Autoadapt at the Session track in TREC 2010

M-Dyaa Albakour, Udo Kruschwitz, Jinzhong Niu, Maria Fasli

School of Computer Science and Electronic Engineering, University of Essex,
Wivenhoe Park, Colchester, CO4 3SQ, UK
{malbak,udo,jniu,mfasli}@essex.ac.uk

Abstract. This paper provides an overview of the experiments we carried out at the TREC 2010 Session Track. We propose an approach for interpreting reformulated queries by using query expansions derived from simulated query logs. We show that this approach can improve the retrieval performance over a baseline system.

1 Introduction

The session track has been introduced at the Text REtrieval Conference (TREC) 2010. The session track aims to evaluate the ability of search engines to utilise previous user interactions in order to provide better results for subsequent queries in a user session and therefore 'point way' to what the user is actually looking for.

Our contribution to the session track is based on the idea that related queries can be derived from queries submitted within the same session. The wider context is the AutoAdapt project[1] which looks at automatically building and adapting domain models form the users' search and browsing behaviour (using query logs). These domain models are used to assist users to find information by suggesting query modification or browsing suggestions in their search. We have shown that learning query modification suggestions based on log data using adaptive algorithms (such as an ant colony optimization approach) can be effective [4]. We envisage that these adaptive domain models can be particularly useful for the problem introduced in the session track. Our approach is to use these models to apply query expansion based on query relations extracted from large query logs.

Due to the lack of availability of query logs suitable for this year's task, we used anchor logs instead. Anchor text has shown to be effective for a variety of information retrieval tasks. This includes ad-hoc search and the diversity task [9],[2]. Anchor text can be considered as a replacement to user queries as often web authors use similar labels to describe web pages to those used by searchers to find them [5]. Moreover, Dang and Croft have recently shown how anchor text can be used to simulate user sessions. They have considered all the anchor text pointing to the same document as queries in the same user session [3]. In this work we adopted this technique to simulate query sessions.

[1] http://autoadaptproject.org

Our objective of taking part in this year's session track is to see whether adaptive models created from simulated query logs can be actually utilised in interpreting query reformulations. We show that expanding the reformulated query using query terms and phrase derived from these models can improve the performance over a baseline system.

The rest of the paper is structured as follows. In section 2 we give a brief description of the task introduced this year. We describe the dataset and the resources used in our runs in section 3. We explain the experiments and the runs submitted to TREC in section 4. The results of those runs are then discussed in section 5. Finally, we give a brief conclusion and a plan for future work in section 6.

2 The Task

The session track tries to evaluate the effectiveness of search engines in interpreting query reformulations. The task of the session track in TREC 2010 is described in [8], [1]. Participants are given a set of 150 query pairs, each query pair (original query, query reformulation) represents a user session. These pairs were simulated from the TREC 2009 Web track 2009 diversity topics [8]. The participants are asked to submit three ranked lists of documents form the ClueWeb09 dataset:

- One for the original query ($RL1$).
- One for the query reformulation ignoring the original query ($RL2$).
- One for the query reformulation taking the original query into consideration ($RL3$).

Based on previous work in analysing query logs, the session track identifies three different types of query reformulations. Each query pair provided to participants is considered to belong to one of these types. The session types as explained in [1],[8] are

1. **Generalisation**: In this case, the user starts with a query and gets back some results which may be too narrow or they realise that they wanted a broader spectrum of results, so they reformulate to a more general query. e.g.'low carb high fat diet' → 'types of diets'.
2. **Specification**: In this case, the user starts with a query and gets back some results which may be too broad or they realise that they wanted results within a specific category or subtopic, so they reformulate to a more specific query. e.g. 'us map' → 'us map states and capitals'
3. **Drifting/Parallel Reformulation**: This type represents the case of starting with a query and then reformulating with another query at the same of level of specification but with a different aspect of information need. e.g. 'music man performances' → 'music man script'.

