Retrieving Medical Records with “sennamed”: NEC Labs America at TREC 2012 Medical Records Track

Yanjun Qi and Pierre-François Laquerre
Department of Machine Learning, NEC Laboratories America
Princeton, New Jersey, USA
yanjun@nec-labs.com, pierre.francois@nec-labs.com

Abstract

In this notebook, we describe the automatic retrieval runs from NEC Laboratories America (NECLA) for the Text REtrieval Conference (TREC) 2012 Medical Records track. Our approach is based on a combination of UMLS medical concept detection and a set of simple retrieval models. Our best run, sennamed2, has achieved the best inferred average precision (infAP) score on 5 of the 47 test topics, and obtained a higher score than the median of all submission runs on 27 other topics. Overall, sennamed2 ranks at the second place amongst all the 82 automatic runs submitted for this track, and obtains the third place amongst both automatic and manual submissions.

1 Introduction

The majority of medical information today is stored as an abundant combination of free, structured and semi-structured text. Electronic medical records (EMRs) document clinical information about a patient such as his/her medical history, current medical care, and current illnesses. This information can be leveraged by healthcare professionals to track the progress of patients, guide the diagnosis, and provide more personalized care to the patients. The urgent need for efficient processing and intelligent access of EMRs has led to a rapid increase in research efforts recently. As a notable example, the renowned TREC been holding a Medical Records track [1] since 2011, which has attracted many research groups from all over the world to participate and to evaluate the performance of their EMR retrieval algorithms.

The TREC Medical Records track includes a retrieval task aiming to find EMRs that are relevant to a given natural language query[1]. These EMRs are de-identified medical records, provided by the University of Pittsburgh BLULab NLP Repository [1]. There is a total of more than one hundred thousand medical reports from encounters with patients in various departments from multiple hospitals. This corpus contains nine types of reports, including radiology, emergency department, and radiology reports. These reports can be grouped into ∼17,000 distinct visits, each corresponding to a single patient's stay at the hospital. For the 2011 track, the participants were required to submit relevant records from the above EMR corpus for 35 topic queries (with one of the queries having no reports found in the end). For the 2012 track, submissions were evaluated by judging the relevance of their returned results on 50 given queries, of which 3 were later excluded by the organizers due to the lack of relevant visits for proper evaluation. Submissions were split in two different groups. Automatic submissions include those that do not require any human intervention, while manual submissions include everything else. Topics and relevance judgments were created by physicians who are also students in the bioinformatics program at Oregon Health and Science University.

The NECLA team submitted four automatic runs to the 2012 track. The main techniques used in our runs include medical concept detection, a vector-space retrieval model, a probabilistic retrieval model, a supervised preference ranking model, unsupervised dimensionality reduction, and query expansion. The details of these techniques are given in the next section. Experimental results for each model are presented in Section 3 and are further analyzed in Section 4.

2 Approach

The basic task of the TREC Medical Records track is to return a ranked list of visits that are relevant to a given ad-hoc query such as "Patients taking atypical antipsychotics without a diagnosis schizophrenia or bipolar depression". We explored a number of classical Information Retrieval (IR) technologies for this task and also considered the special properties of medical record text, such as frequent usage of acronyms. We used relevance judgments from the 2011 track for parameter tuning and model selection.

2.1 Preprocessing

We generated simple regular expressions to remove boilerplate text such as "My signature below is attestation that I have interpreted this/these examination(s) and agree with the findings as noted above.". To find such sentences, we searched for the most common substrings of several given lengths in the corpus.

The de-identification tags were converted to simple text to prevent downstream tools from interpreting the special syntax as punctuation. For example, "**DATE[Feb 01 06]" was converted to "Feb 01 06".

[1] `http://www.dbmi.pitt.edu/nlpfront`

The patient denies any abdominal pain.
C0030705 C0332319 negC0000737

Table 1: Semantic concept extraction on raw text tokens.

In the provided EMR collection, reports associated with the same patient stay are grouped into visits. The content-based retrieval task expects to retrieve those visits that are semantically relevant to a given query. We have tested two types of indexing in our runs: visit-based and report-based. In visit-based indexing, a visit's reports are concatenated into a single document. In report-based indexing, individual reports are indexed, and the query results are transformed into unique visits before being returned. There was no significant difference between those two approaches on the 2011 topics. Therefore, we opted to use the visit-based approach for all submissions. Thus, in the rest of this report, we use "document" to refer to all medical reports related to a given visit.

