CWI at TREC 2011: session, web, and medical

Jiyin He
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
J.He@cwi.nl

Vera Hollink
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
V.Hollink@cwi.nl

Corrado Boscarino
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
corrado@cwi.nl

Roberto Cornacchia
Spinque
Science Park 123
1098 XG Amsterdam
roberto@spinque.com

Arjen de Vries
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
Arjen.de.Vries@cwi.nl

ABSTRACT

We report on the participation of the Interactive Information Access group of the CWI Amsterdam in the web, session, and medical track at TREC 2011. In the web track we focus on the diversity task. We find that cluster-based subtopic modeling approaches improve diversification performance compared to a non-cluster-based subtopic modeling approach. While gain was observed on previous years' topic sets, diversification with the proposed approaches hurt the performance when compared to a non-diversified baseline run on this year's topic set. In the session track, we examine the effects of differentiating between 'good' and 'bad' users. We find that differentiation is useful as the use of search history appears to be mainly effective when the search is not going well. However, our current strategy is not effective for 'good' users. In addition, we studied the use of random walks on query graphs for formulating session history as search queries, but results are inconclusive. In the medical track, we found that the use of medical background resources for query expansion leads to small improvements in retrieval performance. Such resources appear to be especially useful to promote early precision.

1. INTRODUCTION

The Interactive Information Access group of the CWI Amsterdam participated in three tracks: the web track, the session track and the medical track. For the web track, we focused on the diversity task. We used a novel method to find query aspects which is based on random walks on a large query graph. The goal of our participation in the session track was twofold. Firstly, we estimated the quality of a search session and studied the effects of differentiating between 'good' and 'bad' users. Secondly, we examined the use of random walks on query graphs for formulating query history as search queries. For the medical track the Search by Strategy framework of Spinque was deployed. We evaluate the use of external knowledge sources to improve medical content retrieval.

2. RETRIEVAL FRAMEWORK

For the web and session track, we use the Indri search engine[1] to index the clueweb09 collection (set A). We use the Waterloo Spam Rankings [4] to remove documents with a spam rate < 70. CMU PageRank scores[2] are used as document priors.

For the medical track, the Spinque framework was used (see Section 5.1).

3. WEB TRACK

We focus on the diversity task of the Web track. The goal of result diversification is to return a ranked list such that top ranked documents are not only relevant to the given query but also cover diverse aspects of the query. Following previous work in result diversification [1, 8], we employ a subtopic based approach in our experiments. Specifically, we aim to extract relevant and diverse subtopics of a query from query logs and use these subtopics for diversification.

3.1 Finding relevant and diverse subtopics

We extract subtopics for a query Q in two steps. First, we find *related queries* of Q in a query log via random walks. Second, we cluster these related queries and each cluster is used as a subtopic.

Finding related queries.

Following [5], we use a random walk based approach to find related queries in a click log. Let $\mathcal{U}$ be the set of urls in a click log, and $\mathcal{Q}$ the logged queries. A weighted graph can be constructed where the vertices are the union of the urls and the queries, and the edges correspond to the user clicks of a url for given a query, weighted by the number of clicks w.

A one step transition probability from node j to k can be calculated as $p_{t+1|t}(k|j) = w_{jk}/\sum_i w_{ji}$. Since we are only interested in the relation between queries, instead of constructing a query-url graph, we construct a query-query graph. The one step transition probability from query j to query k is given by

$$p_{t+1|t}(q_k|q_j) = \sum_i p_{t+1|t}(q_k|u_i)p_{t+1|t}(u_i|q_i). \quad (1)$$

[1] http://www.lemurproject.org/
[2] http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-index.php?page=PageRank

Further, we do not include an additional self-transition as proposed in [5]. Instead, we use the self-transition probability $p_{t+1|t}(q_i|q_i)$ derived from Eq. 1.

With the above defined one-step transition probability, we retrieve queries in $\mathcal{Q}$ for a given query Q and rank them in the decreasing order of their t-step transition probability. Similar to [5], we experiment with forward and backward walking. We then take the top X related queries for subtopic extraction.

Starting nodes.

While conducting random walk on the query-query graph, we start with the a given query Q, e.g., a new query that a user issued to the search engine. Since the query click log we use is a snapshot of the queries and clicks that happened in Web retrieval, Q does not necessarily occurs in the log. Further, queries that formulated slightly different from Q may still refer to the same topic of interest and therefore it may be also useful to include these queries as starting nodes.

