303	0.323

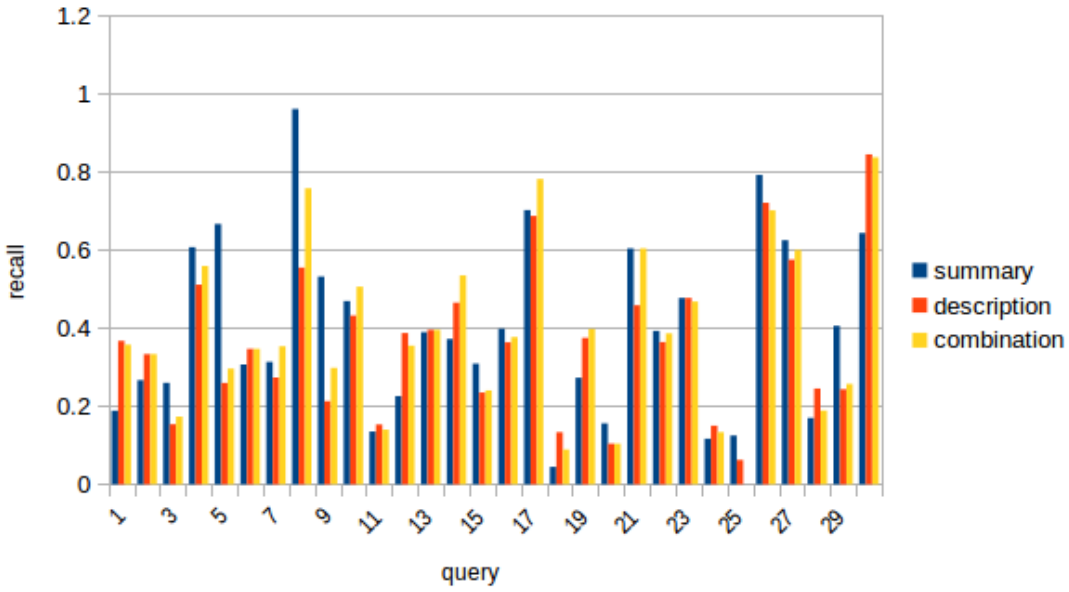

Figure 3.13: 2015 performance comparison among summary, description and combination-recall

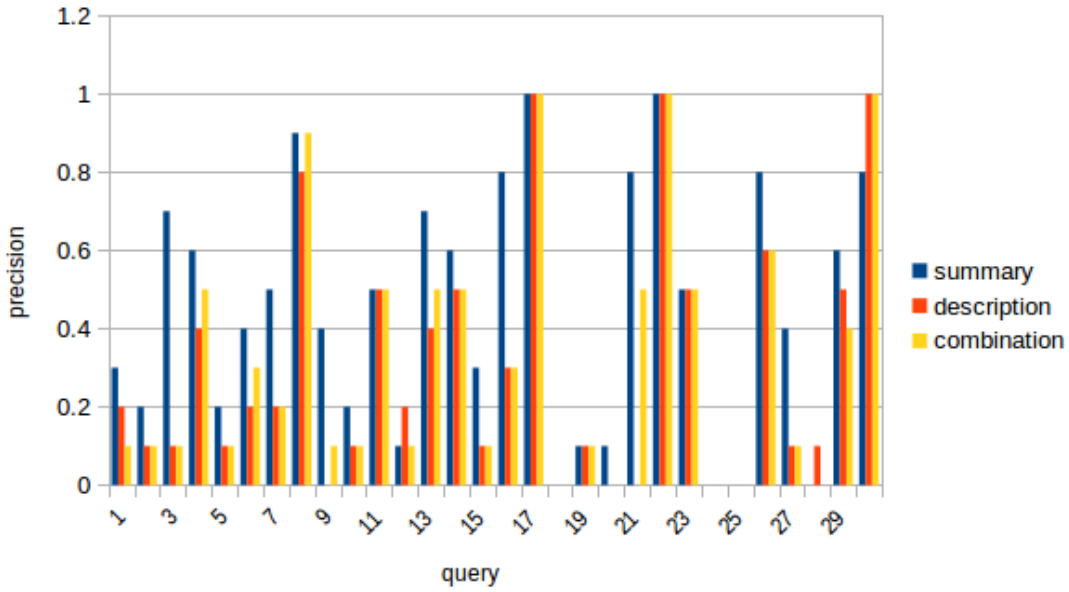

Figure 3.14: 2015 performance comparison among summary, description and combination-precision

compute the average recall for these 3 kinds of queries in table 3.9. The average performance of summaries is highest, but near to the average performance of the combination of summaries and descriptions.

Precision of top 10 retrieval results is also listed in figure 3.14. More queries caught 0 relevant document after expansion.

However diagnosis is added for each test and treatment topic in 2015 task B, so we exploit the combination of summaries and diagnoses if they exist. For diagnosis type topics, just used summaries. There are still 30 topics in total, but 10 topics of diagnosis type.

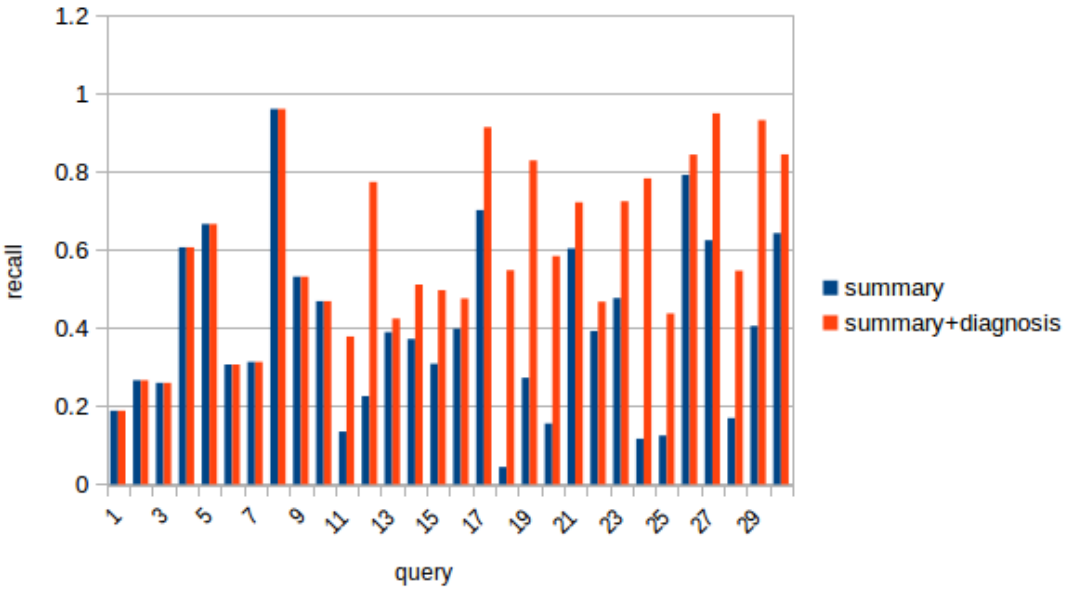

Figure 3.15: 2015 performance comparison between summary and mixture of summary and diagnosis-recall

From topic 11 to topic 30, no matter recall or precision is increased in most cases. These topics are test type or treatment type with diagnosis information in task B, while the left topics are without extra diagnosis of diseases. Thus the recall and precision of the first ten topics keep same.

Summaries are decided as baselines for the first 10 queries, but we select the mixture of summaries and diagnoses as baselines for the remaining topics. For recall, the average performance of new baselines is 0.592, while it is 0.398 in table 3.9 if we only set summaries as queries. For precision, the average performance of new baselines is 0.540, but purely using summaries as queries is 0.450.

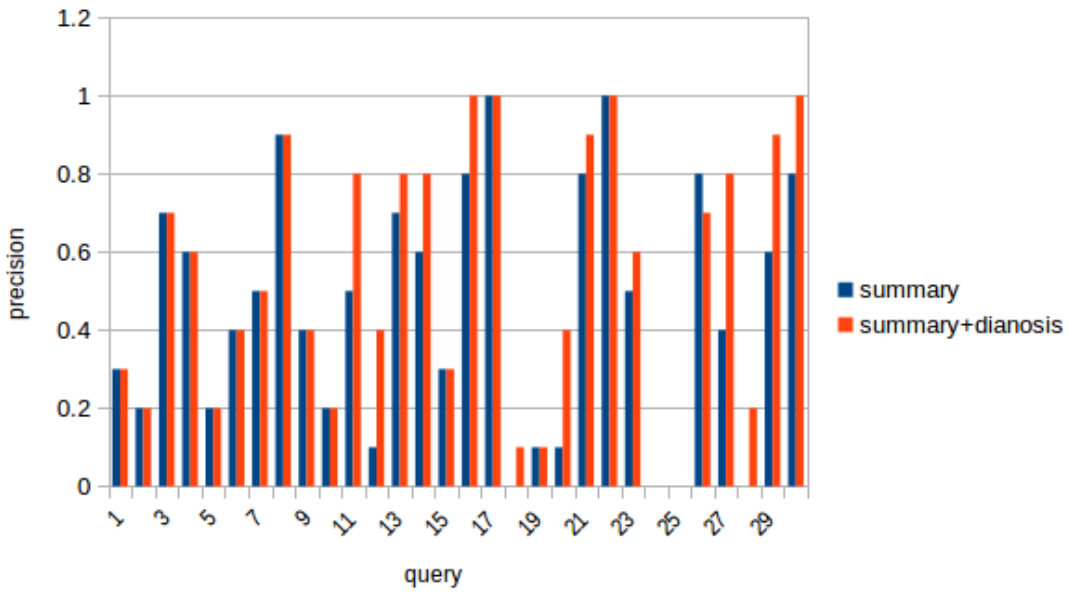

Figure 3.16: 2015 performance comparison between summary and mixture of summary and diagnosis-precision

Web Augmentation

After setting new baselines, we still first remove noisy words according to medical corpus and utilize MeSH to identify keywords in the topics. We complement medical terms in the topics into queries. In addition, we replenish domain knowledge and information obtained from internet to each initial query as what we did in 2016.

