AutoEval: An Evaluation Methodology for Evaluating Query Suggestions Using Query Logs

M-Dyaa Albakour[1], Udo Kruschwitz[1], Nikolaos Nanas[2], Yunhyong Kim[3], Dawei Song[3], Maria Fasli[1], and Anne De Roeck[4]

[1]University of Essex, Colchester, UK
malbak@essex.ac.uk
[2]Centre for Research and Technology - Thessaly, Greece
[3]Robert Gordon University, Aberdeen, UK
[4]Open University, Milton Keynes, UK

Abstract. User evaluations of search engines are expensive and not easy to replicate. The problem is even more pronounced when assessing adaptive search systems, for example system-generated query modification suggestions that can be derived from past user interactions with a search engine. Automatically predicting the performance of different modification suggestion models *before* getting the users involved is therefore highly desirable. *AutoEval* is an evaluation methodology that assesses the quality of query modifications generated by a model using the query logs of past user interactions with the system. We present experimental results of applying this methodology to different adaptive algorithms which suggest that the predicted quality of different algorithms is in line with user assessments. This makes *AutoEval* a suitable evaluation framework for adaptive interactive search engines.

1 Introduction

Interactive search interfaces are becoming more popular in modern search engines. Google wonder wheel[1] and AquaBrowser[2] are examples of such interfaces which provide visualised query refinement suggestions to guide users in search and navigation in addition to providing a list of documents.

In order to provide suggestions for query modification, a domain model that reflects the domain characteristics could be used, e.g. a taxonomy or simply some term association graph. Several methods have been proposed in the literature to build such models. Some of these methods perform statistical and lexical analysis on the document contents to derive term relations, e.g. [10]. With the increasing availability of search logs obtained from user interactions with search engines, new methods have been developed for mining search logs to capture "collective intelligence" for providing query suggestions. This can be done, for example, by looking at the actual queries submitted and building query flow graphs [1], query-click graphs [2] or association rules [4].

[1] http://www.googlewonderwheel.com
[2] http://serialssolutions.com/aquabrowser/

The evaluation of these domain models in providing query recommendations remains a major challenge. The standard evaluation mechanism is to conduct user studies, e.g. [10], but such studies are expensive, not easy to reproduce and they involve a great deal of subjectivity. The automatic evaluation of search systems that does not rely on expensive user judgements has long been attracting IR researchers, e.g. [11]. This is however not an easy exercise and unlike commonly understood TREC measures (such as precision and recall), there is no commonly agreed automatic evaluation measure for *adaptive* search. One approach for automatic evaluation is using search logs. Joachims shows how clickthrough data can replace relevance judgements by experts or explicit user feedback to evaluate the quality of retrieval functions [5]. Zhang *et al.* have recently shown how test collections specific to a library domain can be derived from search logs [12].

In this paper we explore experimentally a new evaluation approach based on search logs. Search logs contain information of what users entered and clicked. It is a reflection of a reality and is representative to both its document collection and its search transactions. *AutoEval* is a methodology that performs simulated query recommendation experiments based on past log data to evaluate different models for generating query suggestions.

The rest of the paper is structured as follows. In Section 2 we give an overview of the *AutoEval* methodology. In Section 3 we describe the experiments we have run. Results are discussed Section 4.

2 The AutoEval Methodolgy

AutoEval is based on the idea that we can assess the quality of a domain model by comparing suggestions derived from the model to query modifications actually observed in the log files. The idea has been proposed recently [8], but no experimental justification has been provided as yet. With *AutoEval*, the model's evaluation is performed on arbitrary intervals, e.g. on a daily basis. For example, let us assume that during the current day, three query modifications have been submitted. For each query modification pair, the domain model is provided with the initial query and returns a ranked list of recommended query modifications. We take the rank of the actual modified query (i.e., the one in the log data) in this list, as an indication of the domain model's accuracy. The assumption here is that an accurate domain model should be able to propose the most appropriate query modification at the top of the list of recommended modifications. This is based on the observation that users are much more likely to click on the top results of a ranked list than to select something further down [6], and it seems reasonable to assume that such a preference is valid not just for ranked lists of search results but for lists of query modification suggestions as well. So for the total of three query modifications in the current day, we can calculate the model's Mean Reciprocal Rank (MRR) score as $(1/r_1 + 1/r_2 + 1/r_3)/3$, where r_1 to r_3 are the ranks of the actual query modifications in the list of modifications recommended by the model in each of the three cases. More generally, given a

day d with Q query modification pairs, the model's Mean Reciprocal Rank score for that day MRR_d is given by equation 1 below.

$$MRR_d = (\sum_{i=1}^{Q} \frac{1}{r_i})/Q \quad (1)$$

Note that in the special case where the actual query modification is not included in the list of recommended modifications then $1/r$ is set to zero. The above evaluation process results in a score for each logged day. So overall, the process produces a series of scores for each domain model being evaluated. These scores allow the comparison between different domain models. A model M_1 can therefore be considered superior over a model M_2 if a statistically significant improvement can be measured over the given period.

