A Full-Text Learning to Rank Dataset for Medical Information Retrieval

Vera Boteva[1], Demian Gholipour[1], Artem Sokolov[1], and Stefan Riezler[1,2]

Computational Linguistics[1] and IWR[2], Heidelberg University, Heidelberg, Germany
{boteva,gholipour,sokolov,riezler}@cl.uni-heidelberg.de

Abstract. We present a dataset for learning to rank in the medical domain, consisting of thousands of full-text queries that are linked to thousands of research articles. The queries are taken from health topics described in layman's English on the non-commercial NutritionFacts.org website; relevance links are extracted at 3 levels from direct and indirect links of queries to research articles on PubMed. We demonstrate that ranking models trained on this dataset by far outperform standard bag-of-words retrieval models. The dataset can be downloaded from: www.cl.uni-heidelberg.de/statnlpgroup/nfcorpus/.

1 Introduction

Health-related content is available in information archives as diverse as the general web, scientific publication archives, or patient records of hospitals. A similar diversity can be found among users of medical information, ranging from members of the general public searching the web for information about illnesses, researchers exploring the PubMed database[1], or patent professionals querying patent databases for prior art in the medical domain[2]. The diversity of information needs, the variety of medical knowledge, and the varying language skills of users [4] results in a lexical gap between user queries and medical information that complicates information retrieval in the medical domain.

In this paper, we present a dataset that bridges this lexical gap by exploiting links between queries written in layman's English to scientific articles as provided on the NutritionFacts.org (NF) website. NF is a non-commercial, public service provided by Dr. Michael Greger and collaborators who review state-of-the-art nutrition research papers and provide transcribed videos, blog articles and Q&A about nutrition and health for the general public. NF content is linked to scientific papers that are mainly hosted on the PubMed database. By extracting relevance links at three levels from direct and indirect links of queries to research articles, we obtain a database that can be used to directly learn ranking models for medical information retrieval. To our knowledge this is the first dataset that provides full texts for thousands of relevance-linked queries

[1] www.ncbi.nlm.nih.gov/pubmed

[2] For example, the USPTO and EPO provide specialized patent search facilities at www.uspto.gov/patents/process/search and www.epo.org/searching.html

and documents in the medical domain. In order to showcase the potential use of our dataset, we present experiments on training ranking models, and find that they significantly outperform standard bag-of-words retrieval models.

2 Related Work

Learning-to-rank algorithms require a large amount of relevance-linked query-document pairs for supervised training of high capacity machine learning models. Such datasets have been made public[3] by search engine companies, comprising tens of thousands of queries and hundreds of thousands of documents at up to 5 relevance levels. The disadvantage of these datasets is the fact that they do not provide full texts but only pre-processed feature vectors. They are thus useful to compare ranking algorithms for given feature representations, but are of limited use for the development of complete learning approaches. Furthermore, Ohsumed, the only learning-to-rank dataset in the medical domain, contains only about a hundred of queries. A dataset for medical information retrieval comprising full texts has been made public[4] at the CLEF eHealth evaluations. This dataset contains approximately one million documents from medical and health domains, but only 55 queries, which makes this dataset too small for training learning-to-rank systems. Large full text learning-to-rank datasets for domains such as patents or Wikipedia have been used and partially made publicly available[5]. Similar to these datasets, the corpus presented in this paper contains full-text queries and abstracts of documents, annotated with automatically extracted relevance links at several levels (here: 3). The proposed dataset is considerably smaller than the above mentioned datasets from the patent and Wikipedia domain, however, it still comprises thousands of queries and documents.

3 Corpus Creation Methodology

The NF website contains three different content sources – videos, blogs, and Q&A posts, all written in layman's English, which we used to extract queries of different length and language style. Both the internal linking structure and the scientific papers citations establish graded relevance relations between pieces of NF content and scientific papers. Additionally, the internal NF topic taxonomy, used to categorize similar NF content that is not necessarily interlinked, is exploited to define the weakest relevance grade.