Figure 3.17 illustrates the comparison result between initial queries and query expansions for recall. Performance is improved significantly. About precision, figure 3.18 displays almost all queries get relevant data in top 10 retrieved results after comparison. The precision in most queries increases.

Table 3.10: 2015 average performance comparison between initial query and expansion with internet

metric	initial query	expansion with internet	improved
recall	0.592	0.850	43.6%
precision	0.540	0.623	15.4%

In addition, we contrast the performance of extension with internet and mixture of summaries and diagnoses for topic 11 to topic 30 in figure 3.19 and 3.20. Expansion with internet is

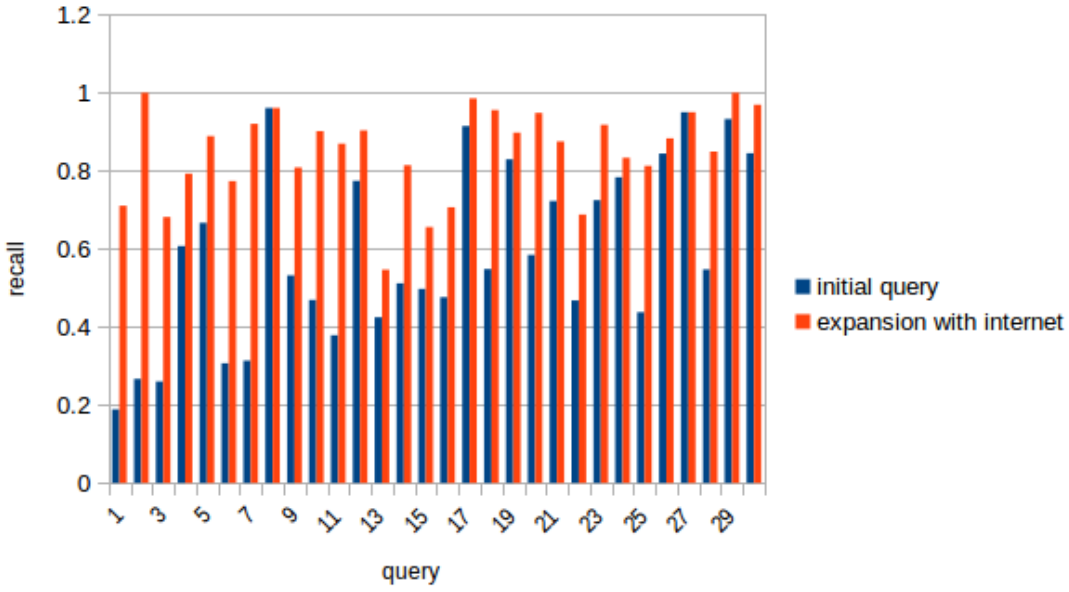

Figure 3.17: 2015 performance comparison between initial query and expansion-recall

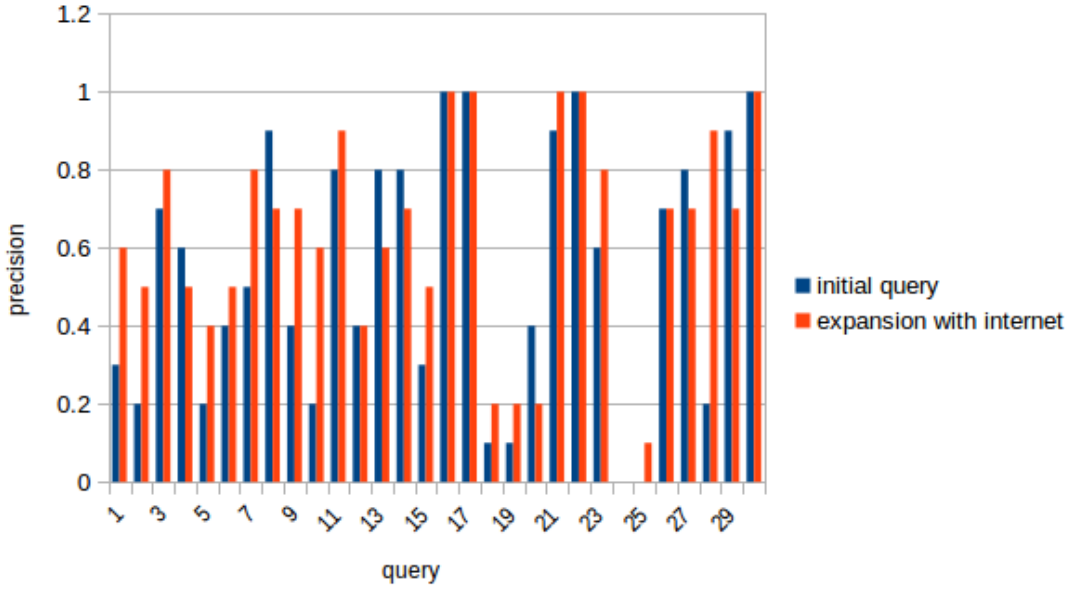

Figure 3.18: 2015 performance comparison between initial query and expansion-precision

significantly superior for the average value of recall. The average value of precision is also better than mixture of summaries and diagnosis listed in table 3.11. Therefore the overall performance of expansion with internet is higher.

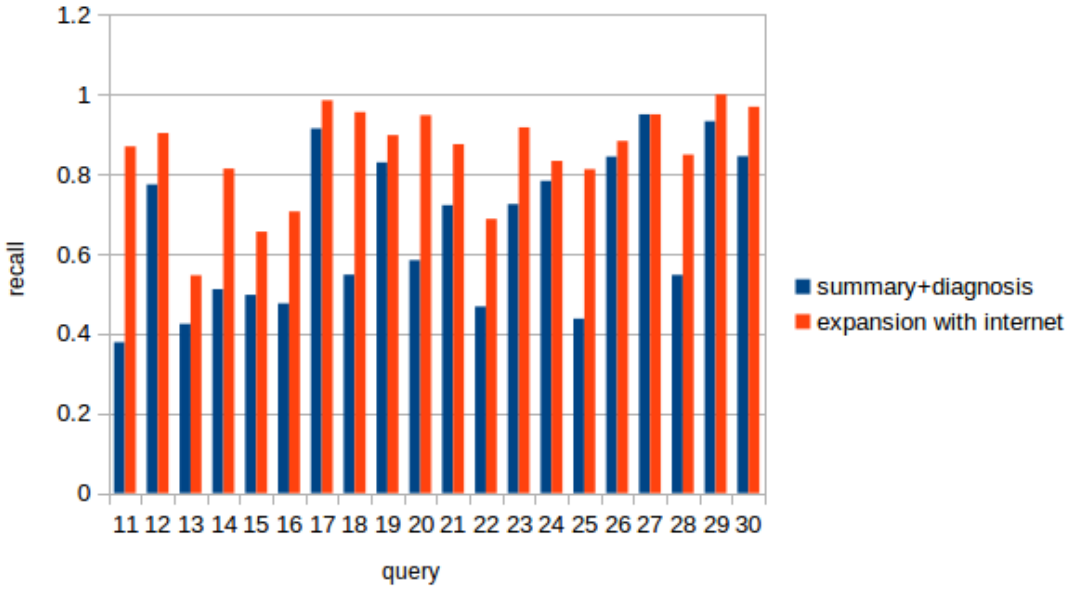

Figure 3.19: 2015 performance comparison between mixture and expansion-recall

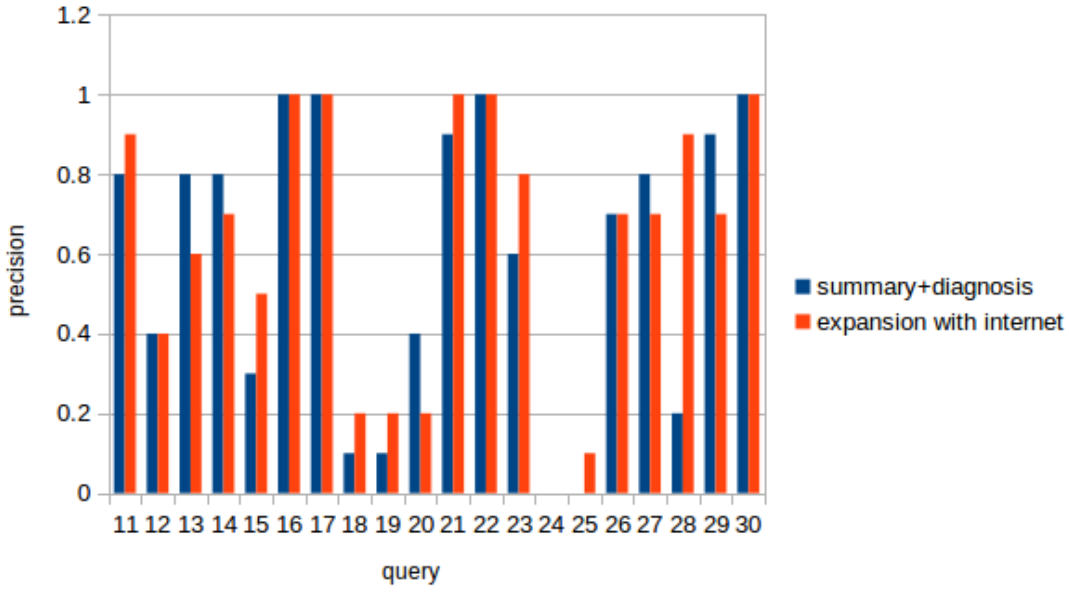

Figure 3.20: 2015 performance comparison between misture and expansion-precision

Table 3.11: 2015 average performance comparison between mixture and expansion with internet

metric	summary + diagnosis	expansion with internet	improved
recall	0.660	0.853	29.2%
precision	0.590	0.630	6.3%

WordNet

Same as the previous discussion for experiments in 2016, to modify previous expanded queries further, we think over importing multiple word formats and synonyms of words in the queries. We record retrieval results when put in more word formats and synonyms respectively or subjoin formats and synonyms one time.