The described process fits perfectly a static model, but in the case of dynamic experiments as we are conducting here, the experimental process is similar. We start with an initially empty domain model, or an existing domain model. Like before, the model is evaluated at the end of each daily batch of query modifications, but unlike the static experiments it uses the daily data for updating its structure.

3 Experimental Setup

The aim of the experiment is to find out whether the performance predicted by *AutoEval* is in line with how users would judge the results. Here, we are not interested in the *absolute* values but instead we would like to know if the *relative* comparison between different systems can be replicated by user judgements. In other words, we would like to find out whether a query suggestion model deemed better by *AutoEval* is in fact producing "better" query suggestions when consulting real users.

We select two adaptive domain models which are continuously learning query modification suggestion from past queries as recorded in log files. In addition, we use an association rule-based approach that operates on the same log data. The three models can be summarized as follows:

- **ACO** uses an ant colony optimization (ACO) approach to learn a graph of related queries that can then be used to make query modification suggestions. The algorithm is described elsewhere [3]. Generally speaking, the model is used to provide suggestions for query modification by first finding the original query phrase in the graph, then listing all the associated nodes (query phrases) ranked by their associated weight.
- **Nootropia** is an immune inspired model for *adaptive information filtering* [9]. Here Nootropia is cast to the problem of continuously learning a domain model for query recommendations, by treating each query as a textual feature and each query session as a "bag" of textual features.

– **Fonseca** is an alternative to graph-based structures which derives query modification suggestions using association rules [4]. The idea is to use session boundaries and to treat each session as a transaction. Related queries are derived from queries submitted within the same transaction.

Following Fonseca's approach and to reduce noise, in all our experiments we only consider sessions where the number of queries is less than 10 and those which span over the period of less than 10 minutes. We use weekly batches to update the domain models.

The search log data in our experiments are obtained from the University of Essex website search engine. These logs have been collected since November 2007 (more than 1.5 million queries have been submitted so far). Each record in our query logs contains a time stamp of the transaction, the query that has been entered and the session identifier.

We first run *AutoEval* on the log data over the period of 64 weeks between the beginning of the academic year 2008 to the end of the autumn term in 2009 using the different models for suggesting query modifications. This gives us MRR scores for each system on weekly intervals. In order to validate our automatic evaluation methodology we performed a user-based assessment as proposed in the literature [10]. In this approach participants are given queries and their refinements and they are asked to determine whether these refinement suggestions are relevant to the original query. We sampled 20 queries from the entire log data. Apart from frequent queries (that make up a large proportion of all queries) we also sampled queries of medium frequency similar to [1]. We randomly selected 10 queries from the top 50 queries in the log data. Then we selected 10 queries within a range of medium frequency (between 50-1000), these do not overlap with the top 50 queries.

In order to select a sensible number of query modification suggestions, for each sampled query we selected the three best (highest weighted) related terms using five different models:

– **ACO1**: this is the ant colony optimisation model learnt over the entire 64 weeks period used in the *AutoEval* run.
– **ACO2**: this is the ant colony optimisation model learnt over a shorter period which is only the autumn term of the academic year 2008.
– **Fonseca**: this the domain model learnt using Fonseca's association rules over the entire 64 weeks period used in the *AutoEval* run.
– **Nootropia**: this the domain model learnt using Nootropia over the entire 64 weeks period used in the *AutoEval* run.
– **Baseline**: As a baseline we selected a method that does not rely on log data (and does not get updated in weekly batches). We assume that the top matching results of a commercial search engine will be a useful resource to derive query modification suggestions. We derived nouns and noun phrases from the top ten snippets returned by *Yahoo!* (restricting the search to the University of Essex website). We identify nouns and noun phrases using text processing methods applied in previous experiments [7].

