UCD IIRG at TREC 2011 Medical Track

James Cogley, Nicola Stokes, John Dunnion and Joe Carthy
School Of Computer Science and Informatics
University College Dublin, Belfield, Dublin 4, Ireland.
james.cogley@ucdconnect.ie,
{nicola.stokes, john.dunnion, joe.carthy}@ucd.ie

Abstract

In this paper, we present several approaches to the retrieval of medical visits in response to user queries on patient demographics. A visit is comprised of one or more medical reports. Given a data collection of medical reports, TREC Medical Track participants had the opportunity to either preprocess the documents concatenating reports into visits, or to post-process by retrieving reports and developing a method to create a ranking of visits given the retrieved reports. This paper outlines attempts at both approaches in order to determine the influence of the disparity of document lengths in the collection. For both these approaches query expansion and concept re-ranking are applied. Concept reranking identifies the number of unique concepts from an expanded query contained in a document, and boosts the rank of documents which contain more unique concepts.

1 Introduction

The inclusion of the TRECMed medical track at TREC this year reflects the growth of interest in the processing and retrieval of medical texts. This year the medical track focused on the retrieval of patients who match certain criteria (demographic, current conditions, treatments undergone etc.) outlined in a natural language query.

For this task, the unit of retrieval was a 'visit', which is a collection of medical reports pertaining to a single episode. The original dataset used in this task, provided by BLULab's[1] NLP repository, contained single reports. TRECMed provided a file that mapped reports to visits.

There are two possible methods to retrieve a visit document. A visit document may be created by concatenating all reports for that visit into a super-document, which is then added to the index. However, there is a great disparity in the length of visit documents, with the average document length (in bytes) 15433.43 with a standard deviation of 23029.76. Given the influence the length may have on the ranking algorithm, a post-processing approach was also adopted. In this approach, reports are treated as the unit of retrieval with their scores combined post-retrieval in order to create a ranking of visits.

The authors' submitted four runs investigating the impact of these approaches, as well as the use of structured querying and language modeling techniques.

The paper is organised as follows: Section 2 describes the system's architecture, outlining each component along with a description of each of the four runs submitted; Section 3 presents the results with Section 4 providing a discussion on the results.

2 System Description

In this section we will discuss the general architecture of the system, outlining its core components in detail, as well as providing a description of the four runs submitted to this track.

This is the inaugural year of the Medical track at

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

TREC and focuses on retrieving patients who fulfill certain medical and demographic criteria. These criteria include conditions that the patient may have, a treatment that they are undergoing, a certain age or gender demographic, or a combination of all of these criteria. This track has direct applicability to the task of finding suitable participants for clinical trials, who must meet a strict set of 'inclusion criteria' much like the criteria described above.

The document collection for this task was obtained from BLULab's NLP repository. This collection comprises of 107,111 individual clinical reports, which have been de-identified and range from surgical pathology reports to discharge summaries. The unit of retrieval for this task was a 'visit', which may consist of one or more reports.

The Information Retrieval engine Indri was used for indexing, retrieving and processing of queries. Indri was developed as part of the Lemur Project[2] at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst and the Language Technologies Institute (LTI) at Carnegie Mellon University. It incorporates state-of-the-art retrieval methods, combining language modeling and inference network approaches (Metzler and Croft, 2004). It was chosen for its efficiency, usability and foremost its structured query language, which allows for more expressive queries indicating ordering, windowing, term-weighting and phrasal searching.

The system developed for this task builds on Indri in the following manner. First the dataset is indexed as described in Section 2.1. Systematic manual query expansion is then performed using resources such as PubMed and MeSH. These queries are then manually translated in Indri's query language and passed into Indri. Concept-based reranking (Stokes et al., 2007) of the documents is performed on the retrieved results. Finally, if the index is built from individual reports (as opposed to visits), post-processing is required to create a ranking of visits as opposed to ranked reports. This process is explained in Section 2.4. The section ends with a description of the four runs submitted to the track.

2.1 Indexing

For the purpose of investigating the effects of document length on the performance of the approaches, two indices were built.