Here, we use the query Q to retrieve top n queries from the query log using the Indri system and use these queries as starting nodes N. Then the t-step transition probability of Q to a query q in $\mathcal{Q}$ is calculated as

$$p_{t|0}(q|Q) = \sum_{n \in N} p(n|Q) p_{t|0}(q|n). \quad (2)$$

Subtopic extraction with clustering on related queries.

While each related queries can be seen as a subtopic of the query Q, we expect that many of them are variants of the same query and therefore not topically diverse. In order to extract a diverse set of subtopics, we cluster the retrieved related queries.

We use a spectral clustering approach, the normalized-cut approach [9] to cluster the query-query graph. For a given graph, the normalized-cut tries to create a partition on the graph so that a normalized cut criterion Ncut is minimized. The Ncut measures both the total dissimilarity between different groups and the total similarity within the groups. The normalized-cut has been proved to closely related to random walks [7]. In particular, if it cuts the graph into two parts such that the value of Ncut is small, then the one-step transition probability from a node in one partition to a node in the other partition is also small.

This property is especially appealing to our purpose. Intuitively, two queries with high transition probability are likely to be requests on a same topic, as they share a large amount of co-clicked urls. Further, it is relatively easy to apply Ncut within our experimental setup, since the same transition matrix that we have constructed from the query-query graph can be re-used for clustering.

To cut a graph, normalized-cut uses the Laplacian matrix of the graph. It finds a partitioning on the elements of the eigenvector corresponding to the second smallest eigenvalue of the matrix such that $Ncut(A, \bar{A})$ is minimized. The partitioning of the eigenvector corresponds to a partitioning on the graph. This procedure repeats until a stop criterion is met. The stop criterion determines the resulting number of clusters.

We find that for different Qs the retrieved number of related queries varies (random walks for some queries result in very small local graphs compared to others), and we expect that some queries have more diverse related queries than others. Therefore it is not desirable to set a fixed number of clusters for all Qs. Here we determine the stop-criterion for each query as follows. Assume in a graph there are m possible points where a cut can be applied. We evaluate if the chosen cutting point is better than a randomly picked cutting point. Intuitively, if all points are equally good for a cut, then the resulting clusters tend to be random. We calculate the expected value of the Ncut scores for the m cutting points $E(x) = p(x)Ncut(x)$, where $p(x) = 1/m$, assuming all cutting points are equally good. We then calculate the z-score of the Ncut value of the chosen cutting point and only apply a cut if the z-score exceeds a threshold. Heuristically, we set the threshold to 3.2.

3.2 Diversification with extracted subtopics

Once the subtopics are extracted, we use them to diversify an initially retrieved ranked list using the state-of-the-art result diversification approach IA-Select [1]. With IA-select the selection of a document is determined by its relevance to the query as well as the probability that it covers subtopics given that all previously selected documents fail to do so.

Given a candidate document set and a set of subtopics C of a query Q, the algorithm selects the document to be included in the returned set S from a candidate set R that maximizes the *marginal utility* at each step:

$$d^* = \arg\max_{d \in R} \sum_{c \in C} p(c|Q, S)\ V(d|Q, c), \quad (3)$$

where $V(d|Q, c)$ is a quality value of d that is computed using the retrieval score of d with respect to Q, weighted by the likelihood that d covers c. Further, $P(c|Q, S)$ is the conditional probability that Q is related to c, given that all documents in S failed to provide information on c.

Here, two essential components need to be calculated: The probability that a document d covers a subtopic $p(d|c)$ and the importance of a subtopic to the original query $p(c|Q)$. We implement these components in different ways, which results in three submitted runs.

Run1 - CWIIAt5b5.

This is our baseline run. Here, we simply use the top 10 related queries as subtopics of the original query. We then use IA-select to re-rank the initial ranked list with these 10 subtopics. For each subtopic, we estimate the $p(d|c)$ by calculating the query likelihood of the related query $q = c$ for a given document d using a multinomial language model.

$$p(d|c) = \prod_{t \in c} p(t|d)^{|t \in d|}. \quad (4)$$

Further, p(c|Q) is estimated as the t-step transition probability from Q to c random walks.

Run2 - CWIcIAt5b1.

