KISTI at CLEF eHealth 2016 Task 3: Ranking Medical Documents using Word Vectors

Heung-Seon Oh and Yuchul Jung

Korea Institute of Science and Technology Information
{ohs, jyc77 }@kisti.re.kr

Abstract. User's searching activity to obtain relevant medical information becomes very common as the general public uses the Web as source of health information. As a response to this phenomenon, there have been a number of approaches to find useful information for diagnosing or understanding their health conditions from the Web or medical literatures. As an ongoing effort to deliver useful medical information, we attempted two different approaches using word vectors learnt by Word2Vec with Wikipedia. At first, initial documents are obtained using a search engine. Based the retrieved documents, pseudo-relevance feedback is applied with two different usage of the word vectors. In the first approach, a feedback model is constructed using new relevance scores using the word vectors while it is constructed with a new query expanded.

Keywords: medical information retrieval, language models, pseudo relevance feedback, word vectors

1 Introduction

Laypeople use the Web to acquire medical information such as symptoms, diagnosis, treatments, diseases, and hospitals. Unfortunately, they may fail to find relevant information due to difficulty of representing information needs. This happens because they are often not only unfamiliar with medical terminology but also uncertain about their exact questions. To mitigate this problem, CLEF eHealth [2, 4] aims to support laypeople for finding and understanding medical documents on the Web by leveraging medical text processing techniques.

CLEF 2016 eHealth [3] continues to make an effort for the same purpose. We participate in task 3 (patient-centered information retrieval) that focuses on evaluating the effectiveness of medical information retrieval on the Web [10]. This task utilizes a vast of Web document collection, ClueWeb12-B, while the previous tasks employs about 1M Web documents collected from several health-related web sites. In this paper, we proposed two different approaches using word vectors obtained from Word2Vec to perform pseudo relevance feedback.

2 Method

2.1 Ranking framework

Our method is to rank medical documents using word vectors constructed from a medical resource, specifically medical Wikipedia. The aim of using the word vectors is to understand the information need of a query properly. For a query Q, a set of documents, $S = \{D_1, D_2, \ldots, D_{|S|}\}$, from a collection C are retrieved using a search engine. For a retrieval model, query-likelihood method with Dirichlet smoothing (QLD) is chosen [8]. Based on S, pseudo relevance feedback (PRF) using the word vectors is performed to re-rank the documents in S with a feedback model. In this step, the word vectors are adopted in two different approaches. In the first approach, they are used to compute relevance scores $score_{WV}(Q, D)$ between Q and D while, in the second approach, they are used to directly expand Q to Q_{WV} by adding more words that are not appear in Q. For each approach, final scores are computed by KL-divergence method with a feedback model constructed using $score_{WV}(Q, D)$ or Q_{WV}.

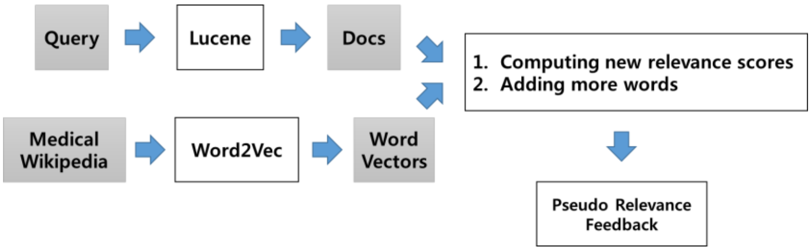

Fig. 1. Overview of ranking framework

2.2 Basic Foundation

KL-divergence method (KLD) is adopted to compute a relevance score between Q and D by estimating language models [5, 7, 9] because it has a principle to incorporate information into a query in PRF:

$$\begin{aligned} score_{KLD}(Q, D) &= \exp\left(-KL\left(\theta_Q || \theta_D\right)\right) \\ &= \exp\left(-\sum_w p\left(w|\theta_Q\right) log \frac{p(w|\theta_Q)}{p(w|\theta_D)}\right) \end{aligned} \quad (1)$$

where θ_Q and θ_D are the query and document unigram language models, respectively.

A query model is estimated by maximum likelihood estimation (MLE), as shown below:

$$p(w|\theta_Q) = \frac{c(w,Q)}{|Q|} \tag{2}$$

where $c(w,Q)$ is the count of a word w in query Q and $|Q|$ is the number of words in Q.

