Efficient Diversification of Search Results using Query Logs

Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, Fabrizio Silvestri
ISTI – CNR, Pisa, Italy
{name.surname}@isti.cnr.it

ABSTRACT

We study the problem of diversifying search results by exploiting the knowledge mined from query logs. Our proposal exploits the presence of different "specializations" of queries in query logs to detect the submission of ambiguous/faceted queries, and manage them by diversifying the search results returned in order to cover the different possible interpretations of the query. We present an original formulation of the results diversification problem in terms of an objective function to be maximized that admits the finding of an optimal solution in linear time.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation, Search process, Selection process*

General Terms

Algorithms, Design, Experimentation, Performance.

Keywords

Query log analysis, search results diversification.

1. INTRODUCTION

Nowadays, users interact with Web Search Engines (WSE) by typing a few keywords representing their information need, and these keywords are often ambiguous and have more than one possible interpretation [3]. It is also known that WSEs collect detailed information about the queries submitted in the past along with a lot of additional information that are extremely valuable for a number of different tasks [7]. Indeed, through query log analysis we are introducing the following novel contributions to result diversification: (*i*) a methodology to detect the ambiguous queries that would benefit from diversification based on query log analysis; (*ii*) a methodology to efficiently and effectively devise the possible topics to include in the diversified list of results along with their probability distribution; (*iii*) a linear time diversification algorithm that re-ranks the results list on the basis of the set of query refinements mined from the log along with the associated probabilities; (*iv*) an objective diversification usefulness-measure to assess how valuable a diversified results list is.

Copyright is held by the author/owner(s).
WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

2. QUERY LOG BASED METHOD

We assume that a query log Q is composed by a set of records $\langle q_i, u_i, t_i, V_i, G_i \rangle$ registering, for each submitted query q_i: (*i*) the anonymized user u_i; (*ii*) the timestamp t_i at which u_i issued q_i; (*iii*) the set V_i of URLs of documents returned as top-k results of the query, and, (*iv*), the set C_i of URLs corresponding to results clicked by u_i.

Users generally query a WSE by submitting a sequence of requests. Splitting the chronologically ordered sequence of queries submitted by a given user into *sessions*, is a challenging research topic. Since session splitting methodologies are out of the scope of this paper, we resort to adopt a state-of-the-art technique proposed in [4].

Let q and q' be two queries submitted by the same user during the same logical session recorded in Q. We adopt the terminology proposed in [4], and we say that a query q' is a "specialization" of q if the user information need is stated more precisely in q' than in q. Let us call S_q the set of specializations of an ambiguous/faceted query q mined from the query log. Given the popularity function that computes the frequency of a query topic in Q, and a query recommendation algorithm trained with query log Q, any algorithm that exploits the query log sessions to provide users with suggestions of related queries, can be easily adapted for devising specializations of submitted queries.

Now, let us give some additional assumptions and notations. $\mathcal{D}$ is the collection of documents indexed by the WSE which returns, for any given query q, an ordered list of documents $R_q \subseteq \mathcal{D}$. The rank of document $d \in \mathcal{D}$ within R_q is indicated with $rank(d, R_q)$. The distance function $\delta : \mathcal{D} \times \mathcal{D} \rightarrow [0,1]$, having non-negative and symmetric properties is defined as $\delta(d_1, d_2) = 1 - cosine(d_1, d_2)$, where $cosine(\,)$ denotes the cosine similarity function.

The utility function specified defined in Equation (1) denotes how good $d \in R_q$ is for satisfying a user intent that is better represented by specialization q'.

$$U(d|R_{q'}) = \sum_{d' \in R_{q'}} \frac{1 - \delta(d, d')}{rank(d', R_{q'})}. \quad (1)$$

The intuition for U is that a result $d \in R_q$ is more useful for specialization q' if it is very similar to a highly ranked item contained in the results list $R_{q'}$.

Using the above definitions of distance (δ) and utility (U), we are able to define three different query-logs-based approaches to diversification: two are adaptations of the Agrawal *et al.* [1] algorithm and the Santos's xQuAD frame-

work [6]; the last one, named MaxUtility(k), refers to our novel formulation and it is defined as follows:

MaxUtility(k): *Given: query q, the set R_q of results for q, two probability distributions $P(d|q)$ and $P(q'|q)\,\forall q' \in S_q$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of having q' as a specialization of q, the utilities $U(d|R_{q'})$ of documents, a mixing parameter $\lambda \in [0, 1]$, and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes*

$$U(S|q) = \sum_{d \in S} \sum_{q' \in S_q} (1-\lambda) P(d|q) + \lambda P\left(q'|q\right) U\left(d|R_{q'}\right)$$

with the constraints that every specialization is covered proportionally to its probability. Formally, let $R_q \bowtie q' = \{d \in R_q | U(d|R_{q'}) > 0\}$. We require that for each $q' \in S_q$, $|R_q \bowtie q'| \geq \lfloor k \cdot P(q'|q) \rfloor$.