For our second run, we cluster the top 100 related queries found by random walk, and use the resulting clusters as subtopics. Under this setup, we estimate the $p(d|c)$ using

$$p(d|c) = \sum_{q \in c} p(d|q) p(q|c), \quad (5)$$

where $p(q|c) = 1/|c|$, that is, we assume all related queries within a subtopic are equally probable to occur given the

RunID	#starting nodes	direction	#steps
CWIIAt5b5	5	back	5
CWIcIAt5b1	5	back	1
CWIcIA2t5b1	5	back	1

Table 1: Parameter settings for random walk over the query graph.

RunID	α-nDCG@20	P-IA@20	ERR-IA@20
Initial run	**0.458**	**0.246**	**0.369**
CWIIAt5b5	0.420	0.228	0.336
CWIcIAt5b1	0.431	0.221	0.347
CWIcIA2t5b1	0.432	0.230	0.349

Table 2: Results of diversity runs.

subtopic. $p(d|q)$ is calculated using Eq. 4. To estimate $p(c|Q)$, we have

$$p(c|Q) = \sum_{q \in c} p(q|Q), \tag{6}$$

where $p(q|Q)$ is the t-step transition probability.

Run3 - CWIcIA2t5b1.

For this run, we use the same strategy as run2. The only difference of the two runs is the estimation of $p(q|c)$ in Eq. 5. Here we no longer assume a uniform distribution for $p(q|c)$, instead, we weigh each query $q \in c$ with their t-step transition probability to the original query Q, and bias towards queries that are more likely to be related to Q.

$$p(q|c) = \frac{p(q|Q)}{\sum_{q \in c} p(q|Q)}. \tag{7}$$

3.3 Experimental setup

Our diversity runs are created by re-ranking the top 100 documents of an initial ranked list. We generate the initial ranked list using the Indri system as described in Section 2.

In our experiment, the Microsoft click log released in 2006 was used to extract related queries and to be grouped into subtopics. In order to determine the parameters of random walk used in our runs, we use the TREC 2009 and 2010 queries as training queries and use Set B as the training collection. The parameter settings for each run is listed in Table 1.

3.4 Results and Discussion

In this section we describe our results with an initial analysis of the results.

To our surprise, while on the training data, all three diversification runs outperform our initial retrieved ranked list, on this year's topic set, they all fail to improve over the initial retrieved results in terms of diversity measures. One possible explanation may be that the difference between the training and testing topic sets/collections (SetB for training and SetA for testing) is rather significant and different strategies or parameter settings should be applied. On the other hand, as described in the the Web track 2011 guidelines, this year's topics tend to be less ambiguous and are expected to have a lower number of relevant documents. As a result, click information would naturally be less reliable for this type of queries.

RunID	nDCG@20	ERR@20	P@20
Initial run	**0.201**	**0.115**	**0.278**
CWIIAt5b5	0.181	0.103	0.255
CWIcIAt5b1	0.181	0.108	0.248
CWIcIA2t5b1	0.183	0.108	0.254

Table 3: Results of our runs in terms of precision oriented measures.

A second observation is that within our submitted runs, the cluster-based runs (CWIcIAt5b1 and CWIcIA2t5b1) perform better compared to the baseline run (CWIIAt5b5), where single queries are used as subtopics. This suggest that cluster-based approach helps in extracting diverse subtopics from the click log. This trend is consistent with our experiments with the TREC 2009 and 2010 data.

A further analysis with the pure-precision based measure (Table 3) shows that our diversification runs all perform worse in terms of precision oriented measures, which partially explains the reason why they underperform in comparison to the initial retrieval results. That is, during reranking, non-relevant documents are pushed on to the top of the ranked list.

4. SESSION TRACK

The goal of the TREC 2011 session track is to evaluate whether and how a system's performance in response to a query at step t^* improves when information about various interactions at previous steps $t_i < t^*$ is included in the retrieval strategy. The 2011 TREC setting allows to submit the results of up to three different retrieval mechanisms to rank the last query of 76 search sessions based on the ClueWeb09 collection. The challenge is to improve a ranking in the last step of the search sessions by gradually adding more information: while the first task (RL1) does not consider any session feedback, progressively adding past queries (RL2), ranked results in response to those queries (RL3) and clicked documents (RL4) as well as dwell times is expected to lead to better performances.

4.1 Approach

Good users, bad users.

For each of the four tasks defined in the session track, we submitted three runs. For all runs, the methods for each task build on top of each other, i.e. each task adds some retrieval mechanism with respect to the previous task, but it does not alter previous methods that were already applied in the preceding tasks.

The first task (RL1) is a baseline without any session information; only the test query can be used, that is: only the final query submitted at step t^*. We limited query preprocessing to interpreting a sequence of N terms between single or double quotation marks as a windowed query of length N.