Crawling queries and documents. The following text sections of NF content pages were extracted:

[3] `research.microsoft.com/en-us/um/beijing/projects/letor`, `research.microsoft.com/en-us/projects/mslr`, `webscope.sandbox.yahoo.com`

[4] `clefehealth2014.dcu.ie/task-3`

[5] `www.cl.uni-heidelberg.de/statnlpgroup/{boostclir|wikiclir}`

- **Videos**: *title*, *description* (short summary), *transcript* (complete transcript of the audio track), *"doctor's note"* (short remarks and links to related NF content), *topics* (content tags), *sources* (URLs to medical articles), *comments* (user comments).
- **Blog articles** (usually summaries of a series of videos): *title*, *text* (includes links to other NF pages and medical articles), *topics*, *comments*.
- **Q&A**: *title*, *text* (the question and an answer with links to related NF pages and medical articles), *comments*.
- **Topic** pages listing NF material tagged with the topic: *title*, *text* (may include a topic definition, with links to NF content but not to medical articles).

Medical documents were crawled following direct links from the NF pages to:

- **PubMed**, where 86% of all links led,
- **PMC** (PubMed Central) with 3%,
- Neither PubMed nor PMC pages, i.e. links to pages of medical journals, 7%,
- Direct links to PDF documents, 4%.

Since PubMed pages could further link to full-texts on PMC and since extracting abstracts from these two types of pages was the least error-prone, we included titles and abstracts of only these two types into the documents side of the corpus.

Data. We focused on 5 types of queries that differ by length and well-formedness of the language. In particular we tested full queries, i.e., *all fields* of NF pages concatenated: titles, descriptions, topics, transcripts and comments), *all titles* of NF content pages, *titles of non-topic pages* (i.e., titles of all NF pages except topic pages), *video titles* (titles of video pages) and *video descriptions* (description from videos pages). The latter three types of queries often resemble queries an average user would type (e.g., "*How to Treat Kidney Stones with Diet*" or "*Meat Hormones and Female Infertility*"), unlike *all titles* that include headers of topics pages that often consist of just one word.

For each relevance link between a query and a document we randomly assigned 80% of them to the training set and 10% for dev and test subsets. Retrieval was performed over the full set of abstracts (3,633 in total, mean/median number of tokens was 147.1/76.0). Note that this makes the test PubMed abstracts (but not the queries) available during training. The same methodology was used in [1] who found that it only marginally affected evaluation results compared to the setting without overlaps. Basic statistics about the different query types are summarized in Table 1.

Extracting relevance links. We defined a special relation between queries and documents that did not exist in the explicit NF link structure. A directly linked document of query $\mathbf{q}$ is considered *marginally relevant* for query $\mathbf{q}'$ if the containment $|t(\mathbf{q}) \cap t(\mathbf{q}')|/|t(\mathbf{q})|$ between the sets of topics with which the queries are tagged is at least 70%. In general this relation may be considered as still weakly relevant and be preferred to, say, some completely out-of-domain (e.g. nutrition-unrelated) document from PubMed. However, we treat such documents

Table 1. Statistics of relevance-linked ranking data (without stop-word filtering).

type	# queries	mean/median # tokens per query	mean # docs per query lev. 2	lev. 1	lev. 0
all fields	3244	1890.0/43.5	4.6	41.6	33.8
all titles	3244	3.6/ 1.5	4.6	41.6	33.8
titles of non-topic pages	1429	6.0/ 4.0	4.6	25.4	26.3
video titles	1016	5.5/ 6.0	4.9	23.6	27.1
video descriptions	1016	24.3/21.0	4.9	23.6	27.1

as irrelevant but still in-domain in training and and testing. The rationale is that we are mostly interested in learning a thin line between relevant and similar but yet irrelevant documents, as opposed to a simpler task of discerning them from completely out-of-domain documents.

