Learning to Rank for Consumer Health Search: a Semantic Approach

Luca Soldaini and Nazli Goharian

Information Retrieval Lab, Georgetown University
Washington, DC, USA
{luca, nazli}@ir.cs.georgetown.edu

Abstract. For many internet users, searching for health advice online is the first step in seeking treatment. We present a Learning to Rank system that uses a novel set of syntactic and semantic features to improve consumer health search. Our approach was evaluated on the 2016 CLEF eHealth dataset, outperforming the best method by 26.6% in NDCG@10.

Keywords: learning to rank, medical search, consumer health search

1 Introduction

In recent years, the internet has become a primary resource for health information[1]. In searching medical information, lay people value access to trustworthy information [6], which has been shown to lead to a better understanding of health topics [9]. However, trustworthy health care resources—even those targeted at laypeople—use proper medical vocabulary, causing consumers to struggle [13].

In this paper, we propose a Learning to Rank (LtR) system that takes advantage of syntactic and semantic features to address the language gap between health seekers and medical resources. LtR algorithms have been successfully employed to improve retrieval of web pages [4]. In the health domain, they have been recently used to promote understandability in medical health queries [5] and retrieve medical literature [4]. The authors of this manuscript has previously experimented with the use of semantic relationships between terms in [7]. In this work we show how semantic features that capture the similarity between the query and retrieved documents can be effectively coupled with classic statistical features—such as those used in the LETOR dataset [4]—to promote relevant medical documents that answer consumer health queries.

Our approach is validated using the 2016 CLEF eHealth IR Task dataset [14], a collection of 300 medical queries designed to resemble laypeople health queries. Documents were retrieved from the category B subset of ClueWeb12[2]. We compared our approach to the best known baseline for this dataset, achieving a 26.6% improvement in terms of NDCG@10.

[1] http://www.pewinternet.org/2013/01/15/health-online-2013/
[2] http://lemurproject.org/clueweb12/

2 Methodology

2.1 Features

We proposed a combination of statistical and semantic features to train a LtR model. The feature set can be partitioned in five groups:

Statistical (STAT, 36 features): We considered a subset of features from the LETOR benchmark dataset, which have been shown to be useful in many LtR systems [4]. These features encode statistical information about the terms in the query and documents(e.g., term frequency (*tf*), inverse document frequency (*idf*)). We remand the reader to [4] for a complete list. We excluded some features because they are not available for our dataset (e.g., HITS scores). We also excluded all features that relied on the titles of webpages, as they showed poor correlation with relevance judgments in our tests.

Statistical Health (ST-HEALTH, 9 features): We expanded the set of statistical features by including health-specific features. We consider whether a document is certified by the *Health on Net Foundation*[3], an organization that publishes a code of good conduct for health websites. Such signal has been shown to be a good indicator of informative web sites [9]. We also extracted *tf* and *idf* of all terms in the document that can be found in the subset of health-related pages in Wikipedia, which were extracted following as in [9]. The average, variance, mode, and sum of *tf* and *idf* were used as features.

UMLS (UMLS, 26 features): The Unified Medical Language System[4] (UMLS) is a medical ontology maintained by the U.S. National Library of Medicine. Terms in this ontology are organized by concepts, each of which is associated with one or more semantic type. UMLS concepts are often present in queries issued by laypeople; thus, we explored their used as to identify relevant search results. To obtain the set of UMLS concepts in each document and in the query we used QuickUMLS [8], a medical concept extraction system. We match UMLS expressions belonging to 16 semantic types that are associated with symptoms, diagnostic tests, diagnoses, or treatments, as suggested in [8].

Latent Semantic Analysis (LSA, 2 features): To extract semantic relationships between terms, we built a 100-dimension Latent Semantic Analysis (LSA) model using a collection of 9,379 entries from the A.D.A.M. Medical Encyclopedia[5] (a consumer-oriented medical encyclopedia) and the MedScape[6] reference guide. The model was used to obtain vector representations of terms in the query and documents, which were summed using two strategies: simple sum and sum weighted by the probability of each term appearing in the health section of Wikipedia. This composition technique, while simple, has been shown to be very effective [1]. To extract LSA features, we computed the euclidean distance between the vector representing the query and the vector for the document. We used the similarity scores from the weighted and unweighted models as features.

