ECNU at 2015 CDS Track:Two Re-ranking Methods in Medical Information Retrieval

Yang Song[2], Yun He[2], Qinmin Hu[1,2], and Liang He[1,2]

[1] Shanghai Key Laboratory of Multidimensional Information Processing
[2] Department of Computer Science & Technology, East China Normal University, Shanghai, 200241, China
{ysong,yhe}@ica.stc.sh.cn, {qmhu,lhe}@cs.ecnu.edu.cn

Abstract. This paper summarizes our work on the TREC 2015 Clinical Decision Support Track. We present a customized learning-to-rank algorithm and a query term position based re-ranking model to better satisfy the tasks. We design two learning-to-rank framework: the pointwise loss function based on random forest and the pairwise loss function based on SVM. The position based re-ranking model is composed of BM25 and a heuristic kernel function which integrates Gaussian, triangle, cosine and the circle kernel function. Furthermore, the Web-based query expansion method is utilized to improve the quality of the queries.

1 Introduction

The goal of the TREC 2015 Clinical Decision Support Track is to retrieve the biomedical articles for answering generic clinical questions about medical records[3]. There are two tasks in this year's CDS track. Task A is identical to the 2014 CDS track. Each topic contains two versions of the case narratives. The topics " descriptions" contain a complete account of the patients' visits, including details such as their vital statistics, drug dosages, etc., whereas the topics "summaries" are simplified versions of the narratives that contain less irrelevant information. The two versions in the topic are functionally equivalent. Task B is similar to CLEF e-Health 2013 & 2014 task, which is provided with a diagnosis field for the treatment type and test type topics.

Our approach is composed of two parts: the learning-to-rank algorithm and a term position based re-ranking model. In the learning-to-rank algorithm [1][2][3], the random forest based pointwise loss function and the SVM based pairwise loss function are adopted. In position based re-ranking approach, a customized kernel function is utilized to create a position based model.

[3] http://www.trec-cds.org/2015.html

2 Methodology

2.1 Query Expansion

In our work, Web search engine is utilized to retrieve topics for more medical terminologies. Related medical technical terms are adopted as the query expansion terms into the query. After that, the new query is matched by the IR models and the ranking documents are achieved. The Web-based query expansion model is proposed as follows.

- Query is searched by Google and the top 10 concurrent Web titles and snippets (if existed) are crawled from the Web page.
- By applying the MeSH database, the medical terms are extracted from both the titles and the snippets.
- The frequency of each stemmed medical term is calculated. Only the terms appearing more than n times are kept for expanding, which can be denoted as Q_{web}.
- The final query is formulated as $Q = Q_0 \cup Q_{web}$, where Q_0 represents the initial query.

In addition, since the topics have three types "what is the patient's diagnose?", "what tests should the patient receive?" and "how should the patient be treated?", we automatically add the keywords 'diagnose', 'test' and 'treatment' as the expansion terms to the queries according to their types.

2.2 Re-ranking Method

2.2.1 Learning-to-rank

Feature Extraction: We extract the weighting score of each document-query pair from a retrieval model as features. The weighting score is the result of the first retrieval round, which represents the relevance assessed by retrieval model. To utilize the advantage of different retrieval models, we obtain the scores from BM25, PL2 and BB2 model. Hence, in our learning-to-rank platform, the dimension of feature vector is 3.
Pointwise based on Random Forest: Random forest is composed of several decision trees which are independent on each other. Each decision tree will classify the sample in the test data set, the final result of classification depends on the vote of all the trees [7]. We apply random forest to implement pointwise approach which classifies the document-query pairs into relevant or not. We apply only 100 decision trees in our forest. The training data is transformed from the results in 2014 task. The weighting scores of BM25, PL2 and BB2 model are extracted to represent the document-query pairs in the previous results.
Pairwise based on SVM: SVM is a maximum margin classifier. We utilize svmRank to implement pairwise approach which compares the relevance between candidates documents in the search results.