We first discuss recall performance among top 1000 retrieved results. In figure 3.22, it's easy to see results of just importing more formats of words are almost highest. After removing synonyms which are less similar to original words based on word2vec model, we can see the performance is improved and near to the performance of just adding multiple word formats generally.

Figure 3.21 and 3.22 illustrate that queries expanded by external online data sources and domain knowledge perform best.

About precision among top 10 retrieved documents, the comparison process is similar to previous step. Only when we concatenate all synonyms and different word formats together, all queries have relevant retrieved documents in top 10 returned results. After adjustment with word2vec, the average performance of subjoining word formats and more similar synonyms is slightly better than just subjoining multiple word formats and better than other expansion ways except expanding just with domain knowledge and external online data sources.

Table 3.12: 2015 expansion average performance with WordNet part 1

metric	initial query	expansion with internet	expansion with word formats
recall	0.592	0.850	0.813
precision	0.540	0.623	0.593

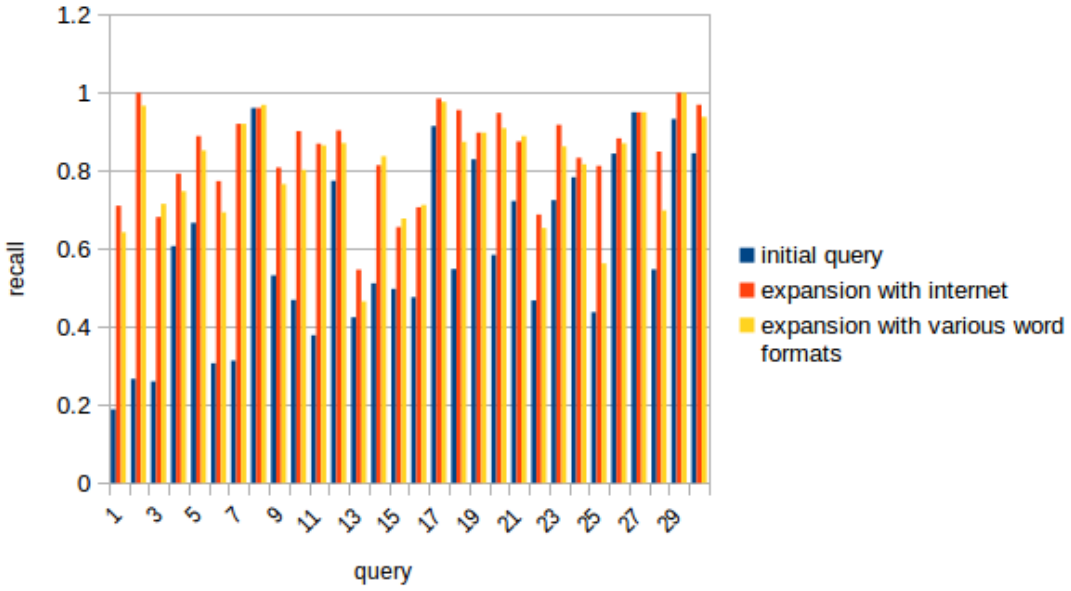

Figure 3.21: 2015 expansion performance with WordNet-recall part 1

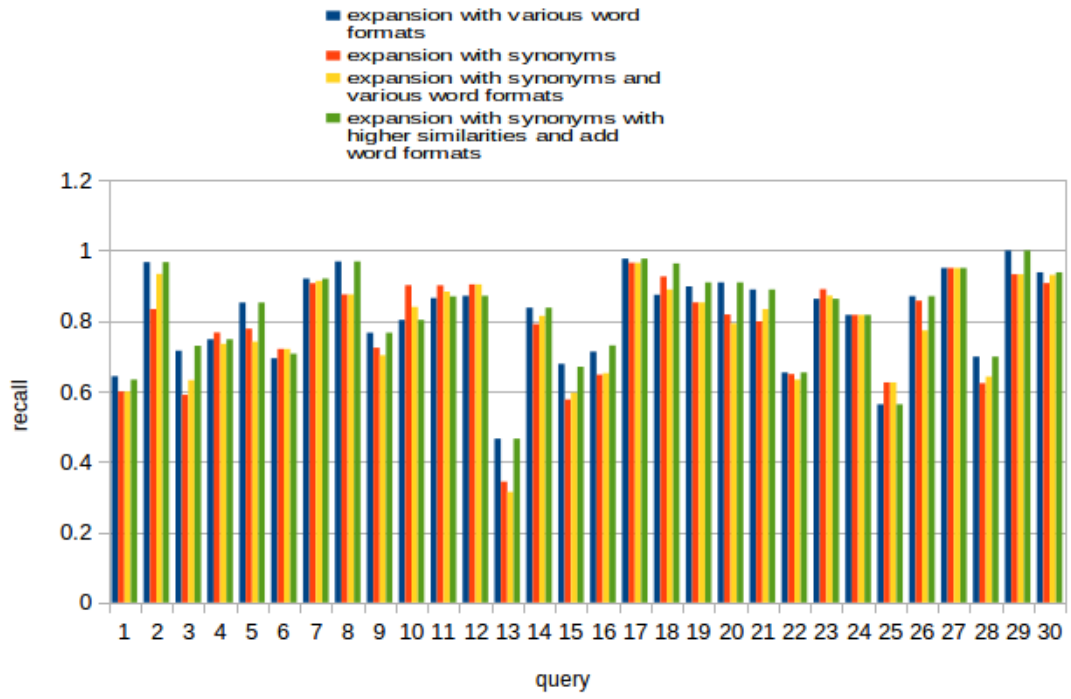

Figure 3.22: 2015 expansion performance with WordNet-recall part 2

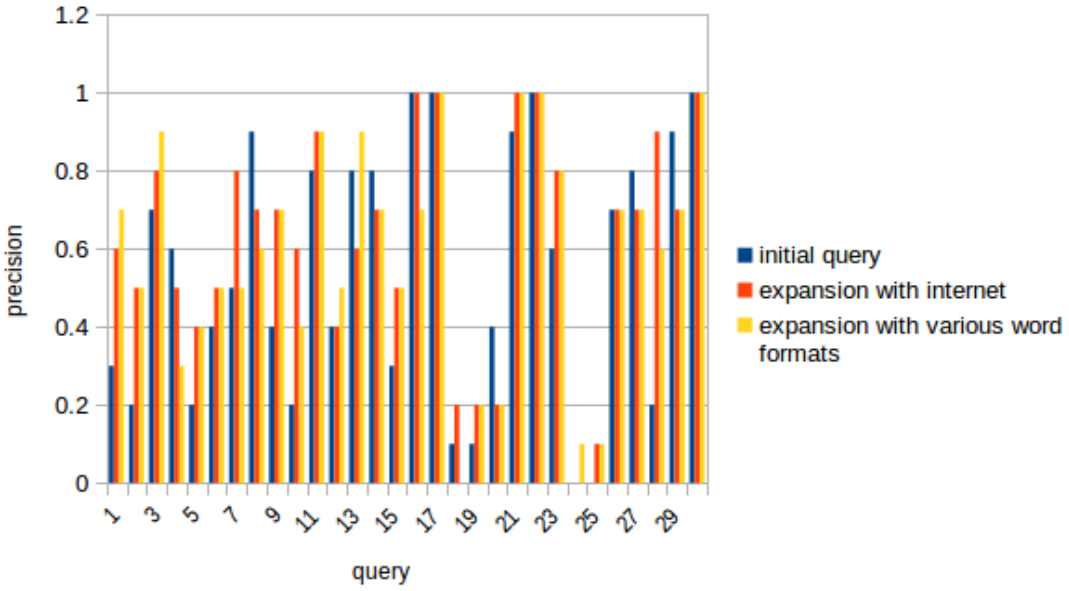

Figure 3.23: 2015 expansion performance with WordNet-precision part 1

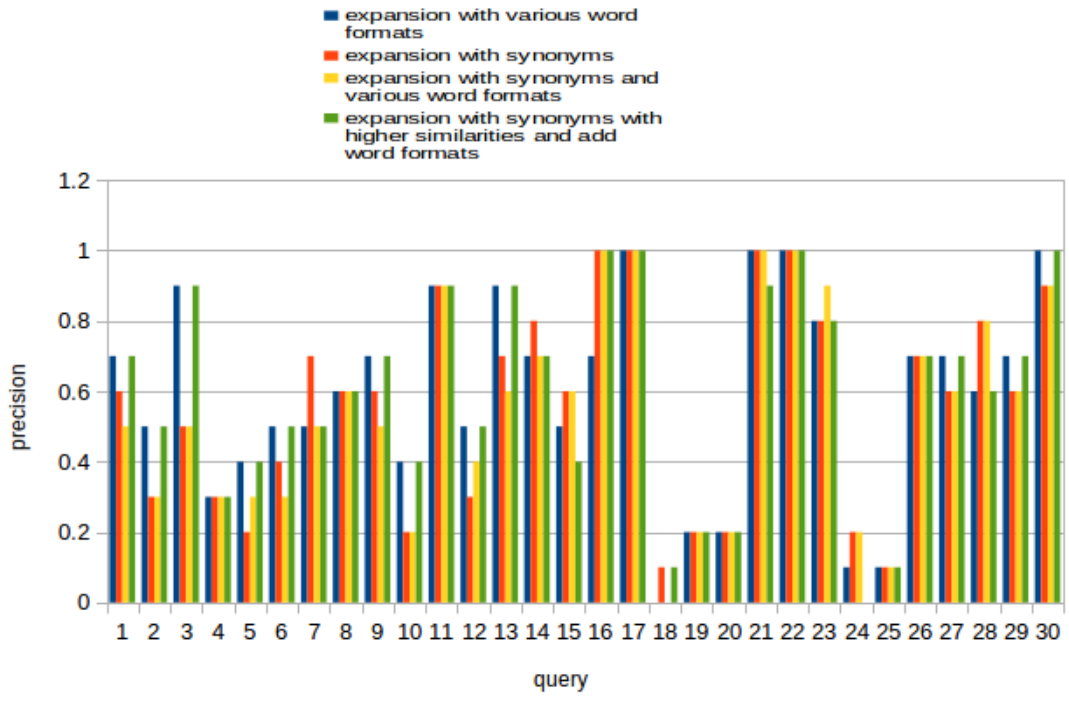

Figure 3.24: 2015 expansion performance with WordNet-precision part 2

Table 3.13: 2015 expansion average performance with WordNet part 2

metric	with word formats	with synonyms	with synonyms and word formats	with more similar synonyms and word formats
recall	0.813	0.782	0.778	0.818
precision	0.593	0.563	0.547	0.597

Doc2vec

According to the previous section, queries expanded by external online data sources and domain knowledge perform best. Given combining the document similarities to expanded queries obtained by doc2vec with TF_IDF to rerank the relevance of documents, we apply the following reranking equation.

$$new\_score = TF\_IDF\_score * doc2vec\_similarity \tag{3.6}$$

In figure 3.25, we detect after reranking the results retrieved by TF_IDF model with expanded queries, the new ranking results increased recall rates in top 1000 documents slightly in a majority of queries. Similarly reranking made precision of the majority of queries increased in figure 3.26.