[3] `https://www.healthonnet.org/`
[4] `https://www.nlm.nih.gov/research/umls/`
[5] `https://medlineplus.gov/encyclopedia.html`
[6] `http://reference.medscape.com/`

Word Embeddings (W2V, 4 features): Similarly to [3], we used word embeddings trained on PubMed[7] and Google News[8] to obtain dense vector representations for terms in the document and in the query. Word embeddings from the medical domain provide a strong representation for medical terms, while general domain word embeddings should capture the terms lay people are be more familiar with. As in LSA, we used a sum and a weighted sum to compose the term vectors into the vector representation of the document or query. In total, 4 features were extracted: weighted and unweighted similarities between document and query using PubMed and Google News models.

2.2 Ranking Algorithms

LtR algorithms are typically partitioned in three groups: point-wise, pair-wise, and list-wise learners. We considered the following LtR algorithms: logistic regression, random forests, LambdaMART [11], AdaRank [12], and ListNet [2]. Logistic regression and random forests are point-wise algorithms; we trained them to predict, for each document, its likelihood of being relevant. LambdaMART, a pair-wise learner, is an ensemble method that aims at minimizing the number of inversions in ranking. ListNet and AdaRank are list-wise learners that are designed to find a permutation of the retrieved results such that the value of a loss function on the list of results is minimized. We used the implementation of LambdaMART, AdaRank, and ListNet available in RankLib[9] v.2.7.

3 Experimental Setup

Dataset: The proposed LtR approach to laypeople medical search was evaluated on the 2016 CLEF eHealth IR Task dataset [14]. The dataset consists of 300 queries modeled after 50 distinct topics. The topics were created by health professional from forum posts from the *AskDocs* section of Reddit; Results for the queries were retrieved from the ClueWeb12 category B dataset, a collection of 53 million web pages. In total, 25,000 documents were evaluated; to each one, a score between 0 and 2 was assigned. Because all queries created from the same forum post share the same information need, relevance judgments of queries on the same topic are identical. On average, 74.1 documents were deemed relevant for each query (min: 1; max: 335; median: 45; std.dev.: 74.7).
Experiments: Documents were indexed using the Terrier search engine, v. 4.0[10]. As a baseline, we consider the BM25 scoring function defined by the CLEF eHealth organizers in [14]. While simple, this baseline outperformed all 10 teams (29 runs) who participated in shared task. We use this baseline to retrieve up to 1,000 documents per query to train the LtR methods. All learners were trained under five fold cross validation and manually tuned using a separate validation

[7] https://github.com/cambridgeltl/BioNLP-2016/
[8] https://code.google.com/archive/p/word2vec/
[9] https://sourceforge.net/p/lemur/wiki/RankLib/
[10] http://terrier.org/

Method	Type of approach	NDCG@10		P@10	
BM25 baseline [14]	*n/a*	0.241		0.291	
Random Forests	point-wise	0.249	(+3.3%)	0.293	(+0.6%)
Logistic Regression	point-wise	0.262*	(+8.7%)	0.317*	(+8.9%)
LambdaMART [11]	pair-wise	**0.305***	**(+26.6%)**	**0.361***	**(+24.1%)**
AdaRank [12]	list-wise	0.239	(-0.8%)	0.292	(- 0.7%)
ListNet [2]	list-wise	0.267*	(+10.8%)	0.333*	(+ 14.4%)

Table 1. Performance of LtR algorithms on the dataset. Runs marked with * are significantly different from the baseline (Paired Student's t-test, $p < 0.05$).

set. Pair-wise and list-wise learners were configured to optimize NDCG@10 on the validation set. To avoid overfitting, we carefully generated the training, validation, and test set so that all queries from the same group are part of the same split. Finally, P@10 and NDCG@10 were used to evaluate all the approaches, as users of online search engines are more likely to pay attention to the first page of retrieved results than the subsequent ones.