The rationale of the first two runs, CWIrun1 and CWIrun2, is to apply a user model that takes the increasingly more thorough session information as input and gives back as its output a set of weights. We use these weights to modify the representation of the last query in a language model retrieval system.

The goal of the first two runs is to investigate whether

an exponential discount model simulates the contribution of interactions before t^* to the retrieval model used to generate the ranked list at t^* and how far the parameters of this model can be extracted from the session information only, without a more thorough user profiling. Our proposed model, more specifically a model of how interactions in the past should be discounted, has fixed parameters for all the sessions in the first run, whereas in the second run the discounting ratio varies between different sessions.

The intuition behind these discount functions stems from behavioural studies on the effect of past experiences on human attitudes. At least in the rather special case of compulsive attitudes, such as excessive gambling and indulgent drinking habits [10, 11] negative experiences have demonstratedly a much stronger impact [2] than experiences that a subject would label as positive. Behavioural scientists suggested both exponential and hyperbolic discount models [6] with fixed or variable discount rates. Since the system that is supposed to generate the collection does not have any notion of past sessions when generating the ranked lists at each step, we considered only exponential discount functions. This model simulates interactions with a memoryless system.

We aim to investigate whether a discount strategy that is functionally similar to this behavioural model can be applied to a user's search history. Our hypothesis is that this discounted history, once included in the last query's representation, improves retrieval.

A fixed discount rate model assigns higher absolute weights to negative information or experiences, whereas a variable discount rate model assigns lower discount rates to negative information, but absolute weights can possibly be drawn from other features. Because of this flexibility, we map metrics for the quality of a search experience to the discount rate parameter of the model.

We define a 'good' user as a user who is capable to issue well-posed queries to a system and, because she learns from interactions, her search experience is poised to be rewarding. A perfectly good user needs a variable amount of steps, where the number of steps only depends upon the difficulty of the topic, to learn a query that alone can produce a satisfactory result. No amount of history will ever outperform a final query of this perfectly good user. On the other hand, a 'bad' user should might just be unaware of a system's fallacies and fail to cope with the retrieval strategy; additional interactions do not improve her capability to issue more effective queries. Our hypothesis is that for satisfying the needs of a bad user, search history is equally valuable as the last query.

A unsuccessful learning step plays thus in our application the role of a negative experience in a behavioural setting. In our hypothesis a negative search experience is negative not because does not provide any information at all on a topic, but just because a user did not succeed in carrying the partial information on the topic further to the final query. A system should therefore supplement a user on this task. Vice versa a positive (search) experience mirrors knowledge that already contributes to the present behaviour. In both cases we only put forward a functional similarity between experiences and learning steps during a search session, without any claim on the relevance of partial interactions to session topic: *if* there were relevant aspects of the topic in the partial interactions *then* a good user would carry them up to the final query.

In the first run, CWIrun1, we assume an average user in between these two extremes, i.e. a user who during an average number of interactions L gets to know a system well enough to issue queries of reasonably good quality. Under these assumptions, we calculate query terms weights w_t for task RL2 according to:

$$w_t = e^{Q \cdot I_{diff} \cdot t},$$

where $Q = \frac{1}{2}$ and I_{diff} is the ratio between the average amount of interactions recorded in the log data and the number of interactions in a particular session.

In the second run, CWIrun2, we attempt to extract the quality parameter Q from the query terms: our assumption is that a good user chooses highly selective terms that are important in a idf sense. We therefore correct Q, subtracting from the CWIrun1 value ($Q = \frac{1}{2}$) the average term importance over all query terms, normalised to take into account different query lengths. Our assumption here is that query length is not an indicator of user quality, but it depends mostly on the topic.

Task RL3 allows for the use of the ranked lists in response to each intermediate query. Our assumption in this case is that a good user will interact with a system in order to investigate either different aspects of a topic or different features of the system. In both cases we expect little overlap in the result set, at least locally from one step to the next one; even when the queries seem very similar to each other. If the average overlap in consecutive steps, weighted for the average session overlap, as some topics may be very specific and inherently prone to result in highly overlapping retrieval sets, appears to show more than 10% overlap, the user quality is decreased with the same amount. As this is already a session dependent parameter, we do not test any alternatives in the second run.