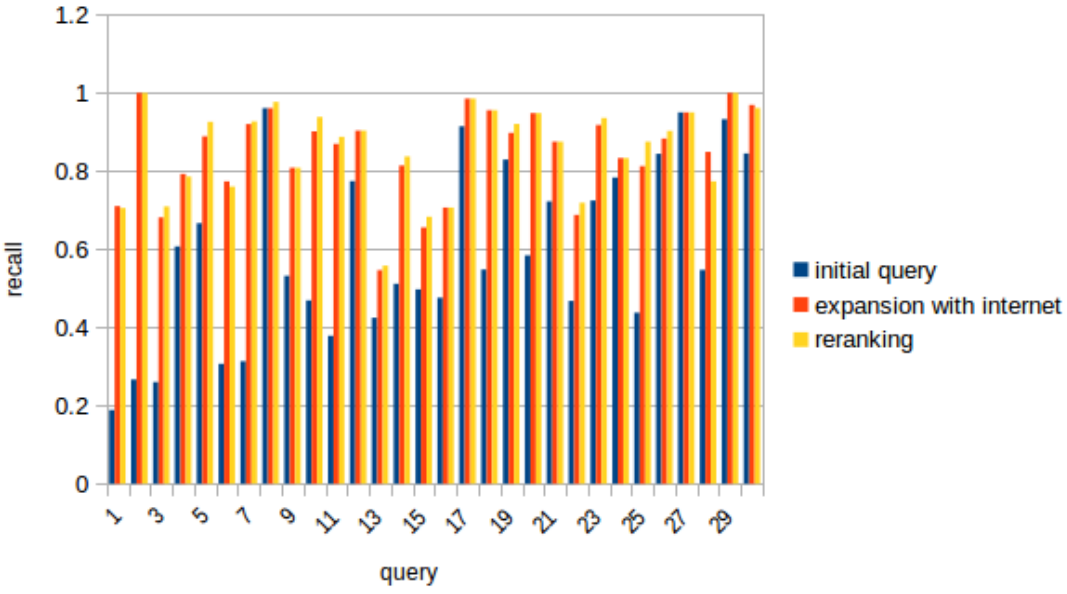

Figure 3.25: 2015 reranking comparison-recall

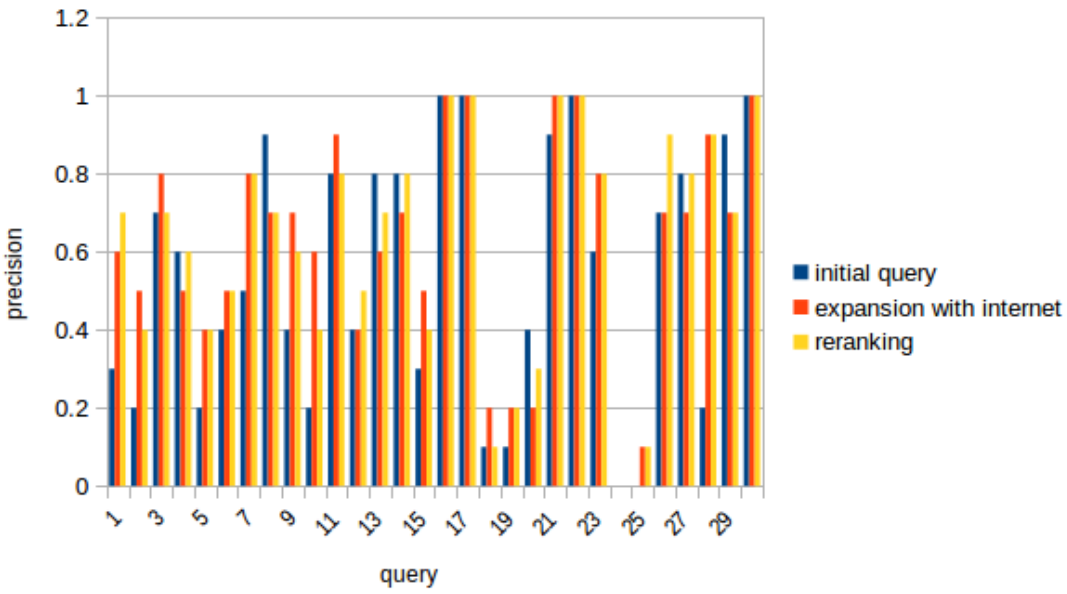

Figure 3.26: 2015 reranking comparison-precision

Table 3.14: 2015 reranking average performance

metric	initial query	expansion with internet	reranking	improved
recall	0.592	0.850	0.858	0.9%
precision	0.540	0.623	0.627	0.6%

3.1.3 TREC CDS 2014

Same as 2015, there are two parts in each topic, a description and a summary. Table 3.15 is the retrieval result when we use summaries as queries.

Expansion with description

We still first compare the performance of summaries and descriptions when they are selected as queries separately.

Figure 3.27 shows the comparison result about top 1000 recall performance. The queries with summaries work better than descriptions in most cases. Results of precision about top 10 retrieval data are shown in figure 3.28. The performance of summaries is obvious. In the majority of queries, precision of summaries is higher.

We choose summaries as initial queries, then combine summaries with descriptions as new

Table 3.15: 2014 search with summaries

query	recall@1000	recall@1000 percentage	p@10	p@10 percentage
1	44	0.537	4	0.4
2	80	0.308	5	0.5
3	18	0.375	2	0.2
4	37	0.481	5	0.5
5	31	0.233	1	1.0
6	22	0.233	1	1.0
7	92	0.902	9	0.9
8	111	0.563	9	0.9
9	6	0.545	4	0.4
10	22	0.759	7	0.7
11	40	0.400	2	0.2
12	38	0.290	3	0.3
13	33	0.294	5	0.5
14	48	0.295	4	0.4
15	60	0.645	8	0.8
16	8	0.229	1	0.1
17	38	0.576	3	0.3
18	41	0.500	1	0.1
19	58	0.611	4	0.4
20	59	0.410	1	0.1
21	76	0.444	8	0.8
22	25	0.490	1	0.1
23	39	0.307	0	0
24	55	0.611	5	0.5
25	11	0.324	0	0
26	62	0.899	10	1.0
27	299	0.659	10	1.0
28	33	0.635	4	0.4
29	78	0.918	8	0.8
30	29	0.326	1	0.1

Table 3.16: 2014 average performance comparison among summary, description and combination

metric	summary	description	combination
recall	0.494	0.441	0.459
precision	0.420	0.260	0.270

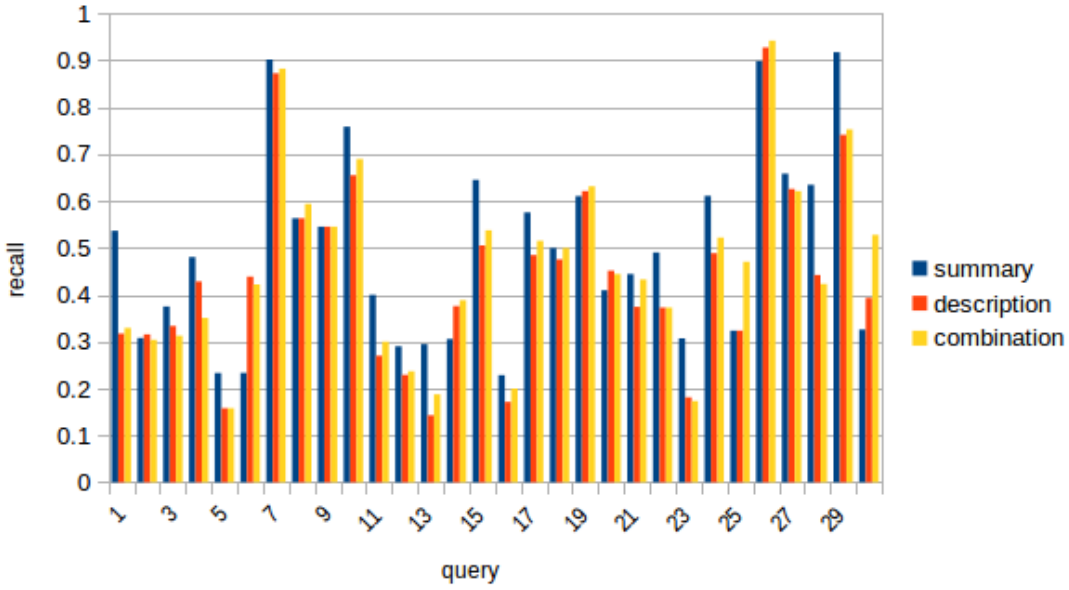

Figure 3.27: 2014 performance comparison among summary, description and combination-recall

