Incorporating Seasonality into Search Suggestions Derived from Intranet Query Logs

Stephen Dignum*, Udo Kruschwitz*, Maria Fasli*, Yunhyong Kim†, Dawei Song†, Ulises Cerviño Beresi†, Anne de Roeck‡
*School of Computer Science and Electronic Engineering
University of Essex, United Kingdom
Email: udo@essex.ac.uk
†School of Computing
Robert Gordon University, Aberdeen, United Kingdom
‡Department of of Mathematics and Computing
The Open University, Milton Keynes, United Kingdom

***Abstract*—While much research has been performed on query logs collected for major Web search engines, query log analysis to enhance search on smaller and more focused collections has attracted less attention. Our hypothesis is that an intranet search engine can be enhanced by adapting the search system to real users' search behaviour through exploiting its query logs. In this work we describe how a constantly adapting domain model can be used to identify and capture changes in intranet users' search requirements over time.**

We employ an algorithm that dynamically builds a domain model from query modifications taken from an intranet query log and employs a decay measure, as used in Machine Learning and Optimisation methods, to promote more recent terms. This model is used to suggest query refinements and additions to users and to elevate seasonally relevant terms.

A user evaluation using models constructed from a substantial university intranet query log is provided. Statistical evidence demonstrates the system's ability to suggest seasonally relevant terms over three different academic trimesters. We conclude that log files of an intranet search engine are a rich resource to build adaptive domain models, and in our experiments these models significantly outperform sensible baselines.

***Keywords*-information retrieval; interactive search; intranet search; local Web search; adaptive domain models; ant colony optimisation**

I. Introduction

Domain modeling can generally be defined as the process of capturing and structuring knowledge embedded within information objects of interest to a selected domain (e.g. community). Domain models are realised in many ways, for example, as an organisation of documents into a classification schema, as a linked network of information objects, as a relational database, and, as a hierarchical or partially ordered graph comprising domain-relevant entities as nodes.

Although much work has concentrated on the building of domain models to aid user search in the form of suggested query modifications little has been attempted to find methods to modify those models over time to capture changing requirements and seasonal preferences.

A natural way to accomplish this is to analyse user query logs, i.e., to look at which phrases are used to interrogate a document collection and how those phrases are refined by the users. This paper uses this idea and builds a domain model based upon *Query Chains* where individual query phrases are represented as nodes, and edges are used to link phrases that belong to the same search mission [1]. If we analyse a query log over time and find a way to combine refinements from different user sessions, we have, in effect, produced a consolidated *Session Graph* [2]. Models that are able to capture *evolving* trends in search query graphs are only just starting to emerge, e.g. [3].

In this paper we use a Machine Learning method to learn weights for our consolidated edges so that suggestions can be ranked and the most promising ones supplied to the user.

Although we can build a *one-shot* model using such a method we wish to find a solution to adding new data, i.e., one that does not require the model to be completely rebuilt for the information it contains to become more current. To do this we employ a method analogous to Ant Colony Optimisation trail traversal and pheromone evaporation [4]. Briefly, each time a user enters a query modification or selects a suggested modification, the weight between it and the original query is increased, but all weights are reduced periodically so that paths that have not been recently traversed reduce in weight.

In the next section we will briefly discuss related work in this area, then in Section III we describe our method in greater detail. In Section IV, we provide details of our evaluation method looking at how quickly the model learns suggestions and how it performs over a number of different seasons, with comparison to a number of non-seasonal baselines. The results of our evaluation are described and discussed in Section V. Finally we draw conclusions and discuss future work in Sections VI and VII.

II. Related Work

Interactive information retrieval has received much attention in recent years, e.g. [5], [6]. The fact that all major Web search engines have now moved to more interactive features reflects the expectation of users to get some support in selecting search words for query formulation.