We assign relevance levels to a query $\mathbf{q}$ with respect to a scientific document $\mathbf{d}$ from three possible values: The most relevant level (2) corresponds to a direct link from $\mathbf{q}$ to $\mathbf{d}$ from the cited sources section of a page, the next level (1) is used if there exists another query $\mathbf{q}'$ that directly links to $\mathbf{d}$ and also $\mathbf{q}$'s text contains an internal link to $\mathbf{q}'$. Finally, the lowest level of (0) is reserved for every marginally relevant $\mathbf{q}'$ and document $\mathbf{d}$.

Finally, once all links are known we excluded queries that wouldn't be of any use for learning, like queries without any text (e.g., many topic pages) and queries with no direct, indirect, or topic-based links to any documents.

4 Experiments

Systems. Our two baseline retrieval systems use the classical ranking scores: *tfidf* and Okapi *BM25*[6]. In addition, we evaluated two learning to rank approaches that are based on a matrix of query words times document words as feature representation, and optimize a pairwise ranking objective [1, 7]: Let $\mathbf{q} \in \{0,1\}^Q$ be a query and $\mathbf{d} \in \{0,1\}^D$ be a document, where the n^{th} vector dimension indicates the simple occurrence of the n^{th} word for dictionaries of size Q and D. Both approaches learn a score function $f(\mathbf{q},\mathbf{d}) = \mathbf{q}^\top W \mathbf{d} = \sum_{i=1}^{Q}\sum_{j=1}^{D} q_i W_{ij} d_j$, where $W \in \mathbb{R}^{Q\times D}$ encodes a matrix of word associations. Optimal values of W are found by pairwise ranking given supervision data in the form of a set $\mathcal{R}$ of tuples $(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)$, where $\mathbf{d}^+$ is a relevant (or higher ranked) document and $\mathbf{d}^-$ an irrelevant (or lower ranked) document for query $\mathbf{q}$, the goal is to find W such that an inequality $f(\mathbf{q},\mathbf{d}^+) > f(\mathbf{q},\mathbf{d}^-)$ is violated for the fewest number of tuples from $\mathcal{R}$. Thus, the goal is to learn weights for all domain-specific associations of query terms and document terms that are useful to discern relevant from irrelevant documents by optimizing the ranking objectives defined below.

[6] BM25 parameters were set to $k_1 = 1.2$, $b = 0.75$.

Table 2. MAP/NDCG results evaluated for different types of queries. Best NDCG results of learning-to-rank versus bag-of-words models are highlighted in **bold face**.

queries	RankBoost	SGD	tfidf	bm25
all fields	0.2632/0.5073	0.3831/**0.6064**	0.1360/0.3932	0.1627/**0.4169**
all titles	0.1549/**0.3475**	0.1360/0.3454	0.1233/0.2578	0.1251/**0.2582**
titles of non-topic pages	0.1615/**0.4039**	0.1775/0.3790	0.0972/0.2851	0.1124/**0.3032**
video descriptions	0.1312/**0.3826**	0.1060/0.3112	0.1110/0.3509	0.1262/**0.3765**
video titles	0.1350/**0.3804**	0.1079/0.3109	0.1010/0.2873	0.1127/**0.3042**

The first method [7] applies the *RankBoost* algorithm [2], where $f(\mathbf{q}, \mathbf{d})$ is a weighted linear combination of T functions h_t such that $f(\mathbf{q}, \mathbf{d}) = \sum_{t=1}^{T} w_t h_t(\mathbf{q}, \mathbf{d})$. Here h_t is an indicator that selects a pair of query and document words. Given differences of query-document relevance ranks $m(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-) = r_{\mathbf{q},\mathbf{d}^+} - r_{\mathbf{q},\mathbf{d}^-}$, RankBoost achieves correct ranking of $\mathcal{R}$ by optimizing the exponential loss

$$\mathcal{L}_{exp} = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)\in\mathcal{R}} m(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-) e^{f(\mathbf{q},\mathbf{d}^-) - f(\mathbf{q},\mathbf{d}^+)}.$$

The algorithm combines batch boosting with bagging over independently drawn 10 bootstrap data samples from $\mathcal{R}$, each consisting of 100k instances. In every step, the single word pair feature h_t is selected that provides the largest decrease of $\mathcal{L}_{exp}$. The resulting models are averaged as a final scoring function. To reduce memory requirements we used random feature hashing with the size of the hash of 30 bits [5]. For regularization we rely on early stopping ($T = 5000$). An additional fixed-weight identity feature is introduced that indicates the identity of terms in query and document; its weight was tuned on the dev set.