2.2 Term Representation Using Plain Text and/or UMLS Medical Concepts Transformation

Besides working on plain text tokens, we also utilized MetaMap[2] to convert the raw text into sequences of UMLS medical concepts. The UMLS metathesaurus [3] is the largest thesaurus in the biomedical domain, and tries to represent biomedical knowledge using semantic concepts and the relationships between them. MetaMap, a program developed by the National Library of Medicine (NLM), maps raw text tokens to corresponding Concept Unique Identifiers (CUIs), where each CUI belongs to a specific biomedical concept in the UMLS metathesaurus. Only top candidate CUIs were kept, and no limitation was put on the UMLS source. Negation detection was used to distinguish between concepts and their negated counterpart. Negated concepts were given unique ids so that downstream systems could tell them apart from the non-negated counterparts. The extraction on the full set of medical records led to a dictionary size of 62553, among which 7388 were negations. Table 1 provides a schematic example of the above procedure. In this table, C0030705 corresponds to "patient", C0332319 to "denies", and negC0000737 to the negation of "abdominal pain". The same process was applied to the query topics. Admission and discharge ICD codes were also converted to their UMLS equivalent and added to each visit. Other metadata from the XML was discarded. The end result is a representation of documents or topics containing a sequence of UMLS concept ids or their negation. In the following, we use "UMLS" to tag those retrieval runs using CUIs extracted from records and CUIs from queries as the basic term tokens. We use "raw text" to tag those retrieval runs using the plain text token (after preprocessing). We also test the combined representation of "UMLS + raw text" in our experiments, which uses the concatenation of plain text tokens and the extracted CUIs to represent records and query topics. See Table 2 for the different representations tried in our experiments.

2.3 Indexing and Ranking

Generally speaking, the task at hand is a standard ad-hoc IR task, where documents that are topically relevant to a query must be returned. Thus, we explore (1) a classic vector space retrieval model, (2) a language model based retrieval approach, and (3) a supervised preference ranking model belonging to the "learning to rank" category. We also test several other classic IR techniques in our runs, including dimensionality reduction using Latent Semantic Indexing (LSI), and query expansion.

2.3.1 Retrieval with a Vector-Space Model

In the vector space model, each document and query is represented as a vector of terms. In our experiments, the terms could be plain text tokens, detected CUIs, or both. Documents are then ranked by the similarity between the query vector and the document vector. Empirical studies of retrieval methods have found that good retrieval performance is closely related to the use of proper heuristics such as TF-IDF weighting [4]. We use one of the best performing vector space retrieval formula, BM25 [5]:

$$\sum_{\omega \in q \cap d} \ln(\frac{N - \mathrm{df}(\omega) + 0.5}{\mathrm{df}(\omega) + 0.5}) \cdot \frac{\mathrm{tf}(\omega, d) \cdot (k_1 + 1)}{\mathrm{tf}(\omega, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{1}$$

Here $\mathrm{tf}(\omega, d)$ represents the count of word ω in the document d, $\mathrm{tf}(\omega, q)$ is the count of word ω in the query q, and N is the total number of documents in the collection. $\mathrm{df}(\omega)$ is the number of documents which contain this term. $|d|$ represents the length of the document. $avdl$ is the average document length. k_1 and b are parameters that can be tuned.

2.3.2 Retrieval with Language Model with Dirichlet Smoothing

Besides the vector space retrieval model, language model based retrieval has attracted a lot of attention recently [6, 7]. Thus we test one retrieval model belonging to this category. This type of model builds a probabilistic language model G_d for each document d, and then ranks documents for a given query based on the likelihood that each document's language model could have generated the query: $P(q|G_d)$. The retrieval function is:

$$logP(q|G_d) = \sum_{\omega \in q \cap d} log \frac{p_s(\omega|d)}{\alpha_d * p(\omega|C)} + |q| * log(\alpha_d) + \sum_{\omega \in q} log(p(\omega|C)) \tag{2}$$

Here $|q|$ is the length of query, and $p(\omega|C)$ is the probability of the term given by the collection language model, which represents how popular the term is in the whole collection, i.e. playing a similar role to the well known IDF.