One possible step towards more interactivity is to improve query modification suggestions proposed by the search engine. It is recognized that there is great potential in mining information from query log files in order to improve a search engine [7], [8]. Given the reluctance of users to provide explicit feedback on the usefulness of results returned for a search query, the automatic extraction of implicit feedback has become the centre of attention of much research. We wish to build a model that captures user refinements and consolidates them to provide a dynamic model that will enable the combined knowledge to be examined e.g., a *learning network* in which algorithms build and extend network representations by acquiring knowledge from examples [9], in that we wish to capture user experience to update the model. One motivation could be that a large proportion of queries submitted to a search engine can be exact repeats of a query issued earlier by the same user [10]. However, our main motivation is to use the model to help make suggestions that can be used by other users.

There are many different ways of structuring such models. Models can be built by extracting term relations from documents or from the actual queries that users submit to search the collection. Past user queries appear to be preferred by users when compared to terms extracted from documents [11], which is one motivation for using log files in our work. Various Web log studies have been conducted in recent years to study the users' search behaviour, e.g. [12], [13], [14], [15], and log files have widely been used to extract meaningful knowledge, e.g. relations between queries [16], or to derive query substitutions [17]. Much of this work however is based on queries submitted on the *Web* and thus presents a very broad view of the world. Our work is different and novel in that we start with a specific document collection, or in other words a search domain (for which suitable knowledge structures are typically not readily available), extract relations from queries submitted within this domain to build and *evolve* the model automatically.

Of interest, and an inspiration for this paper is the Nootropia system [18] for user profiling. This determines hierarchies of terms and disseminates energy using a method based on Artificial Immune Systems. We, however, take a related, if conceptually opposite method, to provide a model based on a *consolidated user* as opposed to learning differences between individuals.

III. The Domain Model

Our domain model takes the form of a graph structure where nodes are query phrases and edges represent possible query refinements, higher weights denoting more common selections. Using such an internal representation allows numerous potential display and interrogation techniques to be presented to the user, these range from very simple tag clouds to more complex graph manipulations, see [19] for an example of the latter.

An algorithm, analogous to Ant Colony Optimisation methods (see Program 1), has been employed to populate the graph. The user traverses a portion of the graph by using query refinements (analogous to the ant's journey), the weights on this route are reinforced (increasing the level of pheromone). Over time all weights are reduced by a set proportion (pheromone evaporation). To reduce noise we only associate immediate refinements, e.g., for a session containing a query modification chain q_1 to q_4, associations will be created between q_1 and q_2, q_2 and q_3, and q_3 and q_4 only.

Program 1 Ant Colony Optimisation Framework

```
do
{
  determine ant starting nodes
  for each ant
  {
     carry out a traversal, leave
     pheromone between each node
  }
  evaporate (reduce) pheromone
} until stopping condition
```

The specifics are as follows:

At the end of each day all edge weights are normalised to sum to 1 and the mean weight of all edges is then calculated. For the next day, all queries in the log are extracted for that day where there are multiple queries in a particular user session. The queries are then time ordered and for each query phrase that follows an earlier phrase in the session an edge is created, or updated if it already exists, by the mean association weight of the previous day. A nominal update value of 1 is used for our first day, however, any positive real number could have been chosen without affecting the outcome of normalisation.

By normalising the weights at the end of each day we reduce the weight of non-traversed edges, hence, over time, penalising seasonally incorrect or less relevant phrase refinements. In addition we expect outdated terms to be effectively removed from the model, i.e., the refinement weight will become so low that the phrase will never be recommended to the user.

One would expect to use the model to provide suggested terms by first finding the original query phrase in the graph, then list the terms with higher weights. Although not addressed in this paper, indirect associations could also be used when data is sparse, or if we wish to investigate subtrees with relatively high weights.

Although we have chosen to run the update on a daily basis, update sessions could be run hourly or weekly, or even when a certain number of user sessions have completed. In addition, it is possible to run the algorithm from any point in the user log to any other, this allows us to compare how the model performs for particular time periods. We investigate this in the next section.