A document model is estimated using Dirichlet smoothing to improve retrieval performance [8]:

$$p(w|\theta_D) = \frac{c(w,D) + \mu \cdot p(w|C)}{\sum_t c(t,D) + \mu} \tag{3}$$

where $c(w,D)$ is the count of a word w in document D, $p(w|C)$ is the probability of a word w in collection C, and μ is the Dirichlet prior parameter.

Pseudo-relevance feedback (PRF) is a popular query expansion approach to update a query. It assumes that the top-ranked documents $F = \{D_1, D_2, \dots, D_{|F|}\}$ relevant to a given query and the words in F are useful to reveal hidden information needs. A relevance model (RM) is a multinomial distribution $p(w|Q)$, which is the likelihood of a word w in a query Q based on F. The first version of the relevance model (RM1) is defined as follows:

$$\begin{aligned} p_{RM1}(w|Q) &= \sum_{D \in F} p(w|\theta_D) p(\theta_D|Q) \\ &= \sum_{D \in F} p(w|\theta_D) \frac{p(Q|\theta_D) p(\theta_D)}{p(Q)} \\ &\propto \sum_{D \in F} p(w|\theta_D) p(\theta_D) p(Q|\theta_D) \end{aligned} \tag{4}$$

RM1 is composed of three components: the document prior $p(\theta_D)$, the document weight $p(Q|\theta_D)$, and the term weight in a document $p(w|\theta_D)$. In general, $p(\theta_D)$ is assumed to have a uniform distribution without knowledge of document D. $p(Q|\theta_D) = \prod_{w \in Q} p(w|\theta_D)^{c(w,Q)}$ indicates the query-likelihood score.

Finally, a new query model is estimated by combining the original query model and RM1. Documents are re-scored and re-ranked using the new query model. RM3 [1] is a variant of a relevance model which is used here to estimate a new query model with RM1,

$$p(w|\theta'_Q) = (1-\beta) \cdot p(w|\theta_Q) + \beta \cdot p_{RM1}(w|Q), \tag{5}$$

where β is a control parameter between the original query model and the feedback model.

2.3 Word Vectors

Word2Vec [6] learns a vector representation for a word using a neural network language model. The resulting vector representations for words (i.e., word vectors) can be used in various tasks because a word is represented by a small-size vector. Learning the word vectors is entirely unsupervised and it can be computed on the text corpus according to purposes.

In our approach, Wikipedia was chosen to an input to train the Word2Vec. We assumed that non-medical pages are not useful to medical-related word vectors. Therefore, we just focused on medical pages by filtering out non-medical pages. To this end, first, categories were collected from a root to leaves. We set *Health/Diseases_and_disorders* and *Health/Health_care/Medicine* to the root because it is assumed that general medical queries want to find out information about diseases and treatments. This filtering procedure produced 7,672 categories. Then, all pages associated with those categories were used as input. The details of the medical Wikipedia pages are summarized at Table 1.

Table 1. Summary of medical Wikipedia pages

Categories	7,672
Pages	154,818
Sentences	5,575,390
Tokens	144,947,575
Voc. Size	1,387,935

We ran Word2Vec with CBOW archiecture and 200 for a size of a word vector. As a result, 1,387,935 X 200 matrix was constructed. It can be used for identifying synonym because the distance between two word vectors is similar.

2.4 Ranking with Word Vectors

Our approaches are based on PRF using the word vectors. In both approaches, reranking is performed with the initial documents obtained by a search engine.

In the first approach, new relevance scores are computed using the word vectors. To do that, cosine similarity is computed between Q and D by averaging associated word vectors respectively:

$$cosine(Q,D) = cosine\left(\frac{1}{|Q|}\sum_{q\in Q}\overrightarrow{W_q}, \frac{1}{|D|}\sum_{w\in D}\overrightarrow{W_w}\right)$$

Then, a new relevance score is computed by multiplying $cosine(Q,D)$ and $score_{QLD}(Q,D)$:

$$score_{WV}(Q,D) = score_{QLD}(Q,D) \cdot cosine(Q,D)$$

PRF is performed with $score_{WV}(Q,D)$. For detail, $p_{RM1}(w|Q)$ is estimated in Equation 4 and $score_{WV}(Q,D)$. In Equation 5, $p\left(w\middle|\theta_Q'\right)$ is constructed by combining $p_{RM1}(w|Q)$ and $p\left(w\middle|\theta_Q\right)$. Finally, re-ranking is performed with $p\left(w\middle|\theta_Q'\right)$ using Equation 1.

