Learning to Selectively Rank Patients' Medical History

Nut Limsopatham[1], Craig Macdonald[2], Iadh Ounis[2]
nutli@dcs.gla.ac.uk[1], firstname.lastname@glasgow.ac.uk[2]
School of Computing Science
University of Glasgow, Glasgow, UK

ABSTRACT

Two main approaches have emerged in the literature for the effective deployment of a search system to rank patients having a medical history relevant to a query. The first approach is to directly rank patients based on the relevance of their medical history, represented as a concatenation of their associated medical records. Instead, the second approach initially retrieves the relevant medical records of patients, and then ranks the patients based on the relevance of their retrieved medical records. However, these two approaches may be useful for different queries. In this work, we propose a novel supervised approach that can effectively identify when to use either of the two aforementioned patient ranking approaches to attain effective retrieval performance. In particular, our approach deploys a classifier to learn to select a ranking approach when ranking patients, by using query difficulty measures, such as query performance predictors and the number of medical concepts detected in a query, as learning features. We thoroughly evaluate our approach using the standard test collections provided by the TREC Medical Records track. Our results show significant improvements over existing strong baselines.

Categories and Subject Descriptors: H.3.3 [Information Search & Retrieval]: Search process

Keywords: Medical Records Search; Selective Ranking Approach; Regression-trees

1. INTRODUCTION

Electronic medical records are increasingly used to improve the quality of healthcare services and the patients safety [8]. For example, these medical records can be used to find patients who have a medical history relevant to a so-called inclusion criteria, in order to possibly recruit them for a clinical trial [18, 19]. Specifically, when conducting a comparative effectiveness research of a particular medical procedure (e.g. a treatment), a set of inclusion criteria, including the medical conditions of patients such as diagnoses and symptoms, are developed and used to search for patients with such conditions. In order to facilitate this process, an effective information retrieval (IR) system is used for identifying the patients whose medical records are relevant to these inclusion criteria.

In the literature, two main types of approaches are used to rank patients based on the relevance of their medical history (e.g. [5, 9, 10, 11, 13, 21]). The first approach (e.g. [5, 9]), which we refer to as the *patient model*, represents a patient using all of the medical records of that patient. In particular, this approach naturally represents patients using all of their medical history as a unit of retrieval [5]. For example, Demner-Fushman et al. [5] and King et al. [9], whose systems achieved the highest retrieval effectiveness among the TREC 2011 participants, concatenated the medical records of each patient into a *history document* and used the resulting history documents for indexing and retrieval.

On the other hand, the second approach (e.g. [10, 13, 21]), referred to as the *document model*, uses a medical record as a unit of retrieval, alleviating the problem of the variation in the size of the patients' medical history. Indeed, the second approach firstly ranks medical records based on their relevance towards the query, and then the relevance scores of the retrieved medical records are aggregated to estimate the relevance of their associated patients [13]. For example, Limsopatham et al. [13] effectively deployed the expCombSUM voting technique [14], which had previously been developed for expert search, to estimate the relevance of patients based on the relevance scores of their associated medical records.

However, it is not clear in the literature under which conditions a particular ranking approach should be deployed to rank patients. Recently, Zhu and Carterette [21] proposed to use a data fusion technique [16], such as CombSUM and CombMAX, to merge the relevance scores from both the *patient model* and the *document model* in order to exploit the effectiveness of both patient ranking approaches. Instead, we hypothesise that a selective approach that can appropriately identify which of the two ranking approaches is more effective for a given query can further improve retrieval performance.

In this work, we propose a novel selective approach for ranking patients based on the relevance of their medical history. In particular, we postulate that some queries are better served with different patient ranking approaches. Therefore, our proposed approach aims to effectively apply a ranking approach that can accomplish a better retrieval performance for a particular query. Specifically, we deploy a regression-trees classifier to learn how to select a ranking approach using query difficulty measures such as AvIDF [3].

We evaluate our proposed approach in the context of the TREC 2011 and 2012 Medical Records track [18, 19]. Our results show that our selective approach to rank patients is

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2507874.

effective. In particular, it significantly outperforms existing effective baselines, such as, when the relevance scores of both models are combined using data fusion techniques.

