Improving Search Relevance for Implicitly Temporal Queries

Donald Metzler[†], Rosie Jones[‡], Fuchun Peng[†], and Ruiqiang Zhang[†]

[†] Yahoo! Labs, 2821 Mission College Blvd., Santa Clara, CA 95054
[‡] Yahoo! Labs, 4 Cambridge Center, Cambridge, MA 02142
{ metzler, jonesr, fuchun, ruiqiang }@yahoo-inc.com

Categories and Subject Descriptors: H.3.3 Information Storage and Retrieval: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Keywords: temporal queries, query log analysis, web search

1. INTRODUCTION

Many web search queries have implicit intents associated with them that, if detected and used effectively, can be used to improve search quality. For example, a user who enters the query “toyota camry” may wish to find the official web page for the car, reviews about the car, or the location of the closest Toyota dealership. However, since the user only entered a couple of keywords, it may be difficult to accurately determine which of these implicit intents the user actually meant. Given such an ambiguous query, a search engine must use personalization, click information, query log analysis, and other means for determining implicit intent.

Rather than solving the general problem of automatically determining user intent, we focus on queries that have a temporally dependent intent. Temporally dependent queries are queries for which the best search results change with time. Simple examples include “new years” and “presidential elections”, which are events that recur over time. The search results for these queries should reflect the freshest, most current results. A slightly more complex example is the query “turkey”. For this query, it may be useful to return turkey recipes or cooking instructions around the Thanksgiving holiday and travel information during peak vacation times. In all of our examples thus far, the events have occurred with (mostly) predictable periodicity. However, for queries such as “oldest person alive”, the best result changes unpredictably, making it difficult for search engines to consistently return correct results. Therefore, temporally dependent queries come in many different forms and pose many challenges to search engines.

In this paper, we investigate a subset of temporal queries that we call *implicitly year qualified* queries. A year qualified query is a query that contains a year. An implicitly year qualified query is a query that does not actually contain a year, but yet the user may have implicitly formulated the query with a specific year in mind. An example implicitly year qualified query is “miss universe”. It is plausible that the user actually meant “miss universe 2008”, “miss universe 2007”, or maybe even “miss universe 1990”, yet did not actually qualify the query with a year. Other examples include “olympics”, “toyota camry”, “easter”, and the names of conferences, such as “SIGMOD” or “KDD”.

Implicitly year qualified queries are particularly interesting and challenging from a search point of view. Our analysis shows that more than *7%* of queries belong to this category. However, correctly handling this type of query is not trivial. As anecdotal evidence, as of this writing, only one of the three major search engines ranks the SIGIR 2009 web page higher than the SIGIR 2008 web page for the query “SIGIR”, despite the fact that most people searching for the SIGIR conference are interested in the upcoming event, rather than the previous event. Therefore, in order to improve search quality for temporal queries, a search engine must be able to detect that certain queries have an implicit temporal intent and use this information to improve the search results. These are precisely the types of issues that we tackle in the remainder of this paper.

Our work is different from previous work on temporal queries (e.g., [1, 2, 3, 4]), because we mine temporal patterns directly from query logs and do not make use of query or document timestamps.

2. MINING ALGORITHM

We propose a simple, yet efficient algorithm for mining implicitly year qualified queries. The algorithm relies only on having access to a query log with frequency information. The approach does not rely on user, click, or session data.

The primary idea behind our proposed approach is to see how often a base query, such as “olympics” is year qualified in the query log. If the base query is commonly year qualified, then we declare that it is implicitly year qualified. For the olympics case, it is very likely that the queries “olympics 2008”, “2004 olympics”, etc. occur with relatively high frequency in the log, which allows us to determine that “olympics” is implicitly year qualified. Thus, the foundations of our algorithm are built upon the following assumptions: 1) implicitly year qualified queries are strongly associated with several different years, and 2) implicitly year qualified queries are associated with years more than they are associated with non-years. We now describe how these two assumptions can be turned into an effective mining algorithm.

