Learning to rank clinical trials with rule-based criteria

Gonçalo Araújo, André Mourão, João Magalhães
NOVA Laboratory for Computer Science and Informatics
Faculdade de Ciências e Tecnologia
Universidade NOVA de Lisboa
2825-516 Caparica
{gc.araujo, a.mourao}@campus.fct.unl.pt, jmag@fct.unl.pt

ABSTRACT

This report describes the NOVASearch retrieval system for the TREC 2018 Precision Medicine Track in the Clinical Trials matching task. The parsing of queries and documents in the Clinical Trials task were structured into multiple fields according to the details about inclusion and exclusion criteria. We also considered multiple text processing filters on the largest text fields.

Keywords

Medical Text Retrieval; Term expansion; Clinical trial retrieval

1. INTRODUCTION

The TREC Precision Medicine Track 2018, aims to provide clinicians with important information to support medical decisions. The clinical decision support is focused on a specific use case, cancer patients, so that clinicians can have access to very specific medical trials. In this paper, we report the NOVASearch team participation in the Clinical Trials retrieval task. The goal is to retrieve clinical trials were the patient could be a participant due to his disease. This is a natural step towards more comprehensive clinical decision support tasks as was explored in past TREC editions[3, 4].

Section 2 details the indexing and retrieval methods implemented. Section 4 discusses the evaluation results.

2. METHODS AND ALGORITHMS

Both indexing and retrieval methods were implemented with Apache Lucene, a text search engine library for Java, that contains very helpful methods for creation of indexes, Queries and text search.

2.1 Documents shallow parser

The documents parser uses a pipeline of filters so that we can mitigate factors like, the number of irrelevant words and variation of words (ex.reduction to primal verbal form *driving* to *drive*), which leads to more effective retrieval results. Most of the methods are widely used in Information Retrieval, that were tuned to the tasks at hand:

- **Tokenization:** Used to remove all form of punctuation and split text into tokens. We also converted all words into lower case.
- **Stop word removal:** Remove specific words like ”this”, ”a”, ”or”, that will occur in most of the English texts.
- **Word grams:** This filter creates tokens from other tokens. We tested a range of minimum and maximum size of neighboring words to create the indexing tokens.
- **Stemming:** We used the Snowball filter to stem the indexed documents. Stemming is the process of reducing words to their word stem.
- **Character grams:** Creates n-grams of words, with a minimum and a maximum length for the words is given as a parameter. This is a technique that has been been successful in the medical domain due to the complex spelling of medical terms (many common prefixes and suffixes).
- **Demographics filter** Removes the demographic information from the trials, *table 1* is a good example. Creates age range and a gender exclusion criteria. This type of processing can be found in a similar way in the PICO(population, Intervention, Control, Outcome) fields extraction [5].

2.2 Document information extraction

The TREC PM 2018 collection of clinical trials, contains a large number of trials and most of them not specific for our topics. To index the information contained in the documents, it was necessary to choose the fields that could be relevant in a medical environment and index them. After some inspection, we indexed the fields *gender*, *minimum age*, *maximum age*, *title*, and *condition(disease)*. The use of the patient specific information can be seen in other articles with proven results [2].

The provided format of the documents in the collection, a field-structured document, helped us to minimize the workload on the pre-processing of the text before indexing it. For example, **<minimum_age>** *18* Years **</minimum_age>**, after retrieving the text of the xml tag **<minimum_age>** , there's only need to split the text by white spaces and the result is an array containing [”18”,”years”], we know for sure that the first value on the array is a string containing a number, that represents the minimum age of the patients for this clinical trial.

The information contained in the larger text fields such as, *brief_title* or *summary* are first processed using the analysis process explained in the previous section, and then indexed. Not all the fields are available in all clinical trials, however, in our implementation we stored all the fields even if they're empty, or nonexistent on the clinical trial. The indexed fields are described in table 1.

Extracted and created fields	Description
Full text	Field containing a concatenation of relevant text fields about the intervention, (brief title, official Title, briefSum, description, det desp).
Brief title	A brief title containing only some keywords of the title.
Official title	The official title of the clinical trial.
Brief Summary	A excerpt of the summary.
Detailed description	A detailed description specifying the intervention made on the clinical trial.
CriteriaInc	Specific inclusion criteria for the patients (other deceases, allergies, other drugs previously used).
CriteriaExc	Specific exclusion criteria for the patients (other deceases, allergies, other drugs previously used).
Intervention	Describes the trial intervention type.
InterventionExp	Expansion with SNOMED and Mesh of the trial intervention type.
StudyType	Type of clinical trial
Purpose	The purpose of the clinical trial
Gender	The genders acceptable for the trial.
MaxAge	The maximum age of patients acceptable for this trial.
MinAge	The minimum age of patients acceptable for this trial.
Condition	Description of the patient clinical condition requirements.
ConditionExp	Expansion with SNOMED and Mesh of the condition.