The main contributions of this paper are three-fold:

1. We introduce an approach that effectively selects which of the two approaches (i.e. either the patient or the document model) should be used for a particular query.
2. We propose to exploit query difficulty measures, such as query performance predictors, as learning features.
3. We thoroughly evaluate our approach using the standard experimental setup provided by the TREC Medical Records track.

The remainder of the paper is organised as follows. Section 2 introduces our novel classification approach that selectively applies either the patient or the document model when ranking patients on a per-query basis. Sections 3, and 4 discuss our experimental setup and results. Finally, we provide concluding remarks in Section 5.

2. A SELECTIVE RANKING APPROACH

In this section, we describe our selective ranking approach, which applies on each query either the patient model or the document model when ranking patients. As mentioned above, we hypothesise that queries benefit differently from the patient model or the document model. Indeed, some queries might be better served by the patient model, while others might be better handled by the document model. We propose an automatic decision mechanism that chooses to apply either the patient model or the document model for a particular query. Specifically, we propose to deploy a regression-trees classifier that, given a particular query, selects a specific patient ranking approach using learning features, such as query performance predictors. Our proposed approach consists of three components: (1) learning features; (2) the learned decision mechanism; and (3) the learning process. The remainder of this section discusses each of the components of our proposed selective ranking approach.

2.1 Learning Features

We first describe the learning features used in this paper. An effective learning feature should correlate well with the labelled data in a training set and should generalise across queries. To effectively select between the patient model and the document model, we learn a classifier using the query features listed in Table 1. Indeed, these features, which measure the difficulty of a query, can be categorised into two groups. The first group (Features 1-10) measures the *relative difference* of the query performance predictor scores between the patient model and the document model. We choose to use the obtained scores as our learning features, since we hypothesise that to attain an effective retrieval performance, a search system should deploy the ranking approach that finds the query the least difficult. The intuition is that an easy query leads to a better retrieval performance. For example, Features 1-4, including the clarity score [4], SCQ [20], MAXCQ [20] and NSCQ [20], measure the ambiguity of a query based on the coherence of the language used in a particular set of documents. If the query model is similar to the language model of the collection, an effective retrieval performance is likely expected. Features 5-8 measure the specificity of a query for a document collection. The more specific the query, the better retrieval performance is expected. These features are AvICTF [3], AvIDF [3], EnIDF [3], and Query Scope (ω) [7]. Features 9-10, γ_1 [7] and γ_2 [7], examine the distribution of the informativeness among the query terms. The more a query term is informative, the better the retrieval effectiveness.

Table 1: List of the query features used by our classifier to decide to apply either the patient model or the document model.

ID	Feature
1	Clarity Score[1] [4]
2	SCQ[1] [20]
3	MAXCQ[1] [20]
4	NSCQ [20]
5	AvICTF[1] [3]
6	AvIDF[1] [3]
7	EnIDF[1] [3]
8	Query Scope (ω)[1] [7]
9	γ_1[1] [7]
10	γ_2[1] [7]
11	Query length [7]
12	Number of detected medical concepts in a query
13	Occurrence probability of symptom concepts in a query
14	Occurrence probability of diagnostic-test concepts in a query
15	Occurrence probability of diagnosis concepts in a query
16	Occurrence probability of treatment concepts in a query

On the other hand, the second group of features (Features 11-16) measures the difficulty of a query by using the information from the query itself. Feature 11 is the number of non-stopword query terms. Moreover, since medical queries normally focus on four aspects of the medical decision criteria (namely: symptom, diagnostic test, diagnosis and treatment) [12], Features 12-16 are based on the occurrences of the medical concepts associated with these four aforementioned aspects. In particular, Feature 12 is the number of the concepts related to the medical decision criteria, which can be extracted from the query. Specifically, following Limsopatham et al. [12], we deploy MetaMap [2] – a medical concept detection tool – to identify those medical concepts in the query[2]. Features 13, 14, 15, and 16 estimate the probability that the medical concepts detected in a query are related to symptom, diagnostic test, diagnosis and treatment, respectively. The probability is estimated using the maximum likelihood by counting the number of medical concepts detected in the query. We hypothesise that the more medical concepts are detected in the query, the more likely that the query is difficult.

2.2 The Learned Decision Mechanism

In this section, we describe our decision mechanism, which is based on a learned classifier. In particular, the classifier decides to apply the patient model or the document model for a particular query using the introduced query features. By doing so, we benefit from the fact that several query difficulty measures, which are used as learning features, can be taken into account when choosing an effective patient ranking approach.