Finally our submission for task RL4, which allows using full log data, builds further on the second run of the preceding sessions, but expands a query with 5 additional terms from each clicked document, if any. These 5 terms are the most important terms according to a tf-idf metric, where the idf is calculated on the entire collection, and they are weighted by the same user model as the user generated query.

Random walk on the query graph.

In the first two runs, terms from clicked documents where used to expand the user's last query. However, these terms were not designed to be queries and are not necessarily suitable as query terms. In the third run of the session track, CWIpostRW, we examine whether the representation of the interaction information in the queries can be improved by exploiting queries that users have issued previously to a web search engine. We used a random walk on a query graph of a major search engine (see Section 3) to find web search queries that represent similar information needs as the TREC sessions.

We made use of the same query graph that was used for the web track (see Section 3). On the basis of the interaction information we constructed Indri queries in a similar way as for CWIrun1. For RL2 and RL3 the users' previous queries (not including the last query) became the query terms. An exponential discount function with a fixed exponential was

	RL1	RL2	RL3	RL4
`allsubtopics`				
CWIrun1	0.2427	0.2434	0.2481	0.2481
CWIrun2	0.2392	0.2424	0.2481	0.3114
CWIpostRW	0.2426	0.2571	0.2571	0.2496
`lastquerysubtopics`				
CWIrun1	0.2047	0.1633	0.1677	0.1677
CWIrun2	0.2012	0.1618	0.1677	0.1997
CWIpostRW	0.2019	0.2115	0.2115	0.2048

Table 4: Expected Reciprocal Rank (ERR) of the session track runs

used to give lower weights to less recent queries. For RL4, we used the users' previous queries as well as terms from the clicked documents with high tf-idf values. Weights were based on tf-idf score and recency.

The constructed queries were used to retrieve an initial set of web queries from our collection of web queries (see Section 3). The web queries functioned as starting points for a random walk on the click graph. As in run CWIIAt5b5 of the web track, a backwards walk was performed with at most 5 steps. The 10 web queries that received the highest probabilities in this walk were selected to represent the users' session history.

For each selected query we used Indri to compute the probability for the top 1000 documents retrieved by the users' last query. The final probability of a document was computed as a linear combination of the probabilities based on each of the selected queries and the probabilities based on the last query.

4.2 Results

The TREC organisation measures performance under two conditions. In the `allsubtopics` condition, a document is considered relevant if it is relevant to one of the subtopics of the main topic of the session. In the second condition, `lastquerysubtopics`, a document is considered relevant only when it is relevant to a subtopic to which the last query in the session refers.

In Table 4 we report the Expected Reciprocal Rank (ERR) for our 3 runs. From the average results of CWIrun1 and CWIrun2 we can already notice that the `allsubtopics` condition favours our method. Under that condition, although our task RL1 performs below median (0.2392 vs 0.24295), by using full log data, we achieve an improvement on that baseline of more than 30% by using session history (RL2, RL3, RL4) whereas the median does not exceed 5%. However, differences between the four tasks are not significant (Wilcoxon signed-rank test, 2-sided $p<.05$). Some sessions are even more remarkable (such as session 64): for that, while our baseline task RL1 performs well below median, we reported the highest score among all the participants when adding session information.

At a session level we observe that when our baseline RL1 provides already reasonably good results (such as for sessions 51, 59, 60, and 61) our method does not affect much the initial performance, in accordance with the median. However, when the score at the baseline turns out to be exceptionally good (such as for session 8 or 68) our methods performance rapidly decreases, again in accordance with, but at a higher rate than the median. In summary these result reports on a method that improves on weak baseline results, is relatively neutral to an average baseline, but makes things worse when the baseline is already adequate.

While the final version of this document will contain a more thorough analysis of the results, the main lesson that we draw from this preliminary assessment is that our method seems able to capture the search process of bad users. Those are the users who perform badly at the baseline. They do not grasp the system features or they do not manage to work around its fallacies. Given the diverging performance under the `allsubtopics` and `lastquerysubtopics` conditions this method seems more useful to support the exploratory phase of a search process.

As a future work we should concentrate on dealing with good users. While session data readily reports on a learning process, we should improve the way this knowledge is used to update the query representation. A fixed quality parameter does not seems to provide the required flexibility, as the disappointing results in run 1 seems to indicate. We aim therefore to a well trained session dependent model that takes into account good users: because they learn from interactions, a system must 'forget' their intermediate and less useful results.