Table 1: The fields created from the information extracted from the clinical trials.

2.3 Filters

Several filter were implemented to remove clinical trials that do not match some of the criteria. Filters use one of the indexed fields and are applied as described in the runs section.

2.4 Retrieval Models

To retrieve relevant documents for each topic (a clinical case), we first did some processing of the TREC topics text fields, corresponding to the same steps we did to the indexed fields. We examined multiple runs using different types of analyzers, so we could test which analyzers would do a better job in filtering MESH terms [1] and overall medical relevant terms. We also used several types of query parsers and various ranking functions. Our main focus was to create the maximum number of inclusion and exclusion criteria, querying the collection of documents with information relevant to the indexed fields, this way we could take advantage of having structured documents in our data set and in our topics.

2.4.1 *Vector Space Model*

Term Frequency-Inverse Document Frequency consists, in a statistical method to define the importance of words for a specific document and simultaneously for the collection. If a word frequency in a collection is low, and its frequency in a document is high its tf-idf value is higher than the value for words that occur more often in the collection, even if they have high frequency in a few documents.

2.4.2 *BM25 and variants*

We use both standard BM25 and variants of BM25 to acommodate documents with different ranks and to lower bound term frequencies.

2.4.3 *Language Models*

Language Models are also known for being competitive in a number of scenarios. We also used LM with Dirichlet smoothing and LM with Jelineck-Mercer Smoothing. This is an important model for cases where the query expasion increase the query length, where the Jelineck-Mercer smoothing is known to work better.

2.5 Learning to Rank Models

We created the features that are described in table 3 and the retrieval models of the previous section. Some of the filters are also described in table 3.

We used two tree based learning to rank methods, based on the assumption that those algorithms would capture the non-linear relations provided by the filters. In particular we used the AdaRank and the LambdaMART algorithms. Also, we used the Coordinate Ascent as a strong baseline.

3. EVALUATION

The Clinical Trials retrieval system, consisted in applying BM25 with multiple combinations of search results filters based on demographic, and exclusion criteria to reduce non relevant retrievals.

3.1 Runs description

The base Query is the disease, gene variant. patient demographics are used as filters to inclusion and exclusion criteria. PRF query expansion using top 3 documents retrieved and exclusion criteria based on other conditions the patient may suffer. In general, the runs for retrieving Clinical Trials are:

- **Run 1:** Search (BM25L similarity) in the trial title, summary and description text. Query is the disease, gene variant text and the terms ”solid tumor” and ”solid neoplasm”. Search (BM25L similarity) in the inclusion criteria, query is the gene variant text. Searching for trials with intervention, study type and primary purpose fitted for oncology treatments. Results filtering by the patient age and gender. Final ranking list obtained after re-ranking documents according official and meSH conditions. Matching documents condition with expanded query disease and gene variation.

- **Run 2:** Search (BM25L similarity) in the trial title, summary and description text. Query is the disease, gene variant text and the terms ”solid tumor” and ”solid neoplasm”. Search (BM25L similarity) in the inclusion criteria, query is the gene variant text. Searching for trials with intervention, study type and primary purpose fitted for oncology treatments. Results filtering by the patient age and gender.
- **Run 3:** LETOR using LambdaMart algorithm. Features based on multiple retrieval functions (BM25, TF-IDF, Language Models) and features based on to Run 1 and 2 filters.
- **Run 4:** LETOR using AdaRank algorithm. Features based on multiple retrieval functions (BM25, TF-IDF, Language Models) and features based on to Run 1 and 2 filters.
- **Run 5:** LETOR using Coordinate Ascend. Features based on multiple retrieval functions (BM25, TF-IDF, Language Models) and features based on to Run 1 and the methods described in the following section.

3.1.1 Run5: Deep IE + CoordAsc

On run 5, we added an additional set of features. We used the score values from run 1 as the main feature. In addition, we added all the features used to train LETOR models from run 3 and 4, we added a set of features that counted the number of matches according to a larger set of criteria.