While any classifier can be deployed, in this work, we use the Gradient Boosted Regression Trees (GBRT) [17] classifier to effectively decide when to apply the patient model or the document model for a given query, because of its simplicity and its effectiveness in several tasks (e.g. [6, 11, 17]). Specifically, we deploy the default setting of GBRT as implemented in the jforests package [6][3].

[1]The difference between the values of the feature computed on the patient model and the document model.

[2]We only use the concepts with the highest scores from MetaMap (i.e. indicated as 'Meta Mapping').

[3]`http://code.google.com/p/jforests`

2.3 The Learning Process

To effectively train the classifier, on a training set, we label each query with 1, if the patient model can achieve a better retrieval performance on a particular target measure; otherwise, it is labelled -1. This allows the classifier to learn which ranking approach is more effective for a particular query. Next, when training the GBRT classifier, we use the obtained accuracy as the loss function. We deploy the query difficulty measures extracted for each query, discussed in Section 2.1, as learning features to train the classifier.

3. EXPERIMENTAL SETUP

This section discusses the experimental setup for evaluating our selective ranking approach. Indeed, Sections 3.1 and 3.2 describe the used test collections, and our ranking strategies, respectively.

3.1 Test Collections

We evaluate our proposed approach using the TREC 2011 and 2012 Medical Records track test collections [18, 19]. The task is to retrieve *patient visits* relevant to a query. Each patient visit contains all medical records associated to a patient's visit to a hospital. Due to the privacy concerns [19], a patient visit is used to represent a patient. The collection contains 101,711 medical records, which can be mapped into 17,265 patient visits. The official measure of TREC 2011 is bpref, while the official measures of TREC 2012 are infAP and infNDCG, because of the deemed possible incompleteness of the gold-standard relevance judgements [18, 19].

3.2 Ranking Approaches

We conduct experiments using Terrier [15][4], applying Porter's English stemmer and removing stopwords. In addition, since dealing with negated language has been shown both in TREC 2011 and TREC 2012 to be useful for this task, we follow [10] and tokenise terms differently depending on whether they appear in a positive or negative context. For example, the term 'fever' is represented as 'fever' or 'n$fever', depending on whether it appears in sentences such as 'high fever' or 'no fever observed', respectively.

3.2.1 Patient Model

As a well-established representative patient model, we follow [9] and concatenate the medical records associated to the same patient visit into a large *visit document*. We use these visit documents to represent all the patient visits. Indeed, we index these visit documents using Terrier, and apply the effective parameter-free DPH weighting model [1] for ranking the patient visits.

3.2.2 Document Model

As a representative of the document model, we follow [13] and deploy the expCombSUM voting technique [14] when ranking patient visits, since it has been shown to be effective for this task both in TREC 2011 and TREC 2012 [10, 13]. Specifically, the parameter-free DPH weighting model is used to firstly rank medical records. Then, the relevance scores of the retrieved medical records are aggregated to estimate the relevance scores of their associated patients. In particular, expCombSUM calculates the relevance score of a patient visit v towards a query Q as follows [13]:

$$score\_visit_{expCombSUM}(v, Q) = \sum_{d \in R(Q) \cap profile(v)} e^{score(d,Q)} \quad (1)$$

where $R(Q) \cap profile(v)$ is the set of medical records associated to the patient visit v that are also in the ranking $R(Q)$; $score(d, Q)$ is the relevance score of medical record d for query Q, as obtained from the DPH model. In this work, we follow [13] and limit the number of the medical records voting for the relevance of patient visits ($|R(Q)|$) to 5,000.

3.2.3 Selectively Ranking the Patients

To evaluate our proposed selective ranking approach, we use a 5-fold cross-validation across the 34 topics of TREC 2011 and the 47 topics of TREC 2012, where each fold has completely separated training and test query sets. We separate the set of queries used in the two years of TREC, because the used relevance assessment mechanism in each query set is different. Indeed, TREC 2011 deploys absolute judgement, while TREC 2012 uses a sampling technique [18, 19]. When labelling the training set, we target the retrieval performance in terms of bpref and infNDCG for TREC 2011 and 2012, respectively.