Language modeling based IR approaches typically employ a smoothing strategy to assign a non-zero probability to unseen terms, which can improve the accuracy of term probability estimation in general [6]. One of the best performing method is Dirichlet prior smoothing. When utilizing Dirichlet prior smoothing [6] to smooth the document language model, we have,

$$p_s(\omega|d) = \frac{\mathrm{tf}(\omega, d) + \mu * p(\omega|C)}{|d| + \mu} \tag{3}$$

$$\alpha_d = \frac{\mu}{|d| + \mu} \tag{4}$$

where $|d|$ is the length of the document, and μ is a parameter.

2.3.3 Retrieval with a supervised "Learning to Rank" Model

In addition, we study a retrieval model which is trained by supervised signals to rank a set of documents for given queries in the pairwise preference learning framework. This model belongs to the "learning to rank" category [8] which learns the preference or relevance function by assigning a real valued score to a feature vector describing a (query, object) pair. Specifically we utilize the so-called "supervised semantic indexing" (SSI) approach [9]. Given a query q and a document d, the relevance score between q and d is modeled as:

$$f(q, d) = q^\top W d = \sum_{i,j} W_{ij} \Phi(q_i, d_j), \tag{5}$$

where $\Phi(q_i, d_j) = q_i d_j^\top$ and W_{ij} models the relationship/correlation between i^{th} query feature q_i and j^{th} document feature d_j. This is essentially a linear model with pairwise features $\Phi(\cdot, \cdot)$ and the parameter matrix W is learned from labeled data. Pairwise features describing relationships between two raw features (e.g. word synonymy or polysemy) have been shown to improve the retrieval precision before [9]. The training labels are based on the 2011 TREC Medical Records track test collection which contains 7100 visits judged not relevant and 1765 judged relevant across 34 query topics. We perform two-fold cross-validation on this reference set for parameter tuning, i.e. half as training and half as testing. Our experimental results showed that SSI does not improve the retrieval results over simple retrieval models. This is in part due to the low quantity of queries and corresponding relevance judgments available for training.

2.3.4 Dimensionality Reduction using LSI

LSI [10] has been widely used for dimensionality reduction in IR. It is treated as one of the most successful tools for learning latent topics from text. Thus we also test this technique in our runs. We used Gensim[11] to train and obtain a model to project the document and query into a reduced space with m latent dimensions. Here m is a hyper-parameter to tune. Before applying LSI, the dictionary size was cut down to 44113 by filtering out tokens that appeared in too many visits ($> 99\%$).

2.3.5 Query Expansion with Pseudo-Relevance Feedback

We also test the classic pseudo-relevance feedback strategy, which has been found to improve performance of multiple TREC ad-hoc tasks before [12]. For a given query, pseudo-relevance feedback uses the designated retrieval model to retrieve the set of top-k ranked documents. It then expands the original query using the top ranked m candidate terms from this set of documents according to:

$$q_1 = \alpha \cdot q_0 + (1 - \alpha) \cdot \sum_{i=1..m} q^i_{rf} \tag{6}$$

Here, q_1 represents the revised query and q_0 is the original query. q^i_{rf} refers to the i-th candidate term from pseudo-relevance feedback. α, m and k are hyper-parameters to tune. This pipeline is based on Lavrenko's relevance models [13] implemented in Indri [7].

2.3.6 Query and Document Expansion with UMLS

We also experimented with several approaches to query and document expansion using UMLS. UMLS provides a hierarchy between concepts through several relations including *narrower than*, *synonymous to*, and others. For query expansion, every concept was expanded by including concepts synonymous to or beneath them in the UMLS hierarchy. Negations were also propagated. For documents, the expansion was done upwards. On the 2011 test topics, we found out that this expansion strategy was detrimental to retrieval performance, regardless of the combination used (query only, document only, both). We thus excluded this strategy from the submitted runs. More intelligently targeted expansion, such as expansion limited to specific concept categories, would likely have been more successful.