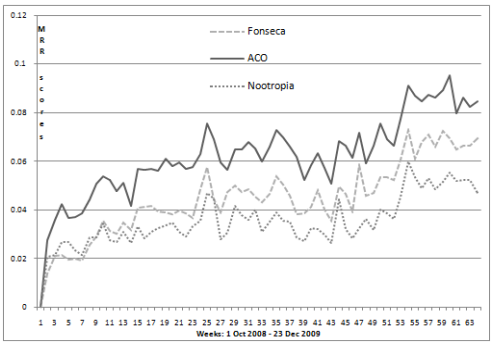

Fig. 1. AutoEval run for ACO, Nootropia, and Fonseca for a period of 64 weeks.

This has resulted in 214 distinct query pairs after removing duplicates due to the overlap of some of the suggestions coming from different systems. An online survey was prepared, and we asked 16 subjects (students and staff at Essex University) to fill in the survey. Participants were not told that various different techniques have been used to generate these query pairs. The form contained a list of all query pairs in random order. With each query pair the participants had to decide whether the refinement is relevant or not relevant. They were also given the choice to choose "do not know" if they were not sure.

4 Results and Discussion

Figure 1 illustrates the results of running *AutoEval*. We see that despite a few spikes the general trend is upwards indicating that different adaptive learning methods are able to learn from past log data over time. The figure suggests that the ACO method is significantly more effective than learning based on association rules and Nootropia.

The results of the user study are in line with this finding. The aggregated results are shown in Table 1. For each user we calculated the percentage of pairs that were judged relevant and then we aggregated the results among the assessors. ACO is the best performing system overall being significantly better than any of the alternatives ($p < 0.05$). The differences in the user assessment scores reflect the differences observed in Figure 1. The order of the three different adaptive approaches is consistent with the automatic evaluation. The user assessment also shows that ACO and Fonseca adaptive models are considered better by the users than the snippet baseline approach which is in line with our previous experiments [3]. Furthermore, ACO1 is significantly better than ACO2 ($p < 0.0001$), i.e. the increase in performance observed over time in the automatic evaluation is reflected in the user assessment. It also means the ACO adaptive model is capable of learning better query suggestions over time.

As a conclusion, *AutoEval* appears to be a sensible methodology capable of identifying performance improvement of an adaptive model for providing query

	ACO1	**ACO2**	**Fonseca**	**Nootropia**	**Baseline**
Relevant	59.38%	50.00%	55.63%	29.16%	54.06%

Table 1. Query suggestions judged 'relevant' by users.

suggestions over time. We show that this methodology can perform comparative experiments where different adaptive models can be tested under the same experimental conditions. For future work we propose to explore the ACO model with different settings, e.g. updating the weights using clickthrough data by giving more rewards for suggestions that lead to a landing page.

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. P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In: *Proceedings of WSCD'09*, pp. 56-63, Barcelona (2009)
2. N. Craswell and M. Szummer. Random Walks on the Click Graph. In: *Proceedings of SIGIR'07*, pp. 239-246, Amsterdam (2007)
3. 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 WI'10*, pp. 425-430, Toronto (2010)
4. 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*, pp. 66-71, Santiago (2003)
5. T. Joachims. Evaluating retrieval performance using clickthrough data. In: J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*, pp. 79-96. Physica/Springer Verlag (2003)
6. T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In: *Proceedings of SIGIR'05*, pp. 154-161, Salvador (2005)
7. U. Kruschwitz. *Intelligent Document Retrieval: Exploiting Markup Structure*, volume 17 of *The Information Retrieval Series*. Springer (2005)
8. N. Nanas, U. Kruschwitz, M.-D. Albakour, M. Fasli, D. Song, Y. Kim, U. Cervino, and A. De Roeck. A Methodology for Simulated Experiments in Interactive Search. In: *Proceedings of the SIGIR'10 SimInt Workshop*, pp. 23-24 Geneva (2010)
9. N. Nanas and A. Roeck. Autopoiesis, the immune system, and adaptive information filtering. *Natural Computing: an international journal*, 8(2), pp. 387-427 (2009)
10. M. Sanderson and B. Croft. Deriving concept hierarchies from text. In: *Proceedings of SIGIR'99*, pp. 206-213, Berkeley, CA (1999)
11. I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In: *Proceedings of SIGIR'01*, pp. 66-73, New Orleans (2001)
12. J. Zhang and J. Kamps. A search log-based approach to evaluation. In: *ECDL 2010*, pp. 248-260, Glasgow (2010)