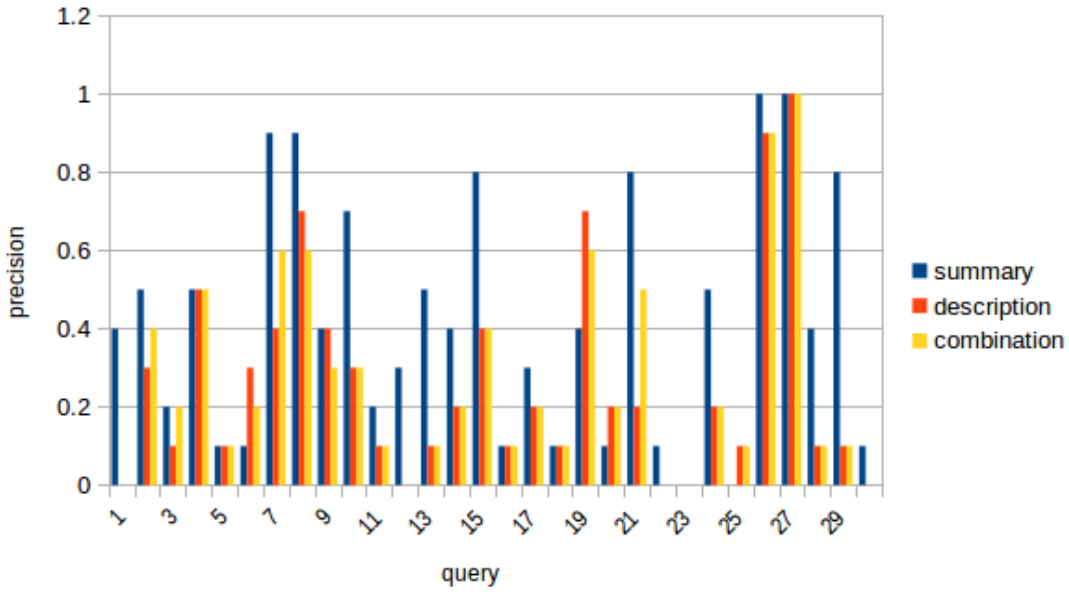

Figure 3.28: 2014 performance comparison among summary, description and combination-precision

queries. In figure 3.27, for the majority of queries, it's not hard to find the original queries with summaries work slightly better than descriptions and combinations. We compute the average recall for these 3 kinds of queries in table 3.16. The average performance of summaries is highest, but not significant.

Precision of top 10 retrieval results is also recorded in figure 3.28. More queries caught 0 relevant document after expansion. The average performance of summaries is obviously high.

Thus summaries are chosen as baselines.

Web Augmentation

We first delete noisy words in the light of medical corpus and apply MeSH on identifying keywords in the topics. In addition, we replenish domain knowledge and information obtained from internet with each initial query as what we did in the last two sections.

Figure 3.29 indicates the comparison result between initial queries and query expansions for recall. Performance is improved distinctly. About precision, figure 3.30 displays query 23 retrieves relevant data in top 10 retrieved results after comparison while no relevant result returned with initial query. The precision in most queries increases.

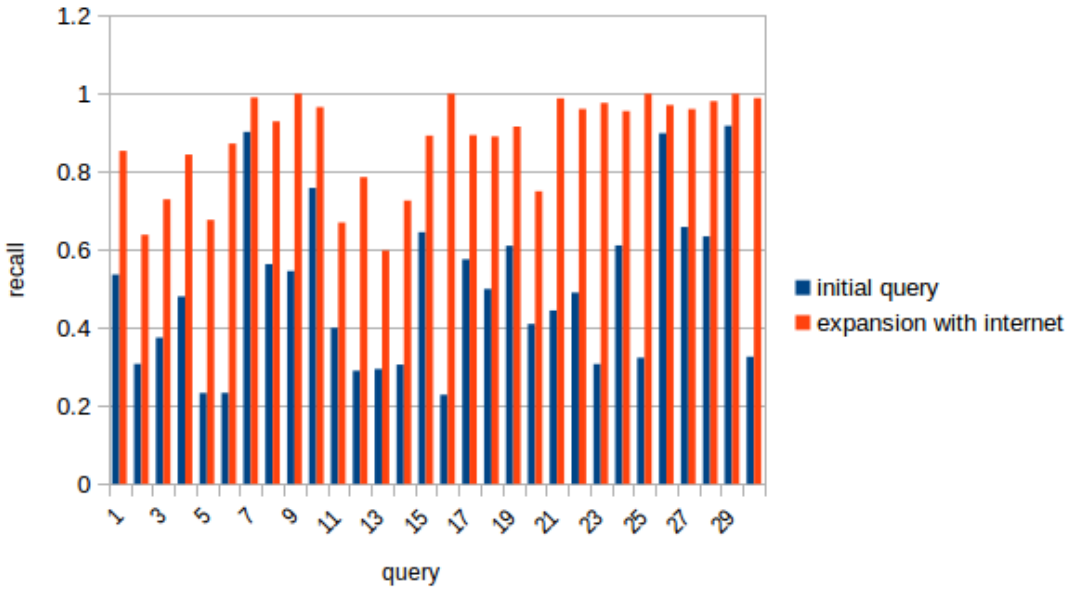

Figure 3.29: 2014 performance comparison between initial query and expansion with internet-recall

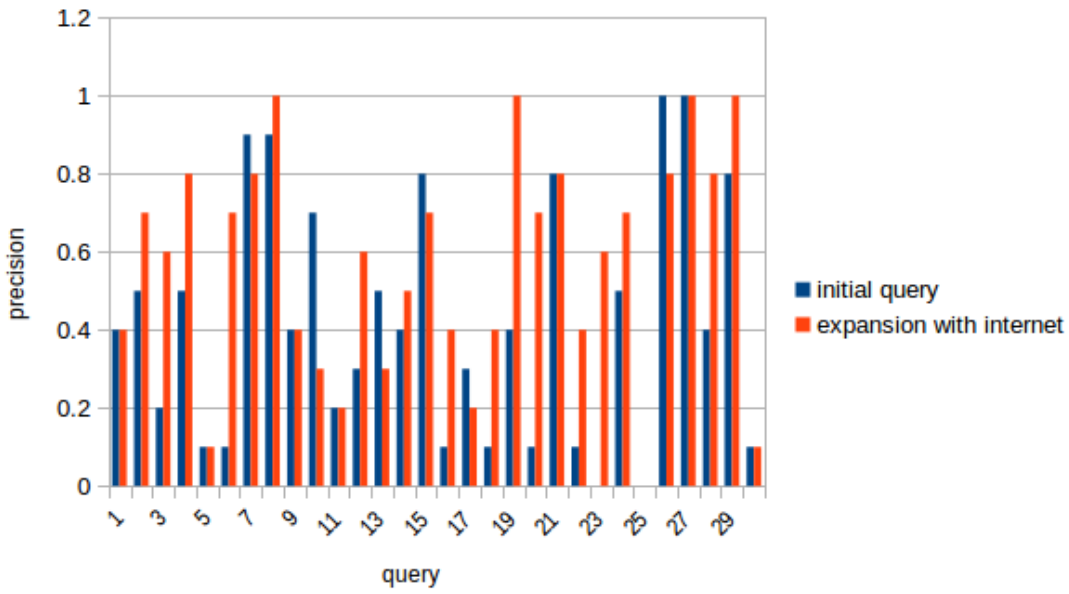

Figure 3.30: 2014 performance comparison between initial query and expansion with internet-precision

Table 3.17: 2014 average performance comparison between initial query and expansion with internet

metric	initial query	expansion with internet	improved
recall	0.494	0.880	78.1%
precision	0.420	0.567	35.0%

WordNet

Still aiming to adjust previous expanded queries further, we import multiple word formats and synonyms of words in the queries as what we did previously. We record retrieval results when complement more word formats and synonyms respectively or complement formats and synonyms one time.

We first discuss recall performance among top 1000 retrieved results. In figure 3.32, it's easy to see results of just importing more formats of words are almost highest. After removing synonyms which are less similar to original words under word2vec model, the performance is better than adding synonyms directly.

Figure 3.31 and 3.32 reveal that queries expanded by external online data sources and domain knowledge perform best.

About precision among top 10 retrieved documents, the comparison process is similar. After adjustment with word2vec, the average performance of adding word formats and more similar synonyms is slightly better than just adding multiple word formats and better than other expansion ways by WordNet.

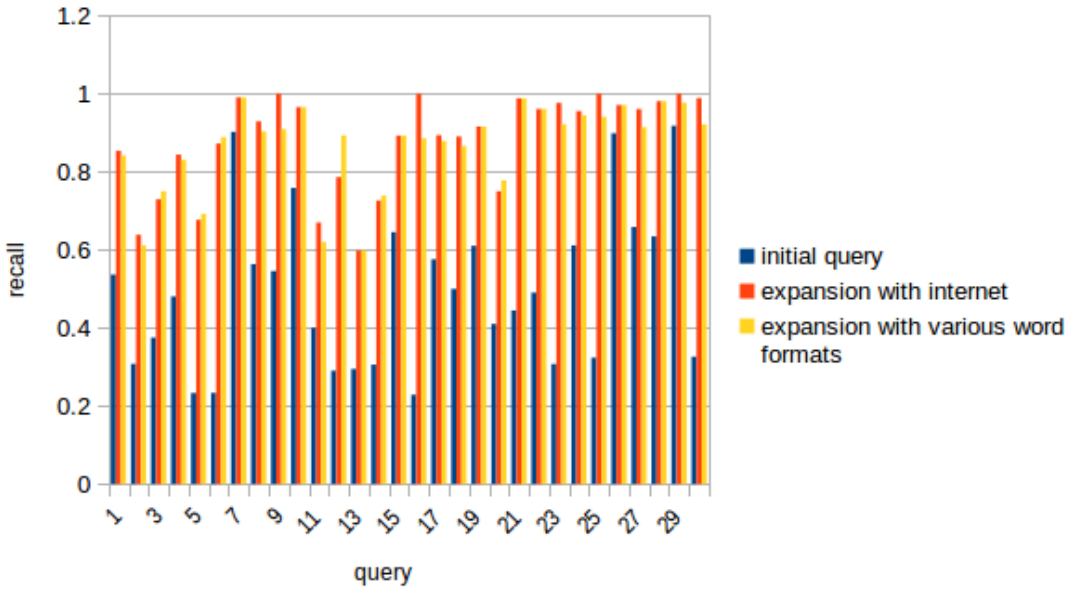

Figure 3.31: 2014 expansion performance with WordNet-recall part 1

Table 3.18: 2014 expansion average performance with WordNet part 1

metric	initial query	expansion with internet	expansion with word formats
recall	0.494	0.880	0.866
precision	0.420	0.567	0.510