IV. Experimental Set-Up

Using a university query log, we chose to start the learning algorithm from 1st September 2008, nominally the start of academic activity for a new academic year, and to end at the official end-of-trimester dates for the Winter 2008, Summer 2009, and Winter 2009 seasons. We started with an initially empty domain model and then run the update as a batch process on a day by day basis.

To capture the most general queries, representative of the typical intranet user, from the same log we extracted the 20 most common phrases submitted to a university search engine[1] from 736,617 user submitted queries in total during the selected period. We ignored two frequent queries, an empty search and the query *'search'* which is the text displayed as default in the search box.

Investigating the most frequently submitted queries (instead of looking at a random sample) has a number of advantages. First of all, these queries represent a substantial proportion of all queries submitted to the search engine. In fact, the top 20 queries make up more than 15% of all queries in the entire query corpus. In other words, by being able to produce useful modification suggestions for these top queries we could address about every sixth user request submitted to the search engine. Furthermore, the results can be directly compared to a previous study which derived modification suggestions from the actual documents (and not the log files) [20]. We do however have to add, that samples from the long tail of rarely or uniquely submitted queries will be interesting to investigate and we leave that as one of the next steps in our future work.

The selected queries together with their frequency can be seen in Figure 1. Using these queries we selected the 3 best, i.e. highest weighted, refinements from each trimester's domain model (we ignored one refinement that suffered from sparse data in the December 2008 trimester). The reason for investigating top-ranked query suggestions is because users are much more likely to click on the top results of a ranked list than to select something further down [21].

To provide a baseline comparison we also extracted the 3 highest ranked refinements from two online search engines, Google[2] and the meta-search engine Clusty[3]. In each case we specified that the search engine only searches from the university web site using the associated *host/site* clause, e.g. on Google: *'library site:essex.ac.uk'*. For Google, we expanded the *show options* option after the first search and then selected the *related searches* option. For Clusty we used the cluster names from the left menu after each search. Suggestions from both systems were taken from top left to bottom right in the presented order. Baseline refinements were captured on 3rd February 2010. See Figure 2 for query-suggestion examples. After inspection, the suggestions from Google were seen, intuitively, to be more general whilst those from Clusty highly domain specific, therefore, we expected these two systems to offer two distinct baselines, the Clusty suggestions being much harder to beat.

Count	**Query**
27133	moodle
16382	library
11624	timetable
10879	search
5510	cmr
4913	enrol
4543	*(empty query)*
3745	accommodation
3740	ocs
3711	accomodation
3565	graduation
3492	psychology
3381	timetables
2969	term dates
2769	courses
2704	student union
2310	fees
2241	law
2238	sports centre
2203	registry
2097	exam timetable
2058	mba

Figure 1. Most frequent queries.

Clusty recommendations are actually just labels for clusters of retrieved documents. These labels are not presented as query modification suggestions, but they can easily be used as query modifications instead.

One question to be asked is whether our baseline suggestions are sensible, as users of the local search engine and the two Web search engines all have access to slightly different index databases and we do not know exactly how the underlying algorithms work. We argue that these baselines are sensible because they are based on existing state-of-the-art systems and because for the type of evaluation in which we ask users to assess whether suggestions are relevant we do not necessarily need to derive such suggestions from the

[1] `http://search.essex.ac.uk/`

[2] `http://www.google.com`

[3] `http://www.clusty.com`

Domain Model / Baseline	**First**	**Second**	**Third**
1, Winter ’08	dates	calendar	N/A
2, Summer ’09	graduation 2009	ceremony	graduation dates
3, Winter ’09	graduation 2009	ceremony	exam results
Baseline-1, Google	graduation gifts	graduation kayne west	graduation lyrics
Baseline-2, Clusty	photographs	department calendar of the university of essex	registration

Figure 2. Suggestions for the query ‘graduation’.

same database to be able to compare them. Furthermore, we will discuss our results with reference to other baselines reported in the literature.