The type of query reformulation is not known to the participants. In our runs we did not attempt to automatically classify a query pair into one of the three categories and therefore we treated all pairs the same.

3 The ClueWeb09 Dataset, Indri Search Engine and The Anchor Log

The ClueWeb09 dataset[2] is a web crawl of more than a billion pages that has been used in last year's Web track. The ClueWeb09 category B dataset is a subset of the larger ClueWeb09 crawl and it consists of 50 million English pages. In this year's task participants were permitted to use either one of the two datasets. An existing Indri[3] index of the ClueWeb09 dataset is already available and searchable via a public web service[4]. The web service would enable us to issue queries and retrieve the top documents returned by the search engine, thus removing the burden of indexing the data internally. The Indri search engine uses language modelling probabilities and supports query expansion.
In our experiments we aim to use anchor logs to simulate query logs. The anchor log for the dataset has been processed and made publicly available by the University of Twente[5]. Each line in the log represents a document in the collection with all the anchor text pointing to the document[7]. We used the anchor log file of the ClueWeb09 Category B dataset. This file contains 43 million lines and thus contains anchor text for about 87% of the documents. Each line is tab separated and consists of the document TREC identifier, its url and all the anchor text pointing to that document.

clueweb09-en0000-23-00060 *http://001yourtranslationservice.com/dtp/* 'website design' 'DTP and Web Design' 'Samples' 'programmers' 'desktop publishing' 'DTP pages' 'DTP samples' 'DTP and Web Design Samples' 'DTP and Web Design Samples' 'DTP and Web Design Samples' 'DTP and Webpage Samples' 'DTP' `http://001yourtranslationservice.com/dtp/`

Fig. 1. A sample of the anchor log file

Figure 1 shows a sample line in the anchor log file. We add quotation marks to group anchor text fields for illustration purposes.
In the next sections describing our runs, we will use the following terminology. For a query q consisting of a number of terms qt_i, our reference search engine (The Indri search engine) would return a ranked list of documents using the query likelihood model from the ClueWeb09 category B dataset:
$D_q < d_{q,1}, d_{q,2}, ..., d_{q,n} >$ where $d_{q,i}$ refers to the document ranked i for the query q based on the reference search engine standard ranking function.

[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[3] http://lemurproject.org/indri.php

[4] http://boston.lti.cs.cmu.edu:8085/clueweb09/search/cataenglish/lemur.cgi

[5] http://wwwhome.cs.utwente.nl/ hiemstra/2010/anchor-text-for-clueweb09-category-a.html

4 Runs

Participants in the session track are asked to submit up to three runs. This year we are proposing three different runs (systems). Two of those runs are considered baseline systems to which we will compare our proposed method against. In all our runs (essex1, essex2, essex3) we will we use the Indri search engine to provide the first two ranked lists ($RL1$) and ($RL2$) by submitting the original query q and the reformulation r, i.e. the first two ranked lists will be D_q and D_r respectively. The maximum number of returned documents in both lists are limited to 1000.
We also used the Waterloo Spam Rankings[6] for the ClueWeb09 dataset to filter the spam documents from the returned ranked lists. Table 1 illustrates the ranked lists matrix of our three runs. In the following subsections we explain how we produced the ranked list $RL3$ for each run.

	RL1	RL2	RL3
essex1	D_q	Dr	(baseline 1)
essex2	D_q	Dr	(baseline 2)
essex3	D_q	Dr	(AutoAdapt Approach)

Table 1. The Runs matrix

4.1 The first baseline - essex1

This baseline represents the simplest way of using previous user interaction with the search engine to interpret reformulated queries. This is done by submitting a new query $q + r$ to our search engine where the terms in this query is the set $qt \cup rt$. i.e. the system will return the ranked list D_{q+r} as ($RL3$).