In the second approach, a query Q is directly expanded to Q_{WV} using the word vectors. To do that, $\overrightarrow{\mathrm{W}_Q}$, the average word vector for all query words, is computed. Then, cosine similarity is computed between $\overrightarrow{\mathrm{W}}_Q$ and $\overrightarrow{\mathrm{W}}_w$ where $w \in V_{WIKI}$. Top-5 words with high cosine similarity that don't appear in Q are chosen and added to Q_{WV}. Then, PRF is performed with Q_{WV} using Equations 1, 4, and 5.

3 Experiments

3.1 Data

This task used ClueWeb12-Disk-B (ClueWeb12B) collection which contains about 50M pages. Text of pages were extracted by removing HMTL tags using JSOUP[1] parser. Table 2 shows a summary of data statistics of ClueWeb12B.

Table 2. Data Statistics (The lengths are counted after stop-word removal.)

	ClueWeb12B
#Docs	51,563,645
Voc. Size	112,790,015
tokens	22,309,025,342
Avg. Doc. Len	432.7

3.2 Evaluation Settings

Lucene[2] was exploited to index and search the initial documents S. For text processing, Stop-words were removed using 419 stop-words[3] in INQUERY. $|S|$ was set to 2500 and obtained using QLD.

To generate the word vectors, Java version of Word2Vec[4] was used. CBOW architecture was used with 200 sized word vector. For input, we removed all punctuations and lowercased words without removing stop-words.

3.3 Results

We submitted three runs for this task. Run1 is our baseline while other two runs are our proposed approaches using the word vectors. Run2 is PRF with new relevance scores using the word vectors. Run3 is PRF with an expanded query using the word vectors.

[1] https://jsoup.org/

[2] http://lucene.apache.org/

[3] http://sourceforge.net/p/lemur/galago/ci/default/tree/core/src/main/resources/stopwords/inquery

[4] https://github.com/medallia/Word2VecJava

Table 3. Descriptions of our Submitted Runs

Run	Description
1	Scoring by KLD with RM1
2	Scoring by KLD with RM1 using $score_{WV}(Q,D)$
3	Scoring by KLD with RM1 using Q_{WV}

References

1. Abdul-Jaleel, N. et al.: UMass at TREC 2004: Novelty and HARD. In: Proceedings of Text REtrieval Conference (TREC). (2004).
2. Goeuriot, L. et al.: Overview of the CLEF eHealth Evaluation Lab 2015. In: CLEF 2015 - 6th Conference and Labs of the Evaluation Forum. Lecture Notes in Computer Science (LNCS), Springer (2015).
3. Kelly, Liadh and Goeuriot, Lorraine and Suominen, Hanna and Névéol, Aurélie and Palotti, Joao and Zuccon, G.: Overview of the CLEF eHealth Evaluation Lab 2016. In: CLEF 2016 - 7th Conference and Labs of the Evaluation Forum. Springer (2016).
4. Kelly, L. et al.: Overview of the ShARe/CLEF eHealth Evaluation Lab 2014. In: Proceedings of CLEF 2014. Springer (2014).
5. Kurland, O., Lee, L.: PageRank without hyperlinks: Structural re-ranking using links induced by language models. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’05. pp. 306–313 ACM Press, New York, New York, USA (2006).
6. Mikolov, T. et al.: Efficient Estimation of Word Representations in Vector Space. In: Proceedings of the International Conference on Learning Representations (ICLR 2013). pp. 1–12 (2013).
7. Oh, H.-S., Jung, Y.: Cluster-based query expansion using external collections in medical information retrieval. J. Biomed. Inform. 58, 70–79 (2015).
8. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22, 2, 179–214 (2004).
9. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the tenth international conference on Information and knowledge management. pp. 403–410 ACM, New York, New York, USA (2001).
10. Zuccon, G. et al.: The IR Task at the CLEF eHealth Evaluation Lab 2016: User-centred Health Information Retrieval. In: CLEF 2016 Evaluation Labs and Workshop: Online Working Notes. (2016).