To assess the quality of query modification suggestions we adopted an evaluation strategy proposed in the literature [22]. An online form was prepared, and participants were asked to determine whether queries and their refinements were relevant for the associated trimester. Some were expected to be relevant to all of the periods whilst others to be specific to the trimester in question. The pairs were ordered randomly and no indication was given regarding the source of the refinement or its order of suggestion. In total 539 decisions had to be made, 180 for each trimester, with one suggestion missing for a query concerning the earliest domain model due to sparsity of data (unsurprisingly for *’graduation’* which takes place at the end of the Summer trimester at the University).

The participants were deliberately selected from only staff and students of the university, i.e., the exact users for which the suggestions would be targeted in a live system. A total of 27 were recruited. This was made up of 3 academic staff, 9 non-academic staff and 15 students.

V. Results and Discussion

Figure 3 shows the survey results for our 27 participants. We see for each trimester, the percentage of first suggestions (i.e. highest ranked) that were judged to be relevant, the percentage of total relevant results, i.e., for each query and every suggestion, and finally the percentage for which the system in question had provided at least one suggestion that was judged relevant.

First, we can see that the domain model consistently outperforms all other models on all measures. The suggestions generated by the clustering search engine Clusty come much closer to the domain model than the Google suggestions in terms of performance. This could be explained by the fact that Google’s suggestions appear to be independent of the domain name provided by a query, e.g. *’graduation site:essex.ac.uk’* (see the example in Figure 2). Clusty, on the other hand, clusters the result set and then extracts terms to describe each cluster. In other words, Clusty’s results appear to be directly derived from the matching documents found in the specified domain unlike suggestions presented by Google. It needs to be pointed out though that we cannot simply assume that terms extracted from the result set are automatically better than terms extracted differently.

Paired t-tests reveal that the domain model suggestions were judged significantly better than all baselines for almost all comparisons ($p<0.01$). The only exception to this is when we consider the first suggestion only, then we observe that the original model (Winter’08) is better than the Clusty baseline but not significantly ($p=0.215$). The updated model (Summer’09) however does become significantly better than the baseline ($p=0.019$), the ultimate model (Winter’09) is better at $p<0.01$.

The results indicate that our domain model is a *quick learner* and will not require relatively long periods of learning examples to be a useful query suggestion aid.

Looking at the First Relevant figures we can see a gradual, though not statistically significant improvement in performance for the domain models suggesting that the very best suggestions will be promoted. However, there is a falling off of performance for Total Relevant suggestions by the Winter 2009 trimester which indicates much more variation in the quality of second and third suggestions. One should not forget, however, that both measures are still significantly better than the associated baselines.

Knowing that the query suggestions we derived from log data are more relevant than those presented by some standard search engines is an interesting finding on its own but it also raises some questions. First of all, what algorithms are we comparing our approach against? More importantly, how does our algorithm score against alternative approaches such as suggestions derived from snippets or from the entire document collection?

The first question is indeed difficult to answer but as we pointed out previously, our main concern was to find out how our adaptive domain model construction process would compare against suggestions derived from state-of-the-art approaches (no matter how these actually work in detail). We are however in a position to compare our results against a snippet-based baseline as well as a method that extracts query suggestions from the entire document collection. A previous study assessed query modifications for the most frequent queries submitted to an earlier search engine within the same domain [20]. Term suggestions were extracted

Domain Model / Baseline	**% First Relevant**	**% Total Relevant**	**% At Least One Relevant**
1, Winter '08	60.37%	63.78%	87.40%
2, Summer '09	62.22%	66.54%	85.56%
3, Winter '09	63.15%	60.74%	82.04%
Baseline-1, Google	41.67%	35.53%	62.96%
Baseline-2, Clusty	55.74%	45.99%	74.81%

Figure 3. A comparison of user judged relevant query refinements for three university trimesters.

from a static domain model that had been automatically constructed from the entire document collection utilizing the documents' markup structure [23]. Only the first relevant suggestion was assessed. This resulted in 59% relevant suggestions. The baseline approach which selected terms using the snippets of the best matching documents resulted in 50% of the suggestions being relevant on average. We observe an improvement of our adaptive model over two very different alternative techniques, one that uses the returned snippets as a source of potential query modification terms and one that uses the complete document collection to acquire a static doman model.