3 Results

Submissions to the TREC 2012 Medical Records track were evaluated by judging the relevance of their submitted results on 47 given queries. The main evaluation metric used is infAP. The inferred normalized discounted cumulative gain (infNDCG), R-precision and the precision at 10 (P@10) were also reported. Before the final submission, we used the 34 test queries and their associated relevance judgments from the 2011 track to perform hyper-parameter tuning, model selection and the evaluation of various possible configurations. Table 2 provides the list of our retrieval variants.

Table 3 summarizes the retrieval performance of various configurations from Table 2 on the TREC 2011 medical test topics. For each retrieval configuration, we tuned the hyper-parameters to optimize the sum of the averaged bpref and R-prec metrics [1]. The value range tried for the hyper-parameters of the vector space retrieval (i.e. k_1 and b) and language model retrieval (i.e. μ) models are based on the suggestions by [4]. We can see that, in general, the UMLS concept based representation gives better retrieval performance, when compared

Submitted runs	Term Representation	Indexing & Ranking
sennamed1	UMLS concept	language model retrieval, query expansion
sennamed2	UMLS concept	vector space retrieval, query expansion
sennamed3	UMLS concept	vector space retrieval
sennamedlsi	UMLS concept	vector space retrieval, LSI
Other runs		
sennamed-4	UMLS concept	language model retrieval
sennamed-5	UMLS concept + raw text	language model retrieval, query expansion
sennamed-6	UMLS concept + raw text	vector space retrieval, query expansion
sennamed-7	UMLS concept + raw text	vector space retrieval
sennamed-8	UMLS concept + raw text	language model retrieval
sennamed-9	raw text	vector space retrieval
sennamed-A	UMLS concept	"learning to rank" retrieval

Table 2: Various retrieval configurations we tried.

with "raw text" or "raw text + UMLS". Finally we selected four different runs (sennamed1, sennamed2, sennamed3 and sennamedlsi) which reflect the various techniques we tried. We use the best selected parameters of these models (based on 2011 track) to rank EMRs for 47 queries requested for the 2012 track.

Table 4 provides an overview of the performance of our four submitted runs based on the relevance judgments for 47 test topics in 2012 medical track. We can see that the performance difference between these four runs on 2012 test queries are quite consistent with their relative differences on the 2011 test collection. Table 6 shows the number of topics in which our best run (sennamed2) was the best, above median, on par with the median, lower than the median, or the worst among all submitted runs, across the four main performance metrics. Finally, tables 7 and 8 compare our best run in terms of the infAP and P@10 for each topic versus the best, median and worst runs among all 2012 submissions. Table 5 lists the best five run among all submissions for 2012 TREC medical track. We can see that overall, sennamed2 ranks second amongst all automatic submissions, and third amongst all runs [14].

4 Discussion

Overall, our submission sennamed2 obtained the best infAP score on 5 of the 47 test topics, and did better than the median on 27 others. This is rather surprising given the simplicity of the approach. To better understand the performance, we present in Table 6 the comparison of sennamed2 based on the number of topics in a given performance metric. In addition, Table 7 and Table 8 present the

Run / Metric	bpref	R-prec	P@10	infAP
sennamed1	0.5012	0.3755	0.5176	0.3322
sennamed2	0.5761	0.4196	0.5129	0.3912
sennamed3	0.5033	0.3839	0.4735	0.3314
sennamedlsi	0.5308	0.3327	0.4118	0.3108
sennamed-4	0.4619	0.3448	0.4706	0.2987
sennamed-5	0.4474	0.321	0.4794	0.2964
sennamed-6	0.5362	0.4026	0.5088	0.3954
sennamed-7	0.4886	0.384	0.4824	0.3697
sennamed-8	0.4444	0.3181	0.4706	0.2966
sennamed-9	0.4388	0.3384	0.4735	0.3157
sennamed-A	0.4782	0.3156	0.3912	0.2669

Table 3: Performance of our retrieval runs on the 2011 test topics. The term representation and methods of ranking/indexing are listed in Table 2.

Metric / Run Name	sennamed1	sennamed2	sennamed3	sennamedlsi	median
infAP	0.2246	0.2745	0.2169	0.2151	0.1695
infNDCG	0.4780	0.5468	0.4688	0.4468	0.4243
R-prec	0.3457	0.3805	0.3298	0.2974	0.2935
P@10	0.5255	0.5574	0.5447	0.4468	0.4702

Table 4: Performance metrics for four submitted runs, compared with the median over all teams on the 2012 test topics.

performance of sennamed2.