Model Application: Firstly, we apply BM25, PL2 and BB2 model to achieve three initial results. Then a new result is achieved ordered by the combination of scaled scores of three retrieval model. For pointwise, random forest is utilized to classify the candidate pairs in the new result. Document-query pairs which are classified as relevant will award extra relevance score. Then, the result is re-ranked by the new score. In pairwise, svmRank is utilized to calculate the relevance score of document-query pairs. Then, the result is re-ranked by the relevance score.

2.2.2 Position Based Approach

We assume that occurrence of a query term has an impact towards its neighbouring text [4][5][6]. This impact attenuates when a position is farther away. The kernel function can be utilized to estimate query term occurrence's impact.

Suppose σ is the farthest distance query term can impact on. For a given query term and the given document, the parameter σ belongs to this query term is defined by equation 1.

$$\sigma = \frac{n}{N} \cdot \frac{1}{qtf}. \tag{1}$$

Where, N is the total number of words in the candidate document. n represents the different words number in the candidate document. qtf is the target query term's word frequency.

Our kernel function is made up by gaussian, triangle, cosine and circle kernel functions, which is defined as follows:

$$\begin{aligned} Kernel(u) = {} & sgn(a-x) \cdot exp(\frac{-u^2}{2\sigma}) + sgn(x-a)(b-x) \cdot (1 - \frac{u}{\sigma}) \\ & + sgn(x-b)(c-x) \cdot \frac{1}{2}[1 + cos(\frac{u\pi}{\sigma})] \\ & + sgn(x-c)(d-x) \cdot \sqrt{1 - (\frac{u}{\sigma})^2}, \quad u \leq \sigma \end{aligned}$$

Where, the parameters a, b, c and d control the range each kernel function works and can be assigned by user arbitrarily. u is the distance away from the query term. The variable x which is relevant to u and σ is defined as follows.

$$x = \frac{u}{\sigma\sqrt{\sigma^2 + u^2}}$$

Given two query terms q_i and q_j, we say that they generating cross term q_{ij} [6]. The word frequency of the cross term q_{ij} in the given document is the sum of $Kernel_i(u_i)$ and $Kernel_j(u_j)$. Where u_i and u_j are belong to q_i and q_j respectively. Word frequency $qtf_{q_{ij}}$ of cross term q_{ij} in query is defined identical to $tf_{q_{ij}}$.

We applied the cross term into BM25 algorithm, then the weight of cross term q_{ij} in document D can be defined by equation 2.

$$w(q_{ij}, D) = \frac{(k_1+1)tf_{q_{ij}}}{K + tf_{q_{ij}}} \cdot \frac{(k_2+1)qtf_{q_{ij}}}{k_2 + qtf_{q_{ij}}} \cdot log\frac{N - n + 0.5}{n + 0.5} \tag{2}$$

Where $k1$ and $k2$ are tuning constants which depend on the dataset used. $K = (1-h) + h \cdot \frac{dl}{avdl}$. dl is the length of the candidate document, and $avdl$ is the average document length.

Finally, a document's position based weight under the given query is awarded by the sum of all cross terms' weights in this document.

$$pos\_weight(D) = \sum_{i<j} w(q_{ij}, D) \tag{3}$$

We combine the results obtained by applying BM25, BB2 and PL2 retrieval models. Each document in the precious combination achieves a position based weight through the position based model. Then, the new score of a candidate document is determined by the sum of its normalized position based weight and the initial weight assigned by combination method. At last, the candidate documents are re-ranked according to their new scores.

3 Experiments and Evaluation

3.1 Document Processing

In the document processing, we first divide every article into six segment, including title, abstract, text, table, figure and reference. Then, we extract age and gender as the additional information.

Analysis of the various parts of the article: The document is given in the NXML format. We believe that title and abstract parts are the most important, followed by the content and reference parts. What is more, we believe that the captions of tables and figures are very useful. Therefore, we extract the title, abstract, text, tables' captions, figures' captions and the reference part from the raw data.