4 Results

4.1 LtR Algorithms

We compare the LtR approaches from Section 2.2 with the baseline used in [14]. For all experiments, learners are trained on all the features described in Section 2.1; we will study the impact of individual features in Section 4.2.

Of all learners reported in Table 1, LambdaMART achieves the best performance (+26.6% NDCG@10, +24.1% P@10 over the baseline). This demonstrates that LtR can be successfully exploited to improve the access to relevant medical resources that satisfy the need of online health seekers. As expected, LambdaMART outperforms point-wise LtR approaches, as it is often the case [4]. LambdaMART also achieves better performance than the two list-wise methods, AdaRank and ListNet (difference is statistically significant for both). This is to be expected, as previous work found LambdaMART to be very competitive in LtR tasks on web results when optimizing for NDCG@10 [10].

4.2 Feature Analysis

The performance of the model trained on each set of features is presented in Table 2. We observe that the model trained only on the statistical features (STAT) obtains better performances than models trained on other sets of features. This is to be expected, as statistical features were modeled after the LETOR feature set, which has been shown to be very effective for LtR tasks [4]. The model trained solely on statistical health features (ST-HEALTH) ranks second, suggesting that the presence and frequency of health terms plays an important role in identifying relevant results. This intuition is reinforced by the findings shown in Table 3, where ST-HEALTH features are among the highest ranked in terms of importance.

The UMLS features set shows limited improvements over the BM25 baseline. However, based on their ranking in Table 3, we argue that they have an important role in model built using all features, as they capture information about symptoms and diseases mentioned in the queries.

Features group	NDCG@10	P@10
BM25 baseline	0.241	0.291
STAT	0.274	0.322
ST-HEALTH	0.260	0.311
UMLS	0.253	0.307
W2V	0.160	0.210
LSA	0.121	0.188
All features	**0.305**	**0.361**

Table 2. Performance of LambdaMART trained on each set of features separately. All runs are significantly different from the best method (Paired Student's t-test, $p < 0.05$).

Feature	Group	Weight
Avg. *idf* in health Wikipedia	ST-HEALTH	0.0995
# of matching UMLS concepts in document	UMLS	0.0776
Avg. *tf* in health Wikipedia	ST-HEALTH	0.0616
BM25 similarity score	STAT	0.0605
# concepts in "*Sign or Symptom*" UMLS semantic type	UMLS	0.0579
Similarity weighted word embeddings PubMed	W2V	0.0521
# concepts in "*Injury or Poisoning*" UMLS semantic type	UMLS	0.0418
LM similarity score	STAT	0.0408
Similarity weighted word embeddings Google News	W2V	0.0393
Spam scores	STAT	0.0335

Table 3. Top 10 features ranked by weight (normalized). The weight of each feature was computed by averaging their information gain.

Lastly, we note that neither word embedding similarity features (W2V) nor latent semantic analysis similarity features (LSA) features are enough to train an effective LtR model by themselves. This outcome could be due to the fact that these features sets, which contain just 4 and 2 features, do not encode enough information to train a comprehensive model. However, while W2V features improve the effectiveness of the model when combined with other features (Table 3), LSA features have less of an impact on the model built by LambdaMART. This might be due to the fact that the set of 9,379 pages the LSA model was trained on is too small to capture the semantic similarity between queries and the retrieved documents. Conversely, similarity features derived by dense word representations are effective for this task as long as the model used to derive them is accurate.