4. EXPERIMENTAL RESULTS

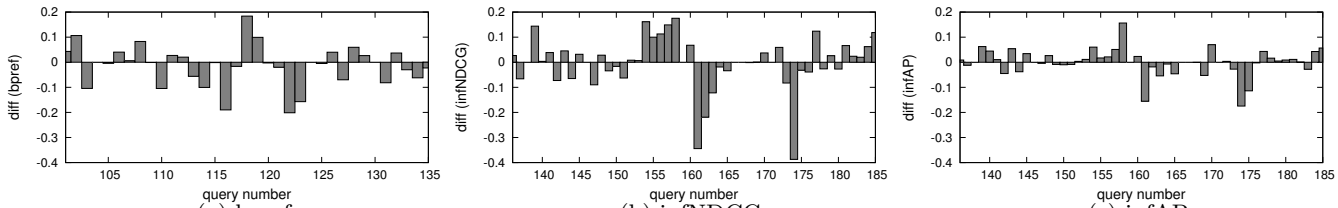
We compare the retrieval performance of our selective ranking approach with existing strong baselines using the TREC 2011 and 2012 Medical Records track test collections. Indeed, our baselines include the patient model, the document model, and the combination of the relevance scores computed from both the patient and document models using either CombSUM or CombMAX, as suggested in [21].

Table 2 compares the retrieval performance of our proposed approach with the aforementioned baselines, in terms of bpref, infNDCG, and infAP. In addition, to gauge the potential effectiveness of our deployed classifier, we also report the best possible retrieval performances that could be attained by our approach (i.e. an oracle), when the classifier correctly identifies an effective patient ranking approach for all of the queries. Firstly, we observe that the patient model and the document model attain a comparable retrieval effectiveness. Indeed, the document model outperforms the patient model in terms of the bpref and infNDCG measures, for TREC 2011 and 2012, respectively (bpref 0.5141 vs. 0.5006 and infNDCG 0.4481 vs. 0.4459), while the patient model performs better in terms of infAP for TREC 2012 (0.1865 vs. 0.1857). Moreover, we find that applying the data fusion techniques, including CombMAX and CombSUM, as suggested in [21], does not in general improve the retrieval performance over both the patient and the document models. On the other hand, we find that, with the 5-fold cross-validation setting, our selective approach outperforms all of the baseline approaches. Specifically, in terms of bpref, our approach (bpref 0.5261) significantly (paired-t test, $p < 0.05$) outperforms the patient model, the CombSUM, and the CombMAX baselines for up to 5.9%. However, in terms of infNDCG and infAP, the increased performances are not statistically different. The observed results demonstrate the effectiveness of our deployed approach in selecting the right ranking approach for a particular query. Note that the CombSUM and CombMAX do not perform as effectively as in [21], partially because we use DPH to rank documents in *the patient model* and to rank medical records at *the first phase of the document model*, instead of a language model. In addition, when aggregating the relevance scores of the medical records in the document model, we use the expCombSUM voting technique, instead of just summing up the relevance scores as in [21]. This suggests that the performances of CombSUM and CombMAX depend on the un-

[4] `http://terrier.org`

Table 2: The retrieval performances of different patient ranking approaches on the TREC Medical Records track collections. Statistical significance (paired t-test) at $p < 0.05$, at $p < 0.01$, and at $p < 0.001$ over an alternative approach are denoted ^{x}, ^{xx} and ^{xxx}, respectively. ^{x} is $^{\oplus}$, $^{\ominus}$, $^{\otimes}$, $^{\oslash}$ or $^{\odot}$ and refers to the patient model, the document model, CombMAX, CombSUM, and our selective approach (5-fold), respectively.