The majority of the errors were due to a lack of higher level query understanding. Our system could not properly interpret constraints such as "[...] developed disseminated intravascular coagulation **in the hospital**". Along similar lines, temporal aspects were also ignored, such as the one in topic 177: "Patients treated for depression **after** myocardial infarction".

While negation detection was useful, a more sophisticated approach that also takes uncertainty into account would have fared better. As is, our system cannot make the difference between "The patient was tested for disseminated intravascular coagulation" and an actual diagnosis of disseminated intravascular coagulation. Furthermore, the scope of negation detection was limited to a single sentence, whereas negations sometimes occur past sentence boundaries.

Finally, errors in MetaMap's concept detection also accounted for some of our errors. Despite its overall reliability, certain topics proved problematic. For instance, in topic 137, "TNF-inhibitor treatments" was converted to two concepts — "inhibitor" and "treatments" — discarding the "TNF" part. Another example is topic 179, where "atypical antipsychotics without a diagnosis schizophrenia" became "atypical schizophrenia (negated)" and "antipsychotics". In the end, it may be better to combine UMLS concepts with the original text,

Run Name / Metric	infNDCG	infAP	P@10
NLMManual*	0.680	0.366	0.749
udelSUM	0.578	0.286	0.592
sennamed2	0.547	0.275	0.557
ohsuManBool*	0.526	0.250	0.611
atigeo1	0.524	0.224	0.519

Table 5: Performance metrics for our best run sennamed2, compared with the best four other runs among all teams on the 2012 test topics [14]. Manual runs are marked with a star(*).

Metric / Number of topics	Worst	< median	= median	> median	best
infAP	0	13	2	27	5
infNDCG	0	13	1	27	6
R-prec	2	10	6	24	5
P@10	4	7	12	13	11

Table 6: Comparison of sennamed2 based on the number of topics in a given performance metric.

albeit with a more elaborate approach than simple concatenation.

5 Conclusion

The NECLA team submitted four runs to the Medical Records track at TREC 2012. We experimented with a set of techniques including dimensionality reduction, medical concept detection, query expansion and various document retrieval approaches for this task. Among our four submitted runs, the best results were achieved using a combination of medical concept detection, vector-space retrieval model and query expansion using pseudo-relevance feedback. This simple pipeline obtained a final infAP score of 0.2745, compared to the median infAP score 0.1695 of all automatic submissions. Our best run, sennamed2 ranks as the third over all 2012 TREC Medical track submissions, and second if we only take automatic runs into account.

References

[1] E. Voorhees and R. Tong. Overview of the TREC 2011 medical records track. In *The Twentieth Text REtrieval Conference Proceedings TREC*, 2011.

[2] Alan R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: The metamap program, 2001.

topic	best	median	worst	sennamed2
136	0.5724	0.0492	0	**0.0494**
137	0.0155	0	0	**0.0113**
139	0.6906	0.2046	0	**0.6634**
140	0.5122	0.2554	0	0.2387
141	0.4549	0.1307	0	**0.22**
142	0.3092	0.1492	0	**0.1856**
143	0.6229	0.458	0.0064	**0.5253**
144	0.1723	0.0804	0	**0.1194**
145	0.6206	0.4394	0	**0.5948**
146	0.4338	0.0132	0	**0.4338**
147	0.2012	0.0737	0.0027	**0.1873**
148	0.5584	0.3994	0	**0.4839**
149	0.092	0.0291	0.0004	**0.0742**
150	0.8196	0.5237	0	0.516
151	0.0552	0.0058	0	0.0026
152	0.1299	0.0475	0	0.028
153	0.5716	0.2226	0	**0.3607**
154	0.4681	0.0601	0.0002	0.0052
155	0.2033	0.0664	0	**0.2033**
156	0.1115	0.0548	0.0031	0.0469
157	0.4214	0.0493	0	**0.1897**
158	0.7885	0.2801	0	**0.7237**
160	0.2486	0.0624	0.0002	**0.2486**
161	0.8444	0.1152	0	**0.754**
162	0.0725	0.0463	0.0031	**0.047**
163	0.2402	0.098	0	**0.2402**
164	0.7426	0.4514	0	**0.6468**
165	0.4974	0.2518	0	**0.3978**
167	0.4324	0	0	0
168	0.1458	0.0355	0.0007	0.0274
169	0.5277	0.4435	0.0363	**0.5069**
170	0.8474	0.5386	0	**0.6471**
171	0.6934	0.2177	0	**0.6202**
172	0.2474	0.0699	0.0006	0.0606
173	0.358	0.0423	0	0.0099
174	0.2768	0.094	0	0.0555
175	0.7323	0.3622	0	**0.4571**
176	0.1819	0.0403	0	**0.0697**
177	0.4027	0.0385	0.0037	0.0349
178	0.9055	0.6168	0	**0.7892**
179	0.0674	0.001	0	**0.028**
180	0.5294	0.2877	0	**0.3262**
181	0.4044	0.0252	0	**0.4044**
182	0.1062	0.0761	0.0036	0.0657
183	0.3542	0.0903	0	0.0689
184	0.5762	0.2946	0.0041	0.2946
185	0.6571	0.0754	0	**0.2375**
Mean	0.4238	0.1695	0.0014	**0.2745**