Documents in the first index were made up of single reports, with a post-processing step determining the rank of a 'visit' given the ranking of its constituent reports. This step is outlined in Section 2.4. For the second index, visit documents are created prior to indexing by using a simple shell script to concatenate a visit's constituent reports. The script reads the mapping file provided by TREC, which mapped reports to visits using unique identifiers, thus concatenating the reports. Neither of the indices were stemmed, as medical texts are an unending source of acronyms and abbreviations. For example, using general English language stemming tools, a term such as 'AIDS' would incorrectly become 'AID', thus completely changing the meaning.

2.2 Query Expansion

As well as containing a vast lexicon of abbreviations and acronyms, clinical narratives have a high degree of synonymy, as healthcare professionals may use different terminology to describe one concept. To combat these problems, manual query expansion was employed. The process described below was performed by one of the authors. Firstly, the query is submitted to PubMed[3]. Pubmed creates chunks from this query which represent entries in MeSH [4]. A manual systematic lookup is then performed on these entries at `http://www.ncbi.nlm.nih.gov/mesh` i.e. all synonyms are then taken from these results and added to the original query. No manual filtering of appropriate terms was conducted. This is in turn developed into a structured query using Indri's query language.

2.3 Concept Re-ranking

For this task, a concept term was taken to be an n-gram that described a patient. For example, "adult", "hypertension" and "end-stage renal disease" are concept terms. As the task requires finding patients that match criteria, these concept terms are

[2] `http://www.lemurproject.org/`

[3] `http://www.ncbi.nlm.nih.gov/pubmed/advanced`

[4] `http://www.nlm.nih.gov/mesh/meshrels.html`

very important in identifying relevant patients. The idea of re-ranking based on these concept terms was initially put forward in (Stokes et al., 2007) to alleviate the situation where a document containing multiple references to the same expanded concept term e.g. "high blood pressure", "HBP", "hypertension", "HT", would be ranked higher than another more relevant document that contained single references to all the concept terms in the query ("adult", "hypertension", "end-stage renal disease"). The re-ranking is a simple set of rules that will boost the ranks of documents that contain more unique concept terms (*ConceptNum*). The rules taken from (Stokes et al., 2007) are given below.

```
if $ConceptNum(D_1) > ConceptNum(D_2)$ then
    $Rank(D_1) > Rank(D_2)$
else if $ConceptNum(D_1) < ConceptNum(D_2)$ then
    $Rank(D_2) > Rank(D_1)$
else if $Score(D_1) \geq Score(D_2)$ then
    $Rank(D_1) > Rank(D_2)$
else
    $Rank(D_2) > Rank(D_1)$
```

2.4 Visit Ranking

In retrieving reports instead of visits, the influence of document length is lessened. However, some post-processing is required in order to obtain the ranking of visits as opposed to single documents. This section will discuss the post-processing performed.

Given a listing of relevant reports along with a mapping of reports to visits, this step aims to create a listing of relevant visits. Originally, for each visit, the ranking score of every document was to be summed, thus creating an overall relevance score for the visit. There are two disadvantages to this approach, firstly it will give preference to large visits, i.e. those that comprise of many reports. Secondly, it does not account for visits that may contain reports with a mixture of very low and very high relevancy scores.

In order to address these problems, the sum of scores for each visit was calculated using a geometric progression, similar to work in (Ravana and Moffat, 2009). The progression used is described in Equation 1 below:

$$visit\_score(V) = \sum_{n=0}^{D} \frac{score(n)}{r^{n}} \tag{1}$$

where *D* is the number of reports in visit *V*, *score(n)* is the score for the n^{th} ranked report in *D*. The variable *r* was given the value 2 after investigatory analysis by the authors.

This method of calculating the relevancy score for a visit reduces the impact of very low ranking reports while maintaining the importance of those with high ranks.

2.5 Run descriptions

Four runs were submitted to the medical track. Each is explained in full in this section. Each run builds on the last, allowing direct comparison of each component for each run.

- `UCDCSIrunOne` is a baseline run using Indri's structured queries to retrieve reports with the only post-processing performed involving the creation of the ranked visit list from the reports. The aim of this run is twofold, to highlight the advantages of structured weighted queries as well as language modeling techniques, while avoiding problems posed by disparity in visit lengths.