As shown in Table 4, using the query graph to formulate queries results in a marginal, non-significant, improvement over the baseline: the results for RL2, RL3, and RL4 are only marginally higher than the the results for RL1. We assess the added value of using the query graph for query representation by comparing this run to the CWIrun1 run. In the `allsubtopics` condition we again see the results are similar, but in the `lastquerysubtopics` condition the query graph method significantly outperforms the method without query graph (for RL4: Wilcoxon signed-rank test, 2-sided $z=2.19$, $p=.0285$). While in the latter condition the performance of CWIrun1 (like the track median) drops when query history is taken into account, using query history by means of the query graph marginally improves performance. This may be an indication that the query graph provides an effective way to use the wisdom of the crowd for representing query history. However, it may also be the case that the results in all four tasks look similar as a result of the conservative reranking scheme that was used. In further experiments we will examine the added value of this method in more depth.

5. MEDICAL TRACK

The goal of the Medical track is to retrieve "visits", a group of hospital discharge reports, given a query requesting information about patients with certain diseases that have certain conditions or treatment.

In this year's participation of Medical track, we aimed to exploit external resources to improve retrieval performance. There exist many medical ontologies/taxonomies that contain information about specific diseases, their definitions, diagnoses, symptom descriptions, etc. These resources can be used to enrich our queries that are usually short sentences and we expect that the retrieval systems can benefit from such enrichment. For instance, a query expansion with external medical taxonomy may be a typical way of reducing term mismatching between queries and the medical reports. In particular, we use the International Statistical Classification of Diseases and Related Health Problems (ICD-10) as our external resources.

Unlike in Web track and Session track, in the Medical track, we use the Spinque retrieval system, which provides a flexible way for incorporating external resources. Below, we first introduce the Spinque system. We then describe our submitted runs, followed by an initial analysis of the evaluation results.

5.1 The Spinque retrieval system

We modeled and executed our runs for the Medical Track as *search strategies* within the Spinque framework. In this approach, nick-named 'search by strategy', search processes are divided into two phases: the search strategy definition and the actual search.

Modelling search strategies in this framework corresponds to designing graph structures, where edges represent data-flows consisting of terms, documents (e.g. medical reports), document-sections (e.g. diagnoses) and named entities (e.g. ICD codes, visits - which identify groups of reports belonging to the same issue). The nodes connected by such edges are general-purpose (but customisable) operational blocks, that either provide source data (the medical report corpus and the topics corpus) or modify their input data-flow applying operations such as selection based on ICD codes, extraction of specific sections from documents, or ranking of sections and documents, to name a few. In Figure 1 we show an example of the designed graph structure of a search strategy, which corresponds to our baseline run described in Section 5.2.2.

Search strategies defined in this framework are automatically translated into a probabilistic relational query language and executed on top of an SQL database engine.

A major advantage of the strategy-based approach is that input and output pins of a graph's blocks are typed (documents, terms, named entities, etc) and certain blocks can change their input type to a different type in output. For example, a block can retrieve all ICD code named entities mentioned in a ranked list of documents in input. When connected on compatible types, it takes just a few blocks to express in one simple strategy complex needs composed by different retrieval units, such as e.g. "(1) use the topic text to identify relevant ICD codes; (2) use these codes as input to perform an ICD code-based search of medical reports; (3) change the retrieval unit into visits by aggregating retrieved reports".

5.2 Approaches

5.2.1 Indexing

Two collections are used in our experiments: the collection of medical reports released by TREC, referred to as the *target collection*, and the ICD-10 taxonomy, referred to as the *external collection*. Note that because ICD-9 codes are used instead in the target collection, we have enriched the ICD-10 taxonomy to include ICD-9 codes as well.

Textual data was indexed by importing xml data in an xml-powered relational dbms (MonetDB/XQuery [3]), and mimicking an inverted file structure on relational tables, after a standard Porter stemmer was applied. Furthermore, additional information about the structure of the collections was extracted and stored as [subject,predicate,object] triples with an additional probability column. All triples in the index are assigned probability 1, while intermediate results carry computed probabilities. Important relations for the two collections include the following information:

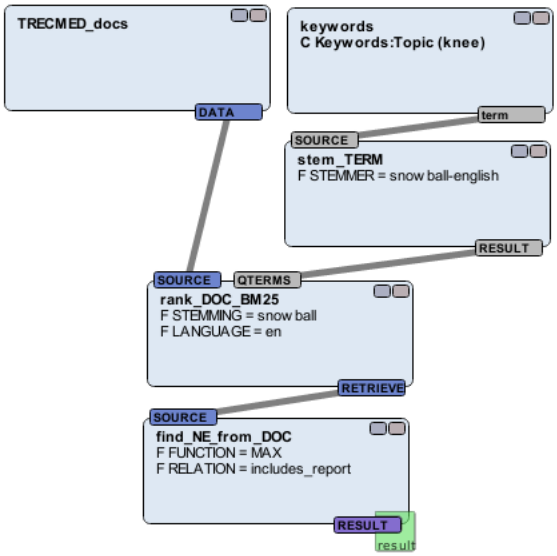

Figure 1: Strategy for Medical Track, run CWI1

Target collection

- reports (documents)
- visits (named entities)
- triples $[visit, \texttt{includes}, report]$
- triples $[report, \texttt{has\_discharge\_diagnosis}, ICDcode]$

External collection

- ICD codes (named entities)
- description,includes,excludes (textual sections)
- triples $[section, \texttt{belongs\_to}, ICDcode]$

5.2.2 Runs

In total we submitted 4 runs. Below we describe them in detail.

CWI1: baseline run.

As our baseline run, we apply BM25 to retrieve reports using all the available text and finally aggregate them into visits (visits may consist of several reports), using the following approach:

$$Score(visit) = \max_{d \in D} (Score(d) \cdot Score([visit, \texttt{includes}, d]) \quad (8)$$

where $Score(d), d \in D$ is the score of document d, as computed at any previous computational step, from a document candidate set D, and $Score([visit, \texttt{includes}, d])$ denotes the score of the event "*visit* includes report d" as described by our index (in our case, this score is either 0 or 1).

CWI2: Combining rank lists.

For the second run, we incorporate the information contained in the ICD10 taxonomy by combining a ranked list retrieved using ICD codes with the baseline run.

Let $R1$ be the baseline ranked list of reports retrieved using BM25 (see CWI1). We create a ranked list $R2$ as follows. First, we use BM25 to retrieve a ranked list of ICD codes based on the estimated relevance of their 'description' section to the query. Then, the report documents that mention the ICD codes found at the previous step are selected and ranked accordingly – the more high-scored ICD codes match a report, the higher the report's score.

Finally, we linearly combine the two ranked lists:

$$Score(d) = \alpha Score_{R1}(d) + (1-\alpha) Score_{R2}(d) \qquad (9)$$

where d is a medical report, and aggregate reports into visits using Equation 8.

As we do not have training data for finding the optimal value of α, we tested different values of α with the 4 example queries released by TREC for development. We set α to 0.084 based on the test. It gives surprisingly low emphasis to the initial baseline scores, although given the number of development queries, this value is most likely far from optimal.

CWI3: Query expansion.

In this run, we incorporate the ICD taxonomy using a query expansion strategy. First, a ranked list of ICD-code description sections is retrieved using BM25 with the topic query. Only the 10 top-ranked description sections are considered, and all terms $t \in T$ from those sections are ranked using their normalized aggregated score for the considered section group E:

$$Score_E(t) = \frac{isf(t) \cdot \sum_{s \in E} tsf(t,s)}{\max_{t' \in T} \left(isf(t') \cdot \sum_{s \in E} tsf(t',s) \right)}$$

Let $isf(t)$ denote the inverse section-frequency of term t and $tsf(t,s)$ the term-section frequency of term t in section s. The 10 top-ranked terms from this list are used to expand the original query. This run uses a linear combination to mix the original and the extracted query terms, whose scores are denoted by $Score_Q(t)$ and $Score_E(t)$ respectively:

$$Score(t) = \alpha Score_Q(t) + (1-\alpha) Score_E(t)(d).$$

The new queries obtained for this run were formulated assigning a weight of 0.99 to α, empirically found, and used to rank report documents with BM25. Finally, reports are aggregated into visits using Equation 8.

CWI4: Combination with alternative query.

This run is similar to run CWI3, in that it extracts additional terms from the ICD taxonomy. Additional terms are however not used to expand the original query, but rather to build alternative queries. The original and alternative queries are used to create two ranked lists of reports, which are then linearly combined.