Table 3.19: 2014 expansion average performance comparison with WordNet part 2

metric	with word formats	with synonyms	with synonyms and word formats	with more similar synonyms and word formats
recall	0.866	0.794	0.791	0.865
precision	0.510	0.413	0.417	0.513

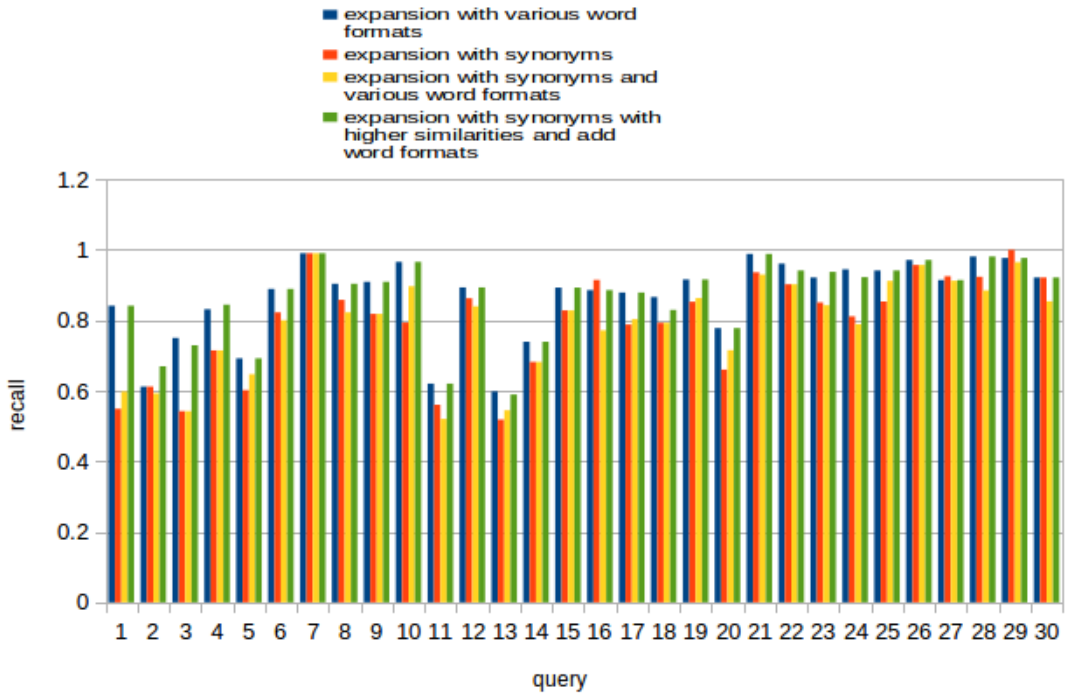

Figure 3.32: 2014 expansion performance with WordNet-recall part 2

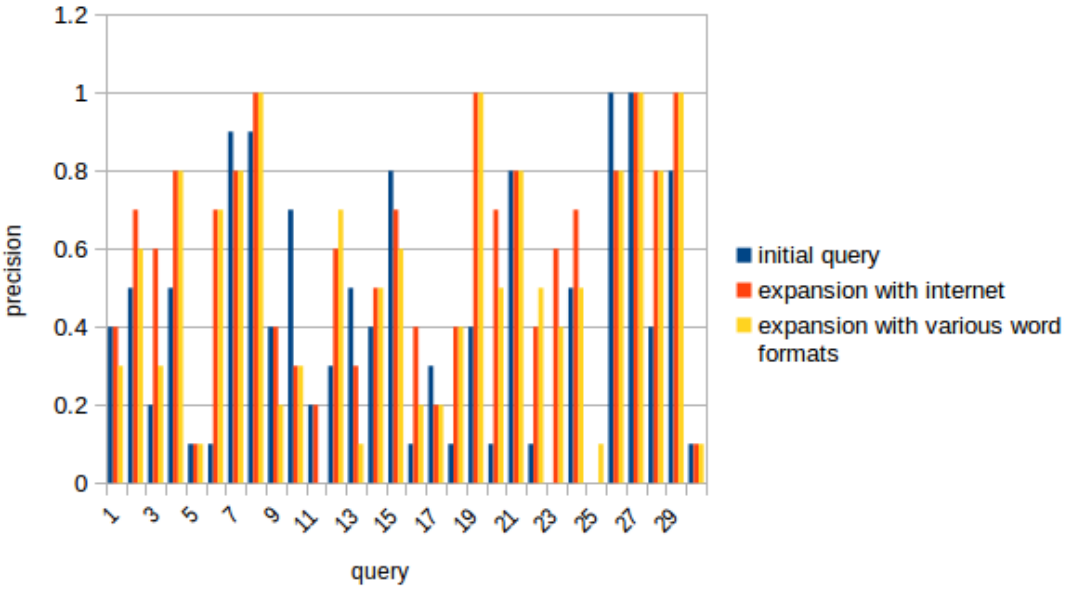

Figure 3.33: 2014 expansion performance with WordNet-precision part 1

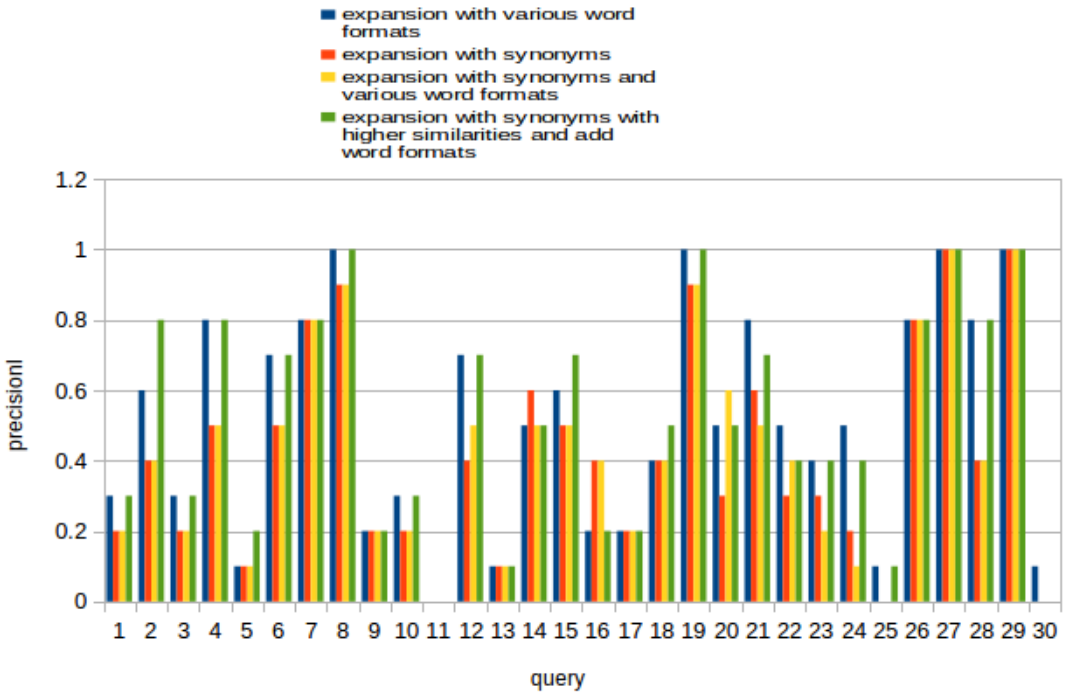

Figure 3.34: 2014 expansion performance with WordNet-precision part 2

Doc2vec

In line with the previous section, queries expanded by external online data sources and domain knowledge perform best. The document similarities to expanded queries obtained by doc2vec are considered for recomputing the relevance scores of documents, we exploit the following reranking equation.

$$new\_score = TF\_IDF\_score * doc2vec\_similarity \tag{3.7}$$

In figure 3.35, we detect after reranking the results retrieved by TF_IDF model with expanded queries, the new ranking results increase recall rates in top 1000 documents a bit for most queries. The average performance is also higher. New rank function also make precision of the majority of queries increased in figure 3.36. In addition, the average performance increases a little.