- `UCDCSIrunTwo` builds on the first run by incorporating concept re-ranking as described in Section 2.3

- UCDCSIrun3 uses the same parameters as UCDCSIrunTwo with the exception that it queries the visits index rather than the reports index. The aim here is to highlight the effect of the disparity in document lengths in the visit index.

- `UCDCSIrun4` combines two relevance scores (language modeling, Okapi) for each document in order to arrive at a new score for this document. The first score, based on language modeling, is generated from `UCDCSIrunTwo`. The Okapi score queries the report index using expanded queries. However, the Indri structured query language is not used, because it is

ID	bpref	R-prec	P @ 10
UCDCSIrun3	0.488	0.342	0.5
UCDCSIrunOne	0.378	0.315	0.4
UCDCSIrunTwo	0.277	0.249	0.3
UCDCSIrun4	0.166	0.106	0.2

Table 1: Results for four submitted runs

ID	bpref	R-prec	P @ 10
Max	0.794	0.606	0.794
Median	0.436	0.302	0.5
UCDCSIrun3	0.488	0.342	0.5

Table 2: Comparison of best performing run (UCDCSIrun3) with the Max and Median runs

not supported in conjunction with Okapi in the CMU system.

3 Experimental Results

As described in Section 2.5 four official runs were submitted to the TREC Medical Track this year by the UCD team. Three scoring metrics were used in evaluating the runs, namely *bpref*, *R-prec* and *Precision @10*. The track median scores of the submitted runs for each of the three metrics are shown in Table 1.

The comparisons in the table above allow us to see the effects of concept re-ranking as well as post-processing of documents to create visits. UCDCSIrun3 was by far the most effective run, outperforming all other runs for all metrics. The high performance score for UCDCSIrun3 can be attributed to the indexing of visits rather than reports. A drop in performance can be seen between UCDCSIrunOne and UCDCSIrunTwo, where concept re-ranking is introduced. UCDCSIrun4 saw the combination of UCDCSIrunTwo's relevancy scores with *Okapi* relevancy scores producing disappointing results. These findings will be analysed and discussed further in Section 4.

The scores for the maximum and median runs are shown in Table 2 with direct comparison to the best performing run, UCDCSIrun3. UCDCSIrun3 achieved moderate yet promising results, achieving higher scores than the median. However, this result possibly says more about the shortcomings of the post-processing approach, than it does about the effectiveness of the report index.

4 Discussion

In this section, we will discuss the advantages and disadvantages of our top three runs, inspecting the results on a per topic basis.

Figure 2 shows the per topic *bpref* score of the maximum and median participant results along with the authors' top performing run, UCDCSIrun3. Figure 1 shows for comparative purposes the per topic bref score of the three best performing runs submitted by the authors.

All three runs performed poorly on seven queries (103, 111, 118, 121, 124, 125, 133). The only common feature in these three runs was the use of query expansion and structured querying. Closer inspection revealed that expansions for these queries and also their structure were inadequate. For example, for Query 111, a mention of the concept of "intraspinal pain-medicine pump" was required. In our query expansion process, no expansions were found for this phrase or its constituent phrases e.g. "pain pump" in MeSH. Our query parsing proved to be too strict by searching for the exact phrase, and missing relevant document that used the phrase "pain pump".

Five topics, namely 104, 108, 116, 126 and 134 proved problematic for runs UCDCSIrunOne and UCDCSIrunTwo, with UCDCSIrun3 producing more acceptable results. The cause of the problem here was the sparse mentions of query concept terms coupled with the post-retrieval creation of visit rankings. If all the reports in a visit have a high rank, then the resulting visit will have a high rank. However, if one report has a high rank and and the other 10 reports in a visit are deemed not relevant, the relevance score for the overall visit will be significantly diluted. The use of a geometric progression in summing report scores to arrive at a visit score aimed to reduce this problem; however, the decay factor used was not strong enough in eliminating the effect of low scoring reports. Given that in UCDCSIrun3 the index is built from visits, the problem of low scoring reports is reduced as they are collectively treated as a single document.