Again, the topic query is used to retrieve the 10 top-ranked description sections of the ICD taxonomy. From each section, the 3 top-ranked terms are extracted, using normalized scores based on inverted section and term-section frequencies, but not aggregated:

$$Score_E(t,s) = \frac{isf(t) \cdot tsf(t,s)}{\max_{t' \in T, s' \in S} \left(isf(t) \cdot tsf(t,s) \right)}$$

This yields 10 new queries $q' \in Q'$, of 3 terms each, which are used to create one new ranked list of reports, by aggregating at document level using the max function:

$$Score_{R2}(d) = \max_{q' \in Q'} \left(BM25(d,q') \right)$$

The rationale for not aggregating terms into a single large query is to preserve the context of where terms co-occur and avoid to generate false term co-occurrence assumptions (e.g. it terms A,B are in query 1, terms B,C in query 2, avoid the false co-occurrence of terms A and C).

If we let again $R1$ be the baseline ranked list of reports retrieved using BM25 (see CWI1), the final report ranking is given by a linear combination of rankings $R1$ and $R2$, as in Equation 9, for which the best value for α was empirically found at 0.98. Reports were aggregated into visits using the same approach as in run CWI1, as in the previous runs.

RunID	R-prec	bpref	P@10	P@5
CWI1	0.2641	0.3568	**0.3882**	0.3824
CWI2	0.2664	0.3507	0.3676	0.3647
CWI3	**0.2676**	0.3573	**0.3882**	**0.4059**
CWI4	0.2648	**0.3577**	0.3794	0.4000

Table 5: Results of our submitted runs.

5.3 Results

In table 5 we list the evaluation results of our submitted runs in terms of R-prec, bpref, P@10 and P@5. In general, the impact of incorporating external resources, positive or negative, is marginal. Run CWI2 is in general worse compared to the baseline, which suggests that ranking using ICD codes contained in the medical reports may not be effective. On the other hand, CWI3 and CWI4 improves over the baselines in most of the cases, although the improvements are marginal. In particular, these two runs have a relatively obvious improvement over the baseline run in terms of P@5. This suggests that query expansion with external medical taxonomy may be useful to promote early precision.

One final remark is that since we do not have sufficient data to determine the optimal parameter settings, more extensive experiments and analysis need to be done in order to draw a conclusion on the usefulness of incorporating external resources in medical content retrieval.

6. CONCLUSION

For the diversity task of the web track. We used a random walk based approach to find related queries in a query log and use a spectral clustering approach to extract subtopics from the related queries. We find that cluster-based subtopic modeling approaches improve diversification performance compared to a non-cluster-based subtopic modeling approach. However, while gain was observed on previous years' topic sets, diversification with the proposed approaches hurt the performance when compared to a non-diversified baseline run on this year's topic set.

From our experiments in the session track we conclude that including session history can improve search performance, especially in the exploratory phases of a search session. However, differentiating between 'good' and 'bad' users is essential, as with our current strategy results deteriorate when searchers are doing well. We observe a marginally positive effect from the use of random walks on a query graph for query formulation.

In the medical track we found that the use of medical background resources for query expansion leads to small improvements in retrieval performance. Using such resources appears to be especially useful to promote early precision.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *WSDM '09: 2nd ACM International Conference on Web Search and Data Mining*. ACM, 2009, pp. 5–14.

[2] R. F. Baumeister, E. Bratslavsky, C. Finkenauer, and K. D. Vohs, "Bad is stronger than good," *Review of General Psychology*, vol. 5, no. 4, pp. 323–370, 2001.

[3] P. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner, "MonetDB/XQuery: a fast XQuery processor powered by a relational engine," in *Proceedings of the International Conference on the Management of Data (ACM SIGMOD)*, 2006, pp. 479–490.

[4] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke, "Efficient and effective spam filtering and re-ranking for large web datasets," *Information Retrieval*, vol. 14, pp. 441–465, 2011.

[5] N. Craswell and M. Szummer, "Random walks on the click graph," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '07. New York, NY, USA: ACM, 2007, pp. 239–246.

[6] J. Leslie, J. Leslie, and M. O'Reilly, *Behavior analysis: foundations and applications to psychology*. Harwood Academic Publishers, 1999.

[7] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *AI and Statistics AI-STAT*, 2001.

[8] R. L. T. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *WWW*, 2010, pp. 881–890.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[10] M. R. Stieg, Matthew D.; Dixon, "Discounting of past and future rewards of texas hold'em gamblers," *European Journal of Behavior Analysis*, vol. 8, pp. 93–97, 2007.

[11] R. E. Vuchinich and C. A. Simpson, "Hyperbolic temporal discounting in social drinkers and problem drinkers." *Experimental and Clinical Psychopharmacology*, vol. 6, no. 3, pp. 292–305, 1998.