Age and gender: Regular expression are used to extract and normalize age and gender information from the documents and queries. According to the age division standard released by the United Nations we make age into 12 categories. Table 1 shows the details of our classification. For gender, we define male, him, his, he, gentleman, gentlemen, man, men, boy, and boys as male. At the same time, we define female, her, she, hers, lady, ladies, woman, women, girl and girls as female.

3.2 Submissions and Evaluation

In task A, our queries are composed of summaries of topics, expansion terms gained though Web based method and topics types. Then, in task B, the queries

Table 1. Age Categories

Age Bracket	Tag
0-1	baby
1-7	childhood
8-12	youth
13-18	young teenage
19-25	middle teenage
26-35	old teenage
36-45	young adult
46-55	middle adult
56-65	old adult
66-75	young old
76-85	middle old
>86	elder

Table 2. Summary of evaluation

Run	MAP	infNDCG	bpref	p@10
ecnu1	0.1641	0.2664	0.2166	0.4500
ecnu2	0.1632	0.2680	0.2157	0.4533
ECNUPB	0.0220	0.0702	0.0782	0.1033
ecnu3	0.2216	0.3821	0.3066	0.5633
ecnu4	0.2207	0.3749	0.3049	0.5533
ECNUBP	0.1010	0.1656	0.2128	0.1967

are made up by queries used in task A and diagnosis field provided by the organizer.

We applied the IR system Terrier[4] to implement BM25, PL2, and BB2 models. Their results are combined by the following strategy. First, we use the 0-1 normalization on each run. Note that we select the top 5000 results in a run, instead of the top 1000 results. Second, we put all the results into a pool such that the most overlapped results have the priority to be selected as the candidates. Finally, the top 1000 results are sorted as the combination run for re-ranking.

Here we submit six runs where the description for each run is as follows. And the evaluation of our submissions is summarized in Table 2.

- ecnu1: We utilize the learning-to-rank model to re-rank the results, which is based on the pointwise loss function.(Task A)
- ecnu2: we utilize the learning-to-rank model to re-rank the results, which is based on the pairwise loss function.(Task A)
- ECNUPB: we utilize the position based model to re-rank the combination run.(Task A)
- ecnu3: a combination of the results gained though BM25, PL2, BB2 model.(Task B)
- ecnu4: we utilize the learning-to-rank model to re-rank the results, which is based on the pairwise loss function.(Task B)
- ECNUBP: we utilize the position based model to re-rank the combination run.(Task B)

The main evaluation is infNDCG. In task A, ecnu1 obtains the fourth place over all the automatic results. In task B, ecnu3 achieves the first place over all the automatic results.

[4] http://terrier.org

Acknowledgment

This research is funded by Science and Technology Commission of Shanghai Municipality (No.15PJ1401700 and No.13511507902).

References

1. David C., Tong Z.: Subset Ranking Using Regression. In: Lecture Notes in Computer Science, 605-616 (2006)
2. Crammer K., Singer Y.:Pranking with Ranking. In: Advances in Neural Information Processing Systems, 641-647 (2001))
3. Chris B., Tal S., Erin R., Ari L., Matt D., Nicole H., Greg H.: Learning to rank using gradient descent. In: Proceeding of the 22th Internnational Conference on Machine Learning, 41(2):89-96 (2005)
4. Owen D.K., Alistair M.: Effective Dcument Presentation with a Locality-Based Similarity Heuristic. In: Proceeding of the 22th ACM SIGIR, ACM, 113-120 (1999)
5. Lv Y.H., Zhai C.X.: Positional Language Models for Information Retrieval. In: Proceeding of the 32th ACM SIGIR, ACM, 299-306 (2009)
6. Zhao J.S., Huang J.X.J, He B.: CRTER: Using Cross Terms to Enhance Probabilistic Information Retrieval. In: Proceeding of the 34th ACM SIGIR, ACM, 155-164 (2011)
7. Breiman L.: Random Forests. In Machine Learning. 45(1):5-32 (2001)