To put the results in context with other log-based approaches, Boldi *et al.* [1] used query sequences submitted to a Web search engine to build a query-flow graph, an aggregated representation of the latent querying behaviour, and their best methods produced 58% of suggestions that are either "useful" or "somewhat useful". Unlike in our study, assessments were performed for five rather than three recommendations per query.

We should however add that these comparisons with previous studies not working on the exact same data set and context should only be taken as indicative as there are many parameters that may affect the results.

VI. Conclusions

In this paper we have described how to build a domain model that can learn associations, over time, through the analysis of a user query log. Association weights are learnt through the traversal of queries within sessions by the users increasing weights where traversal has taken place but also by the reduction of the weights of all associations at the end of a particular learning period. This method is analogous to the alteration of pheromone levels used within Ant Colony Optimisation techniques. The outlined approach is novel in that it presented a method to build truly *adaptive* domain models.

The model that we built using a log file of a university intranet search engine has been shown to outperform a number of baselines in the suggestion of relevant query modifications. This is true from the very first learning period where only four months of learning examples have been provided. Gradual improvement of First Relevant term suggestions is reported indicating that the system has the potential to improve performance over time as well as keeping pace with seasonal changes in user interests. We have also demonstrated that a domain model acquired from log data has the potential to outperform approaches that are simply based on knowledge extracted from the actual documents (therefore strengthening observations made for general *Web* search [11]).

The learning process has been kept deliberately as simple as possible and we avoided any domain-specific customization, however, in light of the research a number of alterations may be beneficial. This is discussed in the next section.

VII. Future Work

We intend to continue the ACO analogy and add the idea of *distance* often used to temper the effect of a pure pheromone trail. We can do this using a document collection of commonly related terms or from a pre-search, i.e., looking at the terms in a top number of returned documents [19]. This can easily be combined with the actual displayed suggestions i.e., not to affect the underlying model directly.

On analysis of use of our pheromone / weight parameter, we can see that the added weight will diminish as the number of nodes in the underlying model grows. This provides, in effect, an inertia where it becomes harder for newer suggestions to overtake older refinements. As suggested in the Nootropia system [24] we may wish to periodically remove associations which fall below certain weight thresholds to counteract this.

Using an edge weighting method lends itself to analysing a number of refinements. We could, for example, use the later terms in a session or dialogue as implicit feedback (such later queries are often more successful [25]) and assign a higher weighting to that term. One would assume that the very final phrase was successful in locating a document in the collection and be a more useful refinement. Another alternative is to analyse the document collection itself applying higher weights to refinements that yield better discriminators.

Finally, we are investigating alternative learning methods to compare them with the approach described in this paper.

Acknowledgements

The research carried out for this paper was conducted within the framework of the project, Automatic Adaptation of Knowledge Structures for Assisted Information Seeking

(AutoAdapt), funded by the EPSRC grants EP/F035357/1 and EP/F035705/1. Nikolaos Nanas of LISyS CERETETH (Volos, Greece) helped to analyse the algorithm and made a number of suggestions regarding learning rates and inertia. We would also like to thank the three anonymous reviewers for detailed feedback.

References

[1] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna, "Query suggestions using query-flow graphs," in *Proceedings of the 2009 workshop on Web Search Click Data (WSCD'09)*, 2009, pp. 56–63.

[2] R. Baeza-Yates, "Graphs from search engine queries," in *33rd Conference on Current Trends in Theory and Practice of Computer Science*. Berlin, Germany: Springer, 2007, pp. 1–8.

[3] R. Baraglia, C. Castillo, D. Donato, F. M. Nardini, R. Perego, and F. Silvestri, "The Effects of Time on Query Flow Graph-based Models for Query Suggestion," in *Proceedings of RIAO'2010*, Paris, 2010.