Approaches	2011	2012	
	bpref	infNDCG	infAP
Patient model	0.5006	0.4459	0.1865
Document model	0.5141	0.4481	0.1857
CombMAX	0.4968	0.4459	0.1865
CombSUM	0.4978	0.4453	0.1866
Our selective ranking approach (5-fold)	**0.5261**$^{\oplus,\otimes,\oslash}$	**0.4500**	**0.1906**
Our selective ranking approach (oracle)	0.5368$^{\oplus\oplus\oplus,\ominus\ominus,\otimes\otimes\otimes,\oslash\oslash\oslash,\odot}$	0.4830$^{\oplus\oplus,\ominus\ominus\ominus,\otimes\otimes,\oslash\oslash,\odot\odot\odot}$	0.2037$^{\oplus\oplus,\ominus\ominus\ominus,\otimes\otimes,\oslash\oslash,\odot\odot}$



(a) bpref (b) infNDCG (c) infAP

Figure 1: The difference between the retrieval performance obtained using the patient model and the document model on each query, in terms of bpref, infNDCG and infAP, respectively.

derlying used retrieval models. In contrast, our proposed approach overcomes this problem by appropriately learning from the features when selecting a patient ranking model. Furthermore, we find that if our classifier could make the correct decisions for each of the queries, the retrieval performances can further improve (See the oracle row in Table 2).

In addition, we compare the difference between the retrieval performance obtained using the patient model and the document model on every query, in Figure 1. Note that the difference in the retrieval performance is positive when the patient model is more effective (see Section 2.3). From this figure, we find that neither the patient model nor the document model is consistently more effective for all of the queries. We also find that the most effective patient ranking approach depends on the used measure (e.g. for query# 183, the patient model is more effective on infNDCG, while for the same query, the document model is more effective for infAP). This confirms the importance of deploying selective ranking approaches when ranking patients.

5. CONCLUSIONS

In this paper, we have argued for the importance to deploy a selective approach that appropriately selects the most appropriate patient ranking approach on a per-query basis. Our proposed selective approach exploits several query performance predictors as learning features within a regression-trees classifier, in order to suitably choose between the patient or the document models when ranking patients on a per-query basis. The results demonstrate the effectiveness of our proposed approach in comparison to recent strong baselines from the literature. Our proposed approach can be easily extended to include new learning features, as well as further possible patient ranking approaches.

6. REFERENCES

[1] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *TREC* 2007.

[2] A. R. Aronson and F. Lang. An overview of MetaMap: historical perspective and recent advances. *J. Am. Med. Inform. Assoc.*, 17(3), 2010.

[3] D. Carmel and E. Yom-Tov. Estimating the Query Difficulty for Information Retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1), 2010.

[4] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting Query Performance. In *SIGIR* 2002.

[5] D. Demner-Fushman, S. Abhyankar, A. Jimeno-Yepes, R. Loane, B. Rance, F. Lang, N. Ide, E. Apostolova, and A.R. Aronson, A Knowledge-Based Approach to Medical Records Retrieval. In *TREC* 2011.

[6] Y. Ganjisaffar, R. Caruana, and C. V. Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *SIGIR* 2011.

[7] B. He and I. Ounis. Query performance prediction. *Inf. Syst.*, 31(7), 2006.

[8] W. Hersh. *Information retrieval: A health and biomedical perspective (3rd ed.)*. New York : Springer, 2009.

[9] B. King, L. Wang, I. Provalov, and J. Zhou. Cengage Learning at TREC 2011 Medical Track. In *TREC* 2011.

[10] N. Limsopatham, C. Macdonald, R. McCreadie, and I. Ounis. Exploiting Term Dependence while Handling Negation in Medical Search. In *SIGIR* 2012.

[11] N. Limsopatham, C. Macdonald, and I. Ounis. Learning to Combine Representations for Medical Records Search. In *SIGIR* 2013.

[12] N. Limsopatham, C. Macdonald, and I. Ounis. A Task-Specific Query and Document Representation for Medical Records Search. In *ECIR* 2013.

[13] N. Limsopatham, C. Macdonald, I. Ounis, G. McDonald, and M. Bouamrane. University of Glasgow at Medical Records track 2011: Experiments with Terrier. In *TREC* 2011.

[14] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM* 2006.

[15] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *OSIR at SIGIR* 2006.

[16] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *TREC* 1994.

[17] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin. Parallel boosted regression trees for web search ranking. In *WWW* 2011.

[18] E. Voorhees and W. Hersh. Overview of the TREC 2012 Medical Records Track. In *TREC* 2012.

[19] E. Voorhees and R. Tong. Overview of the TREC 2011 Medical Records Track. In *TREC* 2011.

[20] Y. Zhao, F. Scholer, and Y. Tsegay. Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *ECIR* 2008.

[21] D. Zhu and B. Carterette. Combining Multi-level Evidence for Medical Record Retrieval. In *SHB* 2012.