Our technique aims at selecting from R_q, the k results that maximize the overall utility of the results list. When $|S_q| \leq k$ the results are in someway split into $|S_q|$ subsets each one covering a distinct specialization. The more popular a specialization is, the greater the number of relevant for it results is.

Contrary to [1], MaxUtility(k) aims to maximize directly the overall utility. This simple relaxation allows the problem to be simplified and solved in a very simple and efficient way. Another important difference is that Agrawal's method needs to select, in advance, the subset S of documents before computing the final score. Therefore, to maximize $U(S|q)$ we simply resort to compute for each $d \in R_q$ the utility of d for specializations $q' \in S_q$ and, then, to *select* the top-k highest ranked documents. Obviously, we have to carefully select results to be included in the final list in order to avoid choosing results that are relevant only for a single specialization. To select results, we use a set of $|S_q|$ min-heaps each of those keeps the top $\lfloor k \cdot P(q'|q) \rfloor + 1$ useful documents for that specialization. Algorithm 1 returns the set S maximizing the objective function of MaxUtility(k) in linear time. Moreover, the running time of the algorithm is linear in the size of document considered. Indeed, all the heap operations are carried out on data structures having a constant size bounded by k.

Algorithm 1 OptSelect$(q, R_q, k) \rightarrow S$

```
1. S ← ∅, q' ∈ S_q, M ← new Heap(), ∀q'. M_q' ← new Heap();
2. For Each d ∈ R_q Do
3.    If U(d|R_q') > 0 Then M_q'.push(d) Else M.push(d);
4. While |S| < k Do
5.    If ∃q' ∈ S_q s.t. M_q' ≠ ∅ Then x ← M.pop();
6.    Else x ← pop d with the max U(d|R_q') from {M_q'}_∀q';
7.    S ← S ∪ {x};
```

3. EXPERIMENTS

We conducted our experiments in the context of the diversity task of the TREC 2009 Web track. The goal of this task is to produce a ranking of documents for a given query that maximizes the coverage of the possible aspects underlying this query, while reducing its overall redundancy with respect to the covered aspects. Two query logs, i.e. AOL and MSN, were preprocessed in order to devise the logical user sessions as described in Section 2. The sessions obtained were used to build the model for the recommendation algorithm described in [5]. We used ClueWeb-B, the subset of the TREC ClueWeb09 dataset collection used in the TREC 2009 Web track's Diversity Task, comprising a total of 50 million English Web documents. A total of 50 topics were available for this task. In our experiments, the query associated to each topic was used as initial ambiguous/faceted query. We evaluate the effectiveness of our method in diversifying the results retrieved using the DPH Divergence From Randomness model [2].

OptSelect and xQuAD [6] behave similarly in terms of effectiveness, while IASelect [1] performs always worse. In our tests xQuAD performs better than reported in its original version [6]. Essentially, this behavior could be explained by the following two reasons: (i) our method for measuring the "diversity" of a document based on Equation (1) is superior to the one used in [6]; (ii) our method for deriving specializations and their associated probabilities is able to carry out more accurate results.

k	$\lvert R_q \rvert$	OptSelect	xQuAD	IASelect
1,000	100,000	13.92	2,849.81	4,071.87

Table 1: OptSelect, xQuAD, and IASelect execution time (in msec).

Table 1 reports the average time required by the three algorithms to diversify the initial set of documents for the 50 queries of the TREC 2009 Web Track's Diversity Task. In terms of efficiency, the average time needed by OptSelect is noticeably less than the time required by its competitors. In particular, OptSelect is two orders of magnitude faster than its competitors. Tests were conducted on a Intel Core 2 PC with 8 GB of RAM and Ubuntu 9.10 (kernel 2.6.31-22).

4. ACKNOWLEDGEMENTS

This research has been funded by the EU CIP PSP-BPN ASSETS Project. Grant Agreement no. 250527.

5. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. ACM WSDM'09.*

[2] G. Amati, E. Ambrosi, M. Bianchi, and C. Gaibisso. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Proc. TREC*, 2007.

[3] A. Anagnostopoulos, A. Z. Broder, and D. Carmel. Sampling search-engine results. In *WWW'05, ACM.*

[4] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. From 'dango' to 'cakes': Query reformulation models and patterns. In *Proc. WI'09.* IEEE CS Press, 2009.

[5] D. Broccolo, L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri. An efficient algorithm to generate search shortcuts. Technical Report 2010-TR-017, ISTI CNR Pisa.

[6] R. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proc. WWW'10.* ACM Press, 2010.

[7] F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 1(1-2):1–174, 2010.