[4] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant colony algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 3, pp. 137–172, 1999.

[5] I. Ruthven, "Interactive information retrieval," *Annual Review of Information Science and Technology (ARIST)*, vol. 42, pp. 43–92, 2008.

[6] G. Marchionini, "Human-information interaction research and development," *Library and Information Science Research*, vol. 30, no. 3, pp. 165–174, 2008.

[7] J. Jansen, A. Spink, and I. Taksa, Eds., *Handbook of Research on Web Log Analysis*. IGI, 2008.

[8] F. Silvestri, *Mining Query Logs: Turning Search Usage Data into Knowledge*, ser. Foundations and Trends in Information Retrieval. Now Publisher, 2010, vol. 4.

[9] J. F. Sowa, "Semantic networks," in *Encyclopedia of Artificial Intelligence*, S. C. Shapiro, Ed. New York, NY, USA: John Wiley & Sons, 1992, pp. 1493–1511.

[10] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information Re-Retrieval: Repeat Queries in Yahoo's Logs," in *Proceedings of the 30^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, 2007, pp. 151–158.

[11] D. Kelly, K. Gyllstrom, and E. W. Bailey, "A comparison of query and term suggestion features for interactive searching," in *Proceedings of the 32^{nd} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, 2009, pp. 371–378.

[12] P. Anick, "Using Terminological Feedback for Web Search Refinement - A Log-based Study," in *Proceedings of the 26^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada, 2003, pp. 88–95.

[13] P. Wang, M. W. Berry, and Y. Yang, "Mining Longitudinal Web Queries: Trends and Patterns," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 54, no. 8, pp. 743–758, June 2003.

[14] M. Chau, X. Fang, and O. R. L. Sheng, "Analysis of the Query Logs of a Web Site Search Engine," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 56, no. 13, pp. 1363–1376, November 2005.

[15] B. J. Jansen, A. Spink, and S. Koshman, "Web Server Interaction with the Dogpile.com Metasearch Engine," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 58, no. 5, pp. 744–755, March 2007.

[16] R. Baeza-Yates and A. Tiberi, "Extracting semantic relations from query logs," in *Proceeding of the 13^{th} ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, San Jose, California, 2007, pp. 76–85.

[17] R. Jones, B. Rey, O. Madani, and W. Greiner, "Generating Query Substitutions," in *Proceedings of the 15th International World Wide Web Conference (WWW'06)*, Edinburgh, 2006, pp. 387–396.

[18] N. Nanas and A. Roeck, "Autopoiesis, the immune system, and adaptive information filtering," *Natural Computing: an international journal*, vol. 8, no. 2, pp. 387–427, 2009.

[19] S. Dignum, Y. Kim, U. Kruschwitz, D. Song, M. Fasli, and A. De Roeck, "Using Domain Models for Context-Rich User Logging," in *Proceedings of the SIGIR 2009 Workshop on Understanding the User*, Boston, 2009.

[20] U. Kruschwitz, "An Adaptable Search System for Collections of Partially Structured Documents," *IEEE Intelligent Systems*, vol. 18, no. 4, pp. 44–52, July/August 2003.

[21] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proceedings of the 28^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 2005, pp. 154–161.

[22] M. Sanderson and B. Croft, "Deriving concept hierarchies from text," in *Proceedings of the 22^{nd} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999, pp. 206–213.

[23] U. Kruschwitz, *Intelligent Document Retrieval: Exploiting Markup Structure*, ser. The Information Retrieval Series. Springer, 2005, vol. 17.

[24] N. Nanas, V. Uren, and A. De Roeck, "A review of evolutionary and immune inspired information filtering," *Artificial Intelligence Review*, 2009.

[25] F. Radlinski and T. Joachims, "Query chains: learning to rank from implicit feedback," in *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM Press, 2005, pp. 239–248.