We generated two versions of all fields in the query (Disease, Gene, Preconditions):

- A BROAD expansion query, with all expansion terms (for example, ”breast cancer” will be expanded using SNOMed and MeSH for all terms [”Breast Cancer”, ”Breast”, ”Cancer”], resulting in a query that adds expansions to all terms: [”Breast Cancer” -> [”Malignant Tumor Of Breast”, ”Ca - Breast Cancer”], [”Breast” -> [”Breasts”, ”Breast Structure”, ”Mamma”, ”Breast Anatomy, ”Mammary Gland, ”Entire Breast”], ”Cancer” -> [”Neoplasm”, ”Benign Neoplasm”, ”Cancers”, ”Neoplasms”, ”Benign”, ”Neoplasm”, ”Benign”, ”Tumor”, ”Neoplasms”, ”Tumors”, ”Ca - Cancer”, ”Tumor”, ”Malignant”, ”Malignant Neoplasm”, ”Cancer Morphology”, ”Malignancy”, ”Malignant Tumor”, ”Malignant Tumor Morphology”, ”Neoplasm”, ”Malignant”, ”Malignant Neoplastic Disease”, ”Blastoma”, ”Neoplasm”, ”Malignant (primary)”, ”Unclassified Tumor”, ”Malignant”, ”Malignant Neoplastic Disease (primary)”, ”Malignant Neoplasm”, ”Primary”]
- A NARROW expanded query, which will only expand the NARROWEST terms that are present in the collection: For example: in the previous example, ”Breast Cancer” contains three terms that are present in either MeSH and SNOMed: [”Breast Cancer”, ”Breast”, ”Cancer”,]. The narrow expansion will recognize that ”Breast” and ”Cancer” are BROADER than ”Breast Cancer” and thus, only expand the NARROWER term [”Breast Cancer”-> [”Malignant Tumor Of Breast”, ”Ca - Breast Cancer”]].

The goal is to avoid the query drift that is added by adding too many expansion terms to the query. We wanted to study whether using different versions of the query on the different terms makes a different in feature selection and relevance.

To study the impact of the NARROW and BROADER terms for re-ranking, the following features were extracted for BOTH version of the query:

- the number of query terms (condition, gene, and NEGATED exclusion criteria) present in the inclusion criteria, exclusion criteria, condition,
- number of condition terms present in the title
- number of gene terms present in the title
- number of condition and gene terms is present in the inclusion criteria
- number of condition and gene terms is present in the exclusion criteria

These features were used to train a LETOR model, using coordinate accent.

- **Retrieval Score da RUN 1:** float feature that represents the score returned by the set of techniques used on run 1 methods.
- **title matches gene:** binary feature that describes whether any query term (cancer type, gene name, preconditions (for 2017 data)) or any of the expansions appear in any of the inclusion criteria sentences feature that describes whether the gene name or any of its expansions appear in the title
- **exc matches gene:** binary feature that describes whether any query term (cancer type, gene name, preconditions (for 2017 data)) or any of the expansions appear in any of the exclusion criteria sentences
- **total pos i1:** count feature that describes whether gene name or any of its expansions appear in any of the inclusion criteria sentences non-negated
- **total neg i1:** count feature that describes whether the negated gene name or any of its expansions appear in any of the inclusion criteria sentences
- **total pos e1:** count feature that describes whether gene name or any of its expansions appear in any of the exclusion criteria sentences
- **total neg e1:** count feature that describes whether the negated gene name or any of its expansions appear in any of the exclusion criteria sentences

Run/Method	P@5	P@10	P@15	P@20
(1) Deep IE	0.568	0.516	0.468	0.43
(2) Deep IE + Mesh	0.608	0.544	0.4973	0.466
(3) LambdaMART	0.528	0.484	0.44	0.408
(4) AdaRank	0.496	0.416	0.3667	0.338
(5) CoordAsc	0.612	0.552	0.508	0.464

Table 2: Results for the Clinical Trials Task.

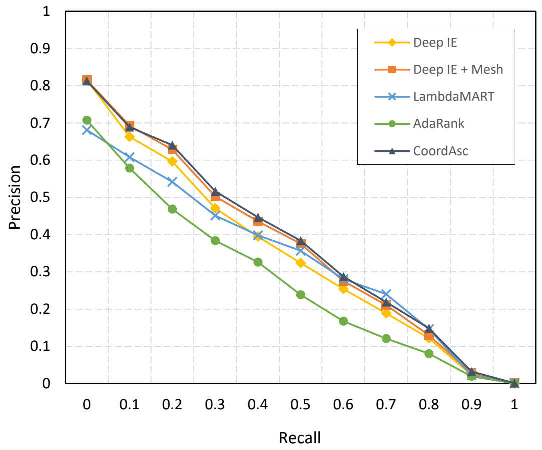

Figure 1: Precision-recall graph illustrates the different runs.