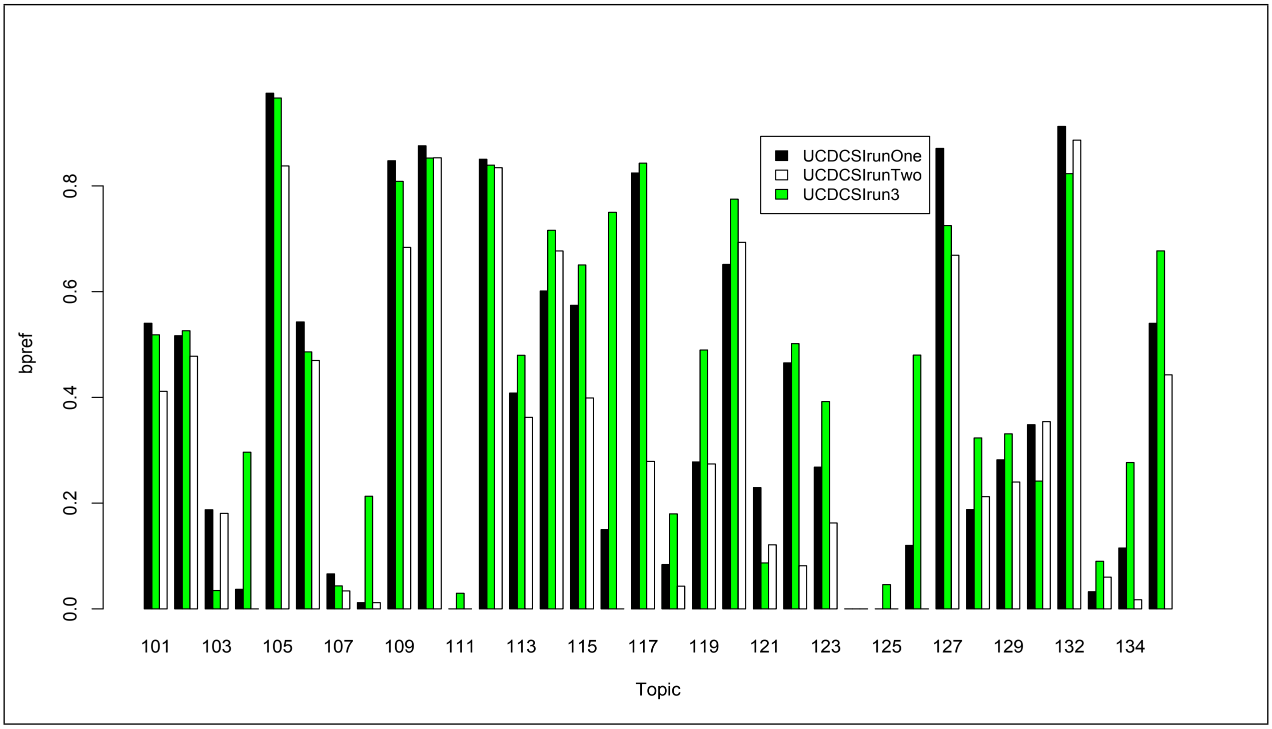

Figure 1: Bpref scores per topic for UCDCSIrunOne, UCDCSIrunTwo and UCDCSIrun3

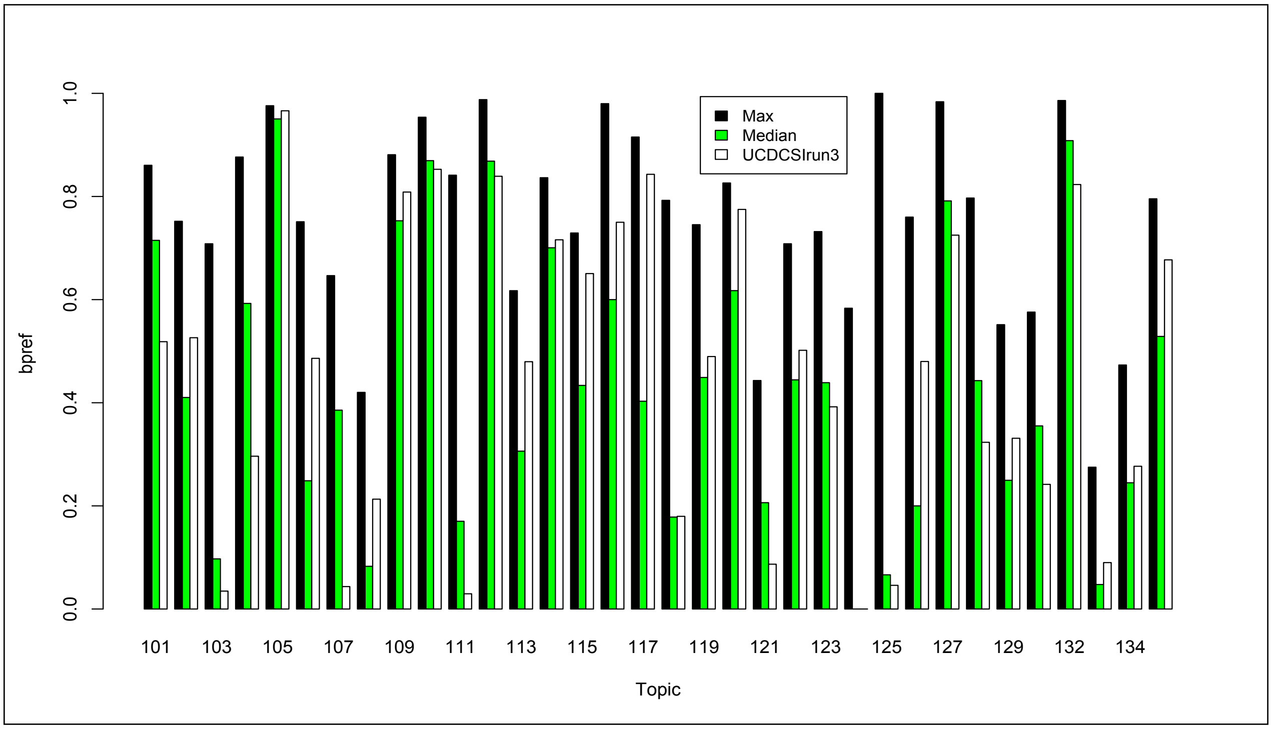

Figure 2: Bpref Max, Median and UCDCSIrun3 scores per topic

5 Conclusion

As part of TREC's medical track, we investigated the impact of document length and concept reranking on the retrieval of medical documents. The highest ranking run submitted by the authors' this year outperformed the median system at the track, showing it has some promise. However, the method and resources adopted for expanding queries were in some cases inadequate, leading to the low ranking of many relevant documents. Furthermore, the method of constructing visit ranks using a geometric progression to sum report scores was inadequate, as the decay factor chosen for this progression was too weak.

There are many possibilities for future work. Although `UCDCSIrun3` was the best performing run, its performance was limited, in part hampered by the disparity of document lengths. The post-processing of the report ranking results to produce a visit rank needs further refinement. As the query expansion resources used in this task produced inadequate results, it would be worthwhile to explore further expansion resources. Our systems tokenization and matching of query terms require additional refinement as they proved to be too strict in this task. Future work will focus on the relaxing of rules for phrasal matching.

References

W. D. Johnston, S. J. Nelson, and B. L. Humphreys. 2002. Relationships in medical subject headings (mesh). In *National Library of Medicine*.

D. Metzler and W.B. Croft. 2004. Combining the language model and inference network approaches to retrieval. In *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, volume 40, pages 735–750.

Sri Devi Ravana and Alistair Moffat. 2009. Score aggregation techniques in retrieval experimentation. In *Twentieth Australasian Database Conference*.

N. Stokes, Y. Li, L. Cavedon, E. Huang, J. Rong, and J. Zobel. 2007. Entity-based relevance feedback for genomic list answer retrieval entity-based relevance feedback for genomic list answer retrieval entity-based relevance feedback for genomic list answer retrieval. In *The Proceedings of TREC Genomics Track 2007*.