3.1.4 Completions

According to the experiment results from 2014 to 2016, we realize the information retrieval performance of utilizing external data sources and domain knowledge is best. It includes

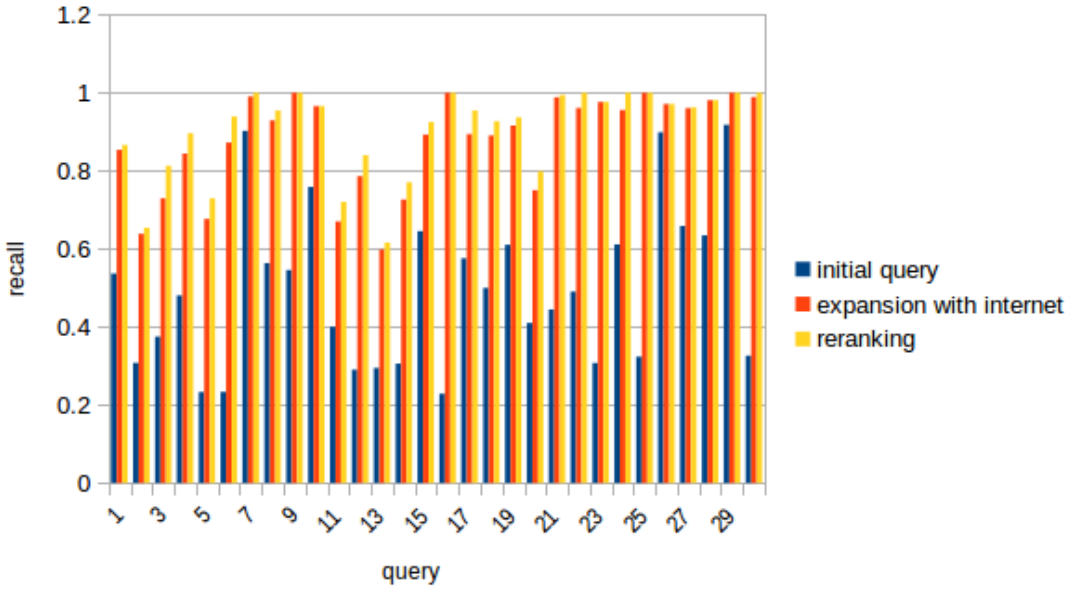

Figure 3.35: 2014 reranking comparison-recall

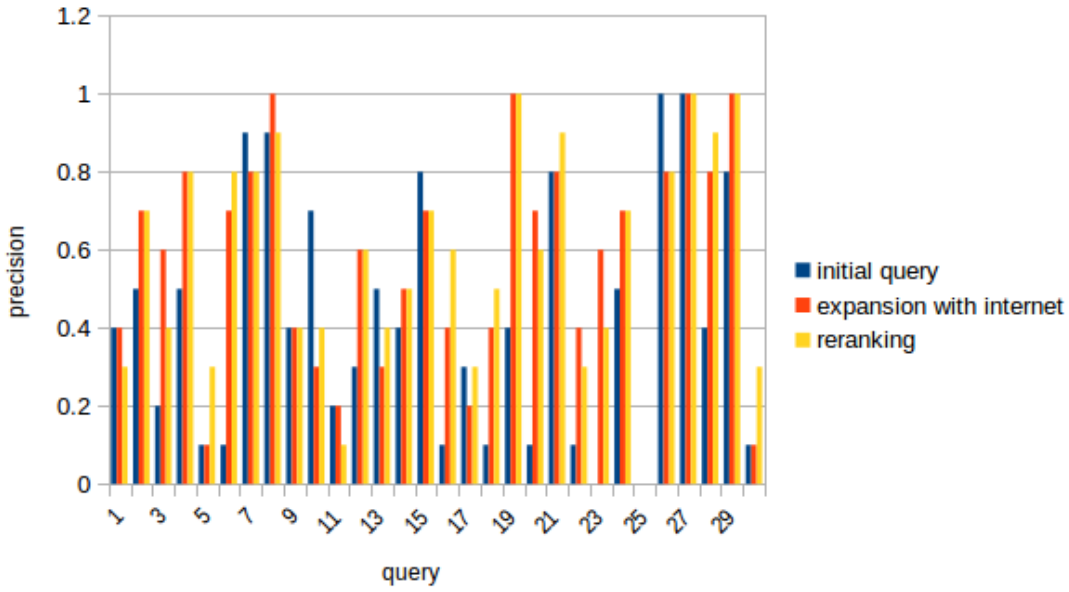

Figure 3.36: 2014 reranking comparison-precision

Table 3.20: 2014 reranking average performance

metric	initial query	expansion with internet	reranking	improved
recall	0.494	0.880	0.906	3.0%
precision	0.420	0.567	0.580	2.3%

two parts. First extract the effective information from topics. It mainly replies on MeSH. The second part is getting detailed clinical information, such as drug usages and causes of symptoms. We obtain these by online search. The average performance about recall@1000 and precision@10 increases obviously in three years comparing to initial queries.

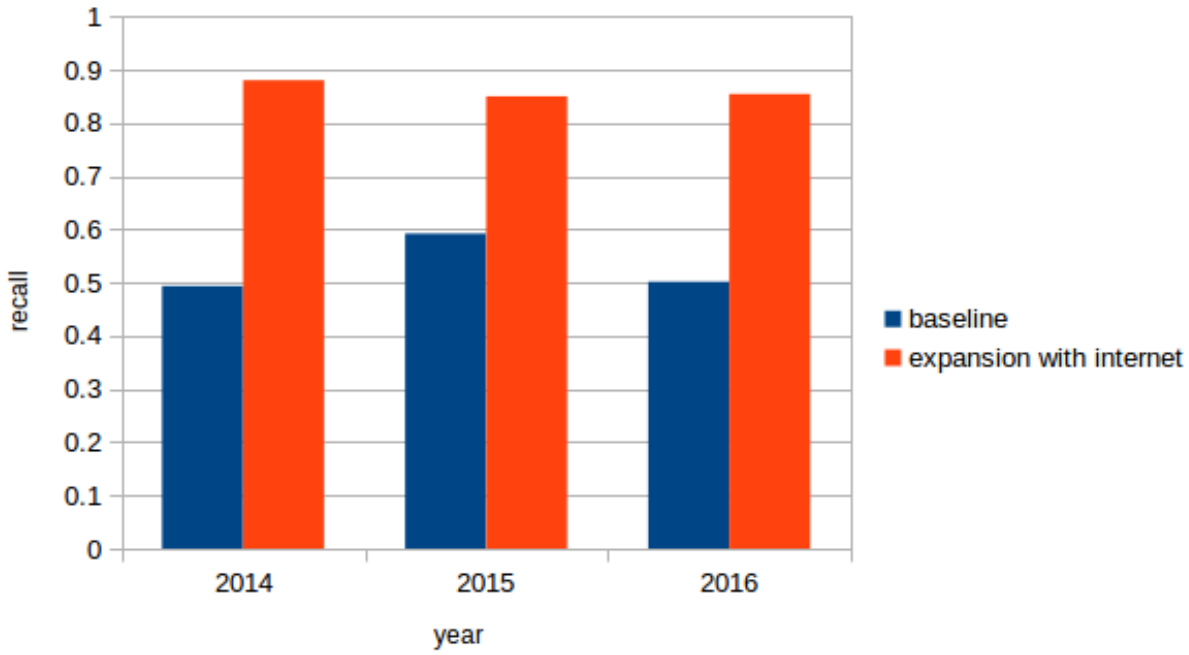

Figure 3.37: Average performance-recall

During the process of expanding queries, we discover sometimes the locations of symptoms are necessary. For example, the location of abdominal pain. Because there are different organs in left, right, upper and lower abdomen. Corresponding to the organs, the diseases are various. However if the location is known, the scope of diseases is limited. It means the possible keywords can be decided more easily. Thus we keep location terms in the queries, then complement relative possible diseases and organs. Besides, age is an important factor in the query, especially for kids. The detailed age is not required. But some indicated terms are effective to improve information retrieval performance, such as "child", "infant", "adolescent" and so on. We replace the age number with the indicated terms. With respect

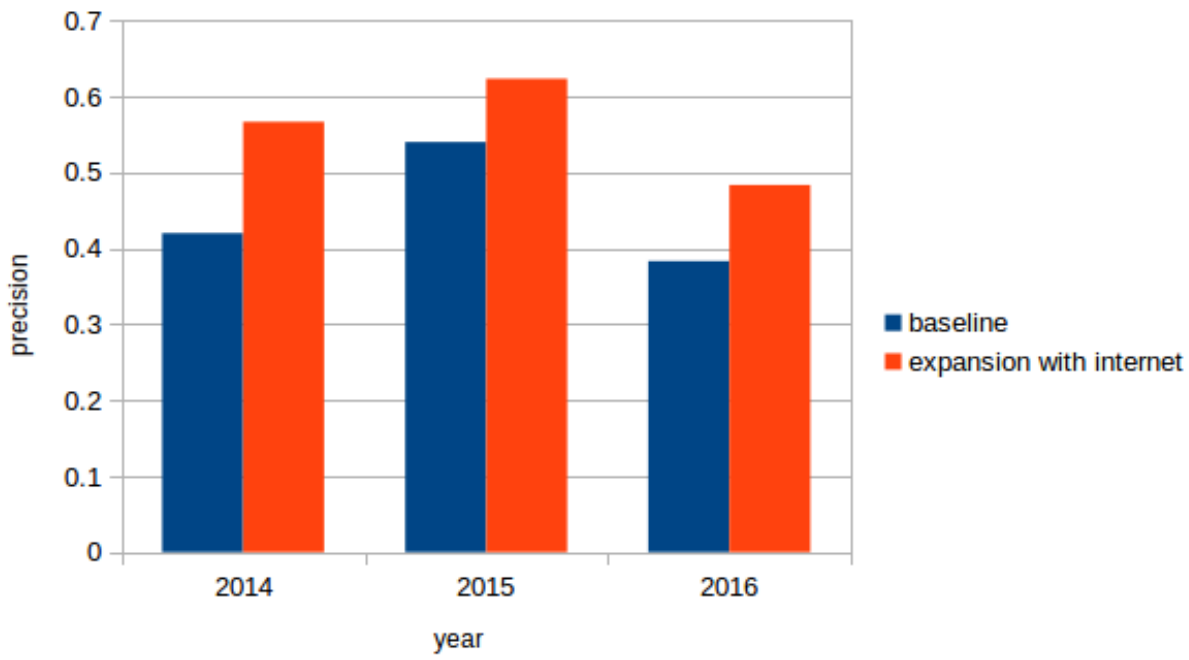

Figure 3.38: Average performance-precision

to age, gender can be ignored in most cases. In addition, there are 8 queries about accidents, 75% of them are expanded with the keyword "trauma". In some cases, "injury" and "wound" are added as keywords for the queries. We discern bathroom is a place where accidents happened frequently. The recall of query is decreased if we delete the term "bathroom". Beyond that, we discover if patients have vascular issues, limbs pain and difficulty in action in the descriptions, "embolism" can be used to expand queries, particularly when the patients are elder.

The mentioned prerequisites can be recognized easily. Once the conditions are satisfied, we can expand queries by relevant words. Our work is useful when researchers want to build the model to map symptoms to diagnosis and treatment. Our findings can also be utilized to rearrange the weights of terms in the queries so as to adjust ranking functions.