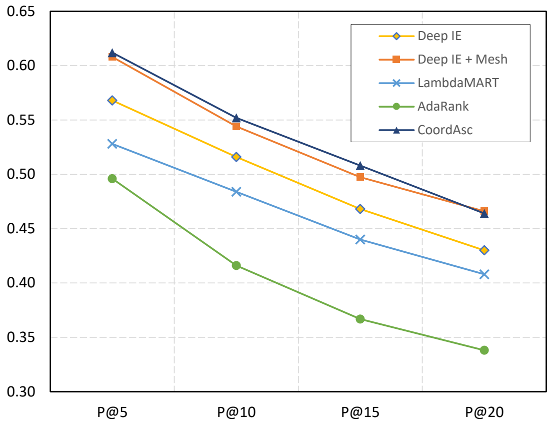

Figure 2: Precision at top retrieved results, i.e., P@5, P@10, P@15 and P@20.

3.2 Discussion

The overall results for the Clinical Trials retrieval were very positive: all runs except one, attained a performance above median. In particular, the two simplest runs, without learning to rank, were the most stable ones (runs 1 and 2). Surprisingly, the learning to rank methods AdaRank and LambdaMART were not as successful as one would expect. Our initial hypothesis, that tree based methos would benefit from filtering criteria, was not confirmed by results.

Nevertheless, the Coordinate Ascent run-5, was able to achieve the best run. It is also positive that such method was the best, because it is also the mehtod that offers interpretable results.

What was evident in all the runs described in this paper is that the information extracted from the documents played a critical role in the success of our system. Hence, as future work, we plan to the generalise feature extraction process to make the patient / clinical trial matching process more effective.

References

[1] J. Dutkiewicz, C. JÄŹdrzejek, M. FrÄĚckowiak, and P. Werda. Put contribution to trec cds 2016.

[2] A. E. W. Johnson, M. M. Ghassemi, S. Nemati, K. E. Niehaus, D. A. Clifton, and G. D. Clifford. Machine learning and decision support in critical care. *Proceedings of the IEEE*, 104(2):444–466, Feb 2016.

[3] A. Mourão, F. Martins, and J. Magalhães. Novasearch at trec 2015 clinical decision support track.

[4] A. Mourao, F. Martins, and J. Magalhaes. Novasearch at trec 2014 clinical decision support track. Technical report, UNIVERSIDADE NOVA DE LISBOA (PORTUGAL), 2014.

[5] H. Scells, G. Zuccon, B. Koopman, A. Deacon, L. Azzopardi, and Geva. Integrating the framing of clinical questions via pico into the retrieval of medical literature for systematic reviews.

Field	**Query**	**Matching or Similarity**
gene	expanded gene	BM25
gene	expanded gene	LMDIR
purpose	treatment, diagnosis,prevention	
intervention	drug, biological, radiation	
studytype	tnterventional	
text	gene/disease	tf
text	gene/disease	idf
text	gene/disease	tfidf
text	gene/disease	bm25
text	gene/disease	bm25l
text	gene/disease	bm25+
text	gene/disease	lmd
text	gene/disease	lmjm
text	text length	
text	gene	of genes keywords
text	gene	% of keywords matched
text	disease	of diseases keywords
text	disease	% of keywords matched
detailed description	gene/disease	tf
detailed description	gene/disease	idf
detailed description	gene/disease	tfidf
detailed description	gene/disease	bm25
detailed description	gene/disease	bm25l
detailed description	gene/disease	bm25+
detailed description	gene/disease	lmd
detailed description	gene/disease	lmjm
detailed description	text length	
detailed description	gene	of genes keywords
detailed description	gene	% of keywords matched
detailed description	disease	of diseases keywords
detailed description	disease	% of keywords matched
official title	gene/disease	tf
official title	gene/disease	idf
official title	gene/disease	tfidf
official title	gene/disease	bm25
official title	gene/disease	bm25l
official title	gene/disease	bm25+
official title	gene/disease	lmd
official title	gene/disease	lmjm
official title	text length	
official title	gene	of genes keywords
official title	gene	% of keywords matched
official title	disease	of diseases keywords
official title	disease	% of keywords matched
brief summary	gene/disease	tf
brief summary	gene/disease	idf
brief summary	gene/disease	tfidf
brief summary	gene/disease	bm25
brief summary	gene/disease	bm25l
brief summary	gene/disease	bm25+
brief summary	gene/disease	lmd
brief summary	gene/disease	lmjm
brief summary	text length	
brief summary	gene	of genes keywords
brief summary	gene	% of keywords matched
brief summary	disease	of diseases keywords
brief summary	disease	% of keywords matched

Table 3: Learning to rank and filtering fields.