4.3 Query Performance

In this section, we compare the per-query performance of the baseline with the best ranker from Table 1. Results are shown in Figure 1. Rather than reporting the individual NDCG@10 for each query, we average the results of all queries that belong to the same query group. This approach is motivated by the fact that all queries in the same group share the same information need (and document relevance judgments). Therefore, by averaging the performance of all queries in the same group, we can study whether the performance of the best ranker relative to the baseline is due to the information need associated with each query. To convince the reader that this representation is justified, the variance for each query group is shown in Figure 1. As the variance for each topic is moderate, we conclude that our approach is appropriate.

The proposed ranker outperforms the baseline on 36 out of 50 topics. Interestingly, LambdaMART outperforms the baseline in all but one query whose NDCG@10 is below median. In other words, there exists a statistically significant correlation between the performance of the baseline on each query and the difference between the NDCG@10 of the baseline and LambdaMART (Spearman's

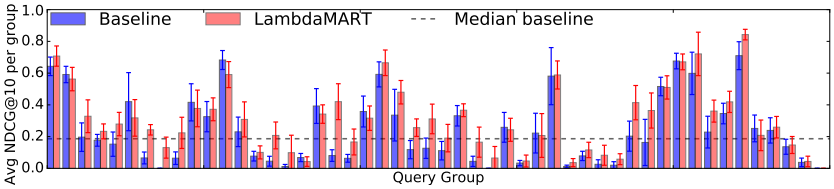

Fig. 1. NDCG@10 of the baseline and the best performing method of Table 1. To increase the clarity of the figure, we averaged the value of NDCG@10 of all queries from the same query group (i.e., all queries sharing the same information need.)

rank correlation, $r_s = -0.38$, $p < 0.05$). This suggests that LtR is a viable strategy for addressing difficult queries; however, its performance are still bounded by the quality of results retrieved by the baseline.

5 Conclusions

In this paper we proposed a novel set of syntactic and semantic features for LtR for consumer health queries. The proposed approach led to a 26.2% increase in NDCG@10 over existing methods. The impact of several Learning to Rank algorithms was studied; furthermore, we discussed the effectiveness of our proposed features. This work demonstrates that semantic features can be effectively exploited for LtR in laypeople health search.

References

1. W. Blacoe and M. Lapata. A comparison of vector-based representations for semantic composition. In ACL. Association for Computational Linguistics, 2012.
2. Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In ICML, 2007.
3. A. Cohan, A. Fong, N. Goharian, and R. Ratwani. A neural attention model for categorizing patient safety events. In ECIR, 2017.
4. T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. LETOR: benchmark dataset for research on learning to rank for information retrieval. In LTR *workshop,* SIGIR, 2007.
5. J. Palotti, L. Goeuriot, G. Zuccon, and A. Hanbury. Ranking health web pages with relevance and understandability. In SIGIR, 2016.
6. J. Palotti, A. Hanbury, H. Müller, and C. E. Kahn. How users search and what they search for in the medical domain. *IR Journal*, 2016.
7. L. Soldaini, W. Edman, and N. Goharian. Team gu-irlab at clef ehealth 2016: Task 3. In CLEF, 2016.
8. L. Soldaini and N. Goharian. QuickUMLS: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop,* SIGIR, 2016.
9. L. Soldaini, A. Yates, E. Yom-Tov, O. Frieder, and N. Goharian. Enhancing web search in the medical domain via query clarification. *IR Journal*, 2016.
10. N. Tax, S. Bockting, and D. Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Inf. Proc. Manag.*, 2015.
11. Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *IR Journal*, 2010.
12. J. Xu and H. Li. AdaRank: a boosting algorithm for information retrieval. In SIGIR, 2007.
13. Q. T. Zeng, S. Kogan, R. M. Plovnick, J. Crowell, E.-M. Lacroix, and R. A. Greenes. Positive attitudes and failed queries: an exploration of the conundrums of consumer health information retrieval. *J. Med. Inf.*, 2004.
14. G. Zuccon, J. Palotti, L. Goeuriot, L. Kelly, M. Lupu, P. Pecina, H. Mueller, J. Budaher, and A. Deacon. The IR task at the CLEF ehealth evaluation lab 2016: User-centred health information retrieval. In CLEF, 2016.