4.2 The second baseline - essex2

This baseline reflects on the assumption that the users are not satisfied with the first set of results and that is why they reformulated their original query. Therefore one possible naive way to utilise the previous query is to filter the results for the next query by eliminating whatever appears in the result set returned for the original (first) query. In this baseline, for the ranked list ($RL3$) we return the ranked list: $D_r - D_q = \{d; d \in D_r, d \notin D_q\}$
The documents in $D_r - D_q$ are ordered using their ranking in D_r
Figure 2 illustrated this filtering process. Note that we are filtering the top returned documents not the entire result set.

[6] `http://durum0.uwaterloo.ca/clueweb09spam/`

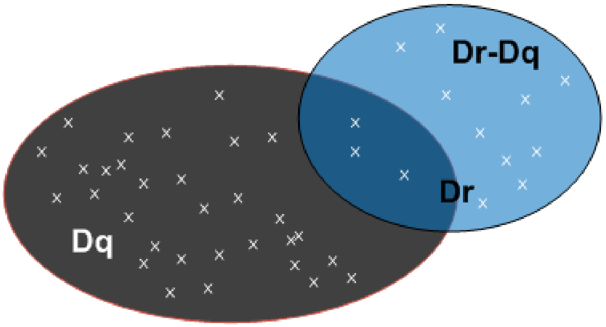

Fig. 2. Illustration of the filtering process

4.3 The AutoAdapt Approach - essex3:

In this run we developed a method for extracting useful terms and phrases to expand the reformulated query in the session. Our method stems from previous work in using query logs to extract related queries and our work in the Autoadapt project to learn domain models from query logs. As described in the previous sections we used an anchor log constructed from the same dataset (the ClueWeb09 category B dataset) to simulate query logs. We consider all the anchor text pointing to one document as a set of queries in a user session. Following these assumptions we can derive suggestions for a user query using association rules proposed by Fonseca et al.[6]. The intersection of suggestions extracted for both queries in the session can be considered useful for query expansion of the reformulated query as they can provide an approximation of the potential user session route.

The following steps were taken for each query pair to extract the query expansion terms and phrases:

- We remove all the stop words from both queries in the session.
- From the anchor log, we extract all the lines (the sessions) in which the anchor text contains either queries.
- If one of the queries is entirely contained in the other one, i.e the queries looks like XY:X or vice versa, then we treat the pair as the pair X,Y.
- Using the association rules approach for log data proposed by Fonseca et al.[6] we extract all the suggestions for both constituents X,Y. In this step, we considered some anchor terms as stopwords and filtered out these stopwords from the anchor text. This is due to the observation made previously by Eiron et al. that links within the site are often navigational links and they results in anchor terms such as ‘click’, ‘next’, ‘here’[5].
- We consider the top 10 phrases or terms in the set composed by the intersection of the suggestions extracted for both constituents as useful expansions to the reformulated query plus the original query.
- To generate the ranked list $RL3$, we submit the following query (using Indri operators to weight the different query terms):

$$\begin{array}{c}\#\,\text{combine}(\\ 0.7\ \#\,\text{combine}(\ rt_1\, rt_2\, ..\, rt_n)\\ 0.3\ \#\,\text{combine}(\ q\, e_1\, e_2\, ..\, e_{10})\\ )\end{array}$$

where rt_i are the individual terms in the reformulated query r, q is the query phrase of the original query q and e_i is an expansion term or phrase extracted as explained in the previous step. Note that in the case where no expanded terms or phrases are extracted in the previous step, we are only expanding with the original query.
Table 2 shows some the extracted expansion terms and phrase for 3 different pairs.