The second method uses *stochastic gradient descent* (SGD) as implemented in the Vowpal Wabbit (VW) toolkit [3] to optimize the ℓ_1-regularized hinge loss:

$$\mathcal{L}_{hng} = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)\in\mathcal{R}} \big(f(\mathbf{q}, \mathbf{d}^+) - f(\mathbf{q}, \mathbf{d}^-)\big)_+ + \lambda ||W||_1,$$

where $(x)_+ = \max(0, m(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-) - x)$ and λ is the regularization parameter. VW was run on the same (concatenated) samples as the *RankBoost* using the same number of hashing bits. On each step, W is updated with a scaled gradient vector $\nabla_W \mathcal{L}_{hng}$ and clipped to account for ℓ_1-regularization; λ and the number of passes over the data were tuned on the dev set.

Experimental results. Results according to the MAP and NDCG metrics on pre-processed data[7] are reported in the Table 2. Result differences between the best performing learning-to-rank versus bag-of-words models were found to be statistically significant [6]. As results show, learning-to-rank approaches outperform

[7] Preprocessing included lowercasing, tokenizing, filtering punctuation and stop-words, and replacing numbers with a special token.

classical retrieval methods by a large margin, proving that the provided corpus is sufficient to optimize domain-specific word associations for a direct ranking objective. As shown in row 1 of Table 2, the *SGD* approach outperforms *RankBoost* in the evaluation on *all fields* queries, but performs worse with shorter (and fewer) queries as in the setups listed in rows 2-5. This is due to a special "pass-through" feature implemented in *RankBoost* that assigns a default feature to word identities, thus allowing to learn better from sparser data. The *SGD* implementation does not take advantage of such a feature, but it makes a better use of the full matrix of word associations which offsets the lacking pass-through if enough word combinations are observable in the data.

5 Conclusion

We presented a dataset for learning to rank in the medical domain that has the following key features: 1) full text queries of various length, thus enabling the development of complete learning models; 2) relevance links at 3 levels for thousands of queries in layman's English to documents consisting of abstracts of research article; 3) public availability of the dataset (with links to full documents for research articles). We showed in an experimental evaluation that the size of the dataset is sufficient to learn ranking models based on sparse word association matrices that outperform standard bag-of-words retrieval models.

Acknowledgments. We are grateful to Dr. Michael Greger for permitting crawling `NutritionFacts.org`. This research was supported in part by DFG grant RI-2221/1-2 "Weakly Supervised Learning of Cross-Lingual Systems".

6 References

[1] Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., Chapelle, O., Weinberger, K.: Learning to rank with (a lot of) word features. Information Retrieval Journal 13(3), 291–314 (2010)
[2] Collins, M., Koo, T.: Discriminative reranking for natural language parsing. Computational Linguistics 31(1), 25–69 (2005)
[3] Goel, S., Langford, J., Strehl, A.L.: Predictive indexing for fast search. In: NIPS. Vancouver, Canada (2008)
[4] Goeuriot, L., Kelly, L., Jones, G.J.F., Müller, H., Zobel, J.: Report on the SIGIR 2014 workshop on medical information retrieval (MedIR). SIGIR Forum 48(2), 78–82 (2014)
[5] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A.J., Strehl, A.L., Vishwanathan, V.: Hash Kernels. In: AISTATS. Irvine, CA (2009)
[6] Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM. Lisbon, Portugal (2007)
[7] Sokolov, A., Jehl, L., Hieber, F., Riezler, S.: Boosting cross-language retrieval by learning bilingual phrase associations from relevance rankings. In: EMNLP. Seattle, WA (2013)