After expanding queries with external/internet resources and domain knowledge, we think over adjusting word formats/part of speech (POS) of terms in the queries through WordNet. About recall, there are 15 queries, 10 queries and 9 queries increasing a little or keeping same from 2014 to 2016. It generally increases a bit for pure pulmonary diseases, after injecting more adjective words. As for precision, the majority of queries doesn't decrease.

There exists limits of current WordNet. As we know, "hypertensive" is the adjective format of noun "hypertension", but we failed to switch between these two formats with current WordNet. In addition, we can't get "gastrointestinal bleeding" as the synonym of "coffee

ground" by generic WordNet either. In case we have the medical WordNet, we can expect increase the performance of query expansion by adjusting POS.

As mentioned, we also utilize WordNet to find the synonyms of terms in the queries. Besides lacking of some mappings among medical terms, the generic WordNet introduces some seldom-used noisy words, like petty. Moreover, it generates redundant words. For example, the original term is "exercise", but WordNet returns "usage" as synonyms too. Another case is getting synonyms of "heart". The returned synonyms contain "center", "pump", "middle" and so on. We take advantage of word2vec, removing synonyms with smaller similarities to original term in the clinical context. Although the average performance is still lower than without WordNet, it is higher than combining synonyms and word formats directly. The average recall improves $5.1\% - 9.4\%$, while precision improves $9.1\% - 26.1\%$.

Finally doc2vec model is utilized to modify ranking functions. The improvements on 2015 and 2016 are not obvious in comparison to 2014. The average increment on 2014 recall is near 3%, as well as precision increases 2.3% approximately. The accuracy of doc2vec model is still an issue.

3.2 Learning to rank

To ensure more relevant retrieval documents in front of irrelevant documents, the technology learning to rank is applied.

Before ranking the top 1000 retrieved documents in 2014 to 2016 separately, training a model is essential. The documents listed in TREC evaluation reference consist of training set. We expect to get a model which learned a binary classifier that can tell if a document is relevant to the query or not. We mark score of relevant document as 1 while score of irrelevant document as 0 in the model. Documents with score as 1 are put on the top. In this section, a logistic regression model is applied.

Documents are represented as vectors by doc2vec model so that they can be used for learning to rank directly. Instead of only utilizing vectors of documents, the vector of related query is appended to the vector of each document. Since there are 90 queries in total, each document has 90 vector representations mapping to the relevance results. In this section, queries expanded with web augmentation are baselines. Results are shown in the following tables and figures. We evaluate ranking performance by the following metrics, mean reciprocal rank (MRR), p@10, discounted cumulative gain (DCG) and R_precision.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}, \tag{3.8}$$

where $|Q|$ is the number of queries and $rank_i$ represents the rank position of the first relevant document of the $i - th$ query.

Table 3.21: Average performance of learning to rank

metric	baseline	learning to rank	improved
MRR	0.784	0.847	8.0%
p@10	0.623	0.620	-0.5%
DCG_{10}	3.424	3.254	-5.0%
R_precision	0.149	0.143	-4.2%

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}, \tag{3.9}$$

where p is the rank position and rel_i is the relevance of the retrieved document at position i. Here p is selected as 10.

The results indicate MRR is improved after learning to rank. More metrics are shown in table 3.21. The values of p@10 are very near. However overall performance of learning to rank still requires to be improved.

Chapter 4

Conclusions and Future Work

After the previous discussion, the expansion effect of introducing internet data sources and domain knowledge is most significant. Wiki provides explanations of symptoms, for example, "melena" means "upper gastrointestinal bleeding". MeSH marks the essential clinical terms and also shows the hierarchy and classification of symptoms, diseases and organs. Besides, the online search with google retrieves possible diagnosis, drug usage, complete information about diseases in the patients' medical history and other clinical information. Such information are utilized to map relevant documents efficiently. Beyond that, in our case, if two queries generated by TREC topics are about a similar issue, like abdominal questions, they may share overlapped domain knowledge. The clinical information in one query can also be used by another query.

Although the average performance of importing synonyms and word formats is lower than external online data sources and domain knowledge, which makes the second step of expansion can be avoided, the performance of some queries is improved indeed. We can expect to increase the performance by building a medical WordNet so that only medical synonyms can be added into queries. Another way is that select more important terms in the query for word formats transition and synonyms addition. It requires the help of experts.

Currently we have searched documents without classification information for each query. If we have the information of document classification, we can only search documents in the same category as the query. Mesh can be exploited to get rough categories of queries. The efficiency and retrieval performance are expected to be raised obviously.

TF_IDF model is the basic weighting model for retrieval. However the frequencies of words in the documents effect searching performance. For example, there are some topics including cardiac issues as medical history, however the frequency of "cardiac" is high in the dataset. When "cardiac" is added into queries and the current possible diagnosis is not about cardiac issues, the performance of retrieval will be decreased significantly. It's hard to remove all medical history directly especially when suffered diseases are relevant to current diagno-

sis. Thus we need to provide a new way of ranking function tuning [28, 30] and relevance weights [29] in the future.

Now we make use of doc2vec model to obtain the similar documents to queries, then adjust the ranking scores returned by TF_IDF model with the similarities of documents. We can consider combine blind feedback and ranking function tuning together [28]. Another future work is to improve the accuracy of doc2vec model. Therefore the similarities among documents and queries will be evaluated more effectively.

We have explored the application of learning to rank to arrange relevant documents on the top. Document features are extracted by doc2vec. Learning to rank basically performs superior when consider the rank position of the first relevant document for each query. But the overall performance needs to be improved. We plan to tune the representations of documents with more complicated ways.

Bibliography

[1] Jainisha Sankhavara and Prasenjit Majumder. Team DA IICT at Clinical Decision Support Track in TREC 2016: Topic Modeling for Query Expansion. *In Proceedings of the 2016 Text Retrieval Conference*, 2016.

[2] Hongyu Liu, Yang Song, Yun He, Yueyao Wang, Qinmin Hu and Liang He. ECNU at TREC 2016: Web-based query expansion and experts diagnosis in Medical Information Retrieval. *In Proceedings of the 2016 Text Retrieval Conference*, 2016.

[3] Danchen Zhang, Daqing He, Sanqiang Zhao and Lei Li. Query Expansion with Automatically Predicted Diagnosis: iRiS at TREC CDS track 2016. *In Proceedings of the 2016 Text Retrieval Conference*, 2016.

[4] Kuang Lu and Hui Fang. Query Performance Prediction and Topic Shift in Microblog Retrieval. *In Proceedings of the 2016 Text Retrieval Conference*, 2016.

[5] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald and Dou-glas Johnson. Terrier Information Retrieval Platform. *In Proceedings of the 27th European Conference on Information Retrieval*, 2005.

[6] Y. Zhang, L. Gong, and Y. Wang. An improved TF-IDF approach for text classification. *Journal of Zheijang University*, 6(1), 4955, 2005.

[7] Stephen. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129146, 1976.

[8] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513523, 1988.

[9] S. E. Robertson and S. Walker. Some simple effective approximations to the 2poisson model for probabilistic weighted retrieval. *In SIGIR 94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 232241, 1994.

[10] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. *In SIGIR 96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 176-184, 1996.

[11] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery*, 7(3):216244, 1960.

[12] E. Ide. New experiments in relevance feedback. *In the SMART Retrieval System*, G.SaltonEd., Prentice Hall, Englewood Cliffs, N. J., 337354, 1971.

[13] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, 313323. Prentice-Hall Inc., 1971.

[14] T. E. Doszkocs. AID, an Associative Interactive Dictionary for Online Searching. *Online Rev.* 2, 2, 163174, 1978.

[15] M. F. Porter. Implementing a probabilistic information retrieval system. *Inf. Tech. Res. Dev. Appl*, 1, 2, 131156, 1982.

[16] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391407, 1990.

[17] T. Hofmann. Probabilistic latent semantic indexing. *In SIGIR 99: In Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, 5057, 1999.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean. Distributed representations of words and phrases and their compositionality. *In Proceedings of NIPS*, 31113119, 2013.

[19] Tie-Yan Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 225-331, 2009.

[20] Chris Burges , Tal Shaked , Erin Renshaw , Ari Lazier , Matt Deeds , Nicole Hamilton and Greg Hullender. Learning to rank using gradient descent. *In Proceedings of the 22nd international conference on Machine learning*, 89-96, 2005.

[21] C. J. C. Burges, R. Ragno and Quoc Viet Le. Learning to rank with nonsmooth cost functions. *In Advances in Neural Information Processing Systems (NIPS 19)*, 193200. 2007.

[22] L. Rigutini, T. Papini, M. Maggini and F. Scarselli. SortNet: Learning to rank by a neural preference function. *IEEE Trans. Neural Netw.*, 22(9), 1368-1380, 2011.

[23] Jun Xu and Hang Li. AdaRank: a boosting algorithm for information retrieval. *In SIGIR 07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 391-398, 2007.

[24] Zhe Cao , Tao Qin , Tie-Yan Liu , Ming-Feng Tsai and Hang Li. Learning to rank: from pairwise approach to listwise approach. *In Proceedings of the 24th international conference on Machine learning*, 129-136, 2007.

[25] Aliaksei Severyn and Alessandro Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *In SIGIR 15: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 373-382, 2015.

[26] Q. Le and T. Mikolov. Distributed Representations of Sentences and Documents. *In Proceedings of ICML 2014*, 2014.

[27] G. A. Miller. WORDNET: A Lexical Database for English. *Communications of ACM*(11): 39-41, 1995.

[28] W. Fan, M. Luo, L. Wang, W. Xi and E. A. Fox. Tuning Before Feedback: Combining Ranking Discovery and Blind Feedback for Robust Retrieval. *In SIGIR 14: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 138145, 2004.

[29] W. Fan, M.D. Gordon and P. Pathak. Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison. *Decision Support Systems*, 40, 213233, 2004.

[30] W. Fan, M. D. Gordon and P. Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *IPM-04*, 40(4), 587602, 2004.