Session	**Expansion terms or phrases**
gps devices → 'garmin'	'gps devices', 'wikipedia','usb', 'gps device', 'gps products', 'garmin nuvi880', 'garmin gps device','visit garmin'
computer worms → malware	'computer worms','computer security', 'category','worm'
us geographic map → us political map	'us political map','article'

Table 2. Example of expansion terms or phrases extracted for three query pairs

5 Results

Table 3 shows the overall performance of our 3 runs using the measures provided by NIST. It also contains the summary results of all participants in the track. The figures in Column 2 represent the session normalised discounted cumulative gain $nsDCG(RL13)$ and used as the main measure for task 2 (evaluating the performance over the entire session). We use the difference $nsDCG(RL13) - nsDCG(RL12)$ to compare the performance of our systems for task 1 (testing whether system can improve their performance by using information from previous queries)

We summarise the findings of analysing these results as follows:

- Each of 'essex1' and 'essex3' systems has achieved a marginal overall improvement of retrieval performance for $RL3$ over $RL2$ i.e. they were both capable of using previous queries to improve retrieval performance.
- The AutoAdpat approach 'essex3' has the best retrieval performance over the two baselines 'essex1' and 'essex2' for both tasks. This suggest that using the anchor log for query expansion was useful.

System	$nsDCG.RL12$	$nsDCG.RL13$	$nDCG.RL1$	$nDCG.RL2$	$nDCG.RL3$
essex1	0.2154	0.2231	0.2077	0.2215	0.2348
essex2	0.2154	0.1993	0.2077	0.2215	0.1700
essex3	0.2154	0.2246	0.2077	0.2215	0.2456
min	0.0666	0.0458	0.0557	0.0900	0.0263
median	0.2044	0.1784	0.1894	0.2144	0.1700
max	0.2488	0.2375	0.2354	0.2658	0.2602

Table 3. The results for our runs and the overall results of the session track

- The 'essex3' system is among the top performing systems in the track for task 2 as its $nsDCG(RL13)$ score is close to the maximum score reported by NIST and is above the median.
- The second baseline 'essex2' failed to improve retrieval performance when using previous queries history.
- When looking at individual query pairs where 'essex3' succeeded in extracting query expansions from the anchor logs, the majority of these resulted in a better retrieval performance over 'essex1' for $RL3$. This is another evidence of the usefulness of the anchor text.

6 Conclusion

This year's session Track provided a platform to evaluate the effectiveness of Information Retrieval systems in interpreting query reformulations. The results of our runs are promising. First, they show that even a very simple way of using previous user interactions can improve retrieval performance for reformulated queries. Second, they provide an evidence that using anchor logs to derive query expansions for the reformulated query can improve performance over a baseline system.

Acknowledgements

This research is part of the AutoAdapt research project. AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1.

References

1. TREC 2010 Session Track Guidelines. `http://ir.cis.udel.edu/sessions/guidelines.html`, July 2010.
2. N. Craswell, D. Fetterly, M. Najork, S. Robertson, and E. Yilmaz. Microsoft research at trec 2009: Web and relevance feedback tracks. In *18th Text REtrieval Conference (TREC)*. NIST, 2009.
3. V. Dang and B. W. Croft. Query reformulation using anchor text. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 41–50, New York, NY, USA, 2010. ACM.

4. S. Dignum, U. Kruschwitz, M. Fasli, Y. Kim, D. Song, U. Cervino, and A. De Roeck. Incorporating Seasonality into Search Suggestions Derived from Intranet Query Logs. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence (WI'10)*, pages 425–430, Toronto, 2010.
5. N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 459–460, New York, NY, USA, 2003. ACM.
6. B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani. Using association rules to discover search engines related queries. In *Proceedings of the First Latin American Web Congress*, pages 66–71, 2003.
7. D. Hiemstra and C. Hauff. Mirex: Mapreduce information retrieval experiments. Technical Report TR-CTIT-10-15, Centre for Telematics and Information Technology University of Twente, Enschede, April 2010.
8. E. Kanoulas, P. Clough, B. Carterette, and M. Sanderson. Session Track at TREC 2010. In *Proceedings of the Workshop on the Automated Evaluation of Interactive Information Retrieval in conjunction with the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2010*, Geneva, Switzerland, 2010.
9. M. Koolen and J. Kamps. The importance of anchor text for ad hoc search revisited. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129, New York, NY, USA, 2010. ACM.