Table 7: Comparison of sennamed2 to best/median/worse of all teams on the 2012 test topics, in term of infAP for every topic. Number in bold when above the median.

topic	best	median	worst	sennamed2
136	0.9	0.1	0	0.1
137	0.1	0	0	0
139	0.9	0.6	0	**0.7**
140	0.8	0.5	0	0.4
141	0.9	0.3	0	0.3
142	0.9	0.6	0	**0.8**
143	1	1	0	**1**
144	1	0.5	0	0.4
145	0.8	0.6	0	**0.7**
146	0.7	0.1	0	**0.6**
147	1	0.7	0	**1**
148	1	0.9	0	**1**
149	0.6	0.2	0	**0.4**
150	0.7	0.5	0	0.4
151	0.7	0.1	0	0
152	0.4	0.1	0	0
153	1	0.9	0	0.8
154	0.9	0.6	0	0.1
155	0.8	0.5	0	**0.8**
156	1	0.6	0.2	**0.7**
157	0.9	0.4	0	**0.8**
158	1	0.6	0	**1**
160	1	0.7	0	**1**
161	1	0.3	0	**1**
162	1	0.8	0.2	0.7
163	0.9	0.6	0	**0.8**
164	1	0.7	0	**0.9**
165	1	0.7	0	**0.9**
167	0.3	0	0	0
168	0.8	0.3	0	0.3
169	1	1	0.4	**1**
170	1	0.9	0	**1**
171	0.7	0.2	0	**0.6**
172	0.8	0.4	0	0.4
173	0.9	0.5	0	0.1
174	0.4	0.1	0	0
175	0.7	0.4	0	0.4
176	0.7	0.2	0	0
177	0.6	0.3	0	0.3
178	1	0.9	0	**1**
179	0.6	0	0	**0.4**
180	1	0.7	0	0.7
181	0.7	0.1	0	**0.7**
182	1	0.9	0.2	0.9
183	0.8	0.2	0	0.2
184	1	0.7	0	0.7
185	0.4	0.1	0	**0.2**
Mean	0.8149	0.4702	0.0213	**0.5574**

Table 8: Comparison of sennamed2 to best/median/worse of all teams on the 2012 test topics, in term of P@10 for every topic. Number in bold when above the median.

[3] A.T. McCray, S.J. Nelson, et al. The representation of meaning in the umls. *Methods of information in medicine*, 34(1-2):193, 1995.

[4] H. Fang, T. Tao, and C.X. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM, 2004.

[5] S. Robertson and H. Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 3. Now Publishers Inc, 2009.

[6] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM, 2001.

[7] Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. Indri: a language-model based search engine for complex queries. Technical report, in Proceedings of the International Conference on Intelligent Analysis, 2005.

[8] T.Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[9] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Supervised semantic indexing. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 187–196. ACM, 2009.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 1990.

[11] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[12] G. Cao, J.Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250. ACM, 2008.

[13] V. Lavrenko and W.B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[14] E. Voorhees and W. Hersh. Overview of the TREC 2012 medical records track. In *The Twenty First Text REtrieval Conference Proceedings TREC*, 2012.