Perhaps one of the most important properties of implicitly year qualified queries is how strongly they are associated with a given year. This value, which we call the year qualified weight, is formally defined as:

$$w(q, y) = \#(q.y) + \#(y.q) \tag{1}$$

SIGIR’09, July 19–23, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-483-6/09/07.

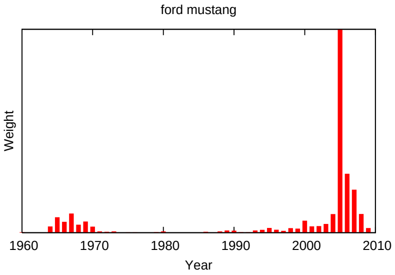

Figure 1: Implicit year distribution mined for the query “ford mustang”.

where $\#(q.y)$ denotes the number of times that the base query q is post-qualified with the year y in the query log. Similarly, $\#(y.q)$ is the number of times that q is pre-qualified with the year y. Figure 1 shows the weight distribution for the query “ford mustang”, a car model that was once popular in the 1960s and now again popular in recent years.

Given these weights, we use the following methodology to automatically mine implicitly year qualified queries. Given a query q, we first compute $w(q, y)$ for all plausible years. We then use the following function to determine if a query is implicitly year qualified:

$$isIYQQ(q) = \begin{cases} 1 & |\{y : w(q, y) > 0\}| \geq 2 \\ 0 & otherwise \end{cases}$$

which simply states that a query is implicitly year qualified if it is qualified by at least two unique years. We set the threshold at 2 because we are interested in temporally recurring events. It is possible to mine one-time events by lowering the threshold to 1, but this may result in spurious detections.

Even though a query is identified as implicitly year qualified does not necessarily mean that the query should always be treated as temporal in nature. Consider the query “chi”, which happens to be the name of a popular human-computer interaction conference. This query is clearly temporal, and indeed, our algorithm detects the query as being implicitly year qualified. However, “chi” is a very common term that is often qualified in many different ways, including “chi squared” (statistical test), “chi chis” (restaurant), and “chi omega” (sorority). In fact, these other qualifications are much more common than the temporal ones. We call this phenomenon *temporal ambiguity*. We quantify temporal ambiguity as follows:

$$\alpha(q) = \frac{\sum_y w(q, y)}{\sum_x \#(q.x) + \sum_x \#(x.q)} \qquad (2)$$

where the sums $\sum_x \#(q.x)$ and $\sum_x \#(x.q)$ go over all pre- and post-qualifications for the query q. It should be easy to see that if the query is always qualified with a year then $\alpha(q) = 1$. Although we call this measure temporal ambiguity, it may also be interpreted as a confidence value that the query has implicit temporal intent.

Ambiguity (α)	Baseline	Temporal Reordering
0.00 - 0.25	10.09	10.06 (-0.31%)
0.25 - 0.50	8.48	8.46 (-0.15%)
0.50 - 0.75	7.67	7.69 (+0.32%)
0.75 - 1.00	7.02	7.07 (+0.84%)†

Table 1: Baseline and temporal reordering results. A † represents a statistically significant ($p < 0.05$) difference versus the baseline.

3. EVALUATION

We can use information mined to improve search relevance for implicitly temporal queries. We propose explicitly adjusting the score of document d in response to query q according to the years found in the document. Given an implicitly year qualified query q, we first weight the qualified years associated with q as follows:

$$z(q, y) = \mathcal{N}(y; \mu, \sigma^2) \cdot \alpha(q) \cdot \frac{w(q, y)}{\max_y w(q, y)}$$

where $\mathcal{N}(y; \mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, $\alpha(q)$ is the temporal ambiguity (Equation 2), and $w(q, y)$ is the number of times that query q is qualified with year y (Equation 1).

Given a ranking function $S(q, d)$ that produces a score for documents d in response to query q, we can use the $z(q, y)$ weights to temporally bias $S(q, d)$ towards q's implicit temporal intent as follows:

$$S'(q, d) = S(q, d) + \sum_{f \in d} \lambda_f \sum_{y \in f} z(q, y)$$

where $S'(q, d)$ is the temporally biased score, $S(q, d)$ is the original score of document d with respect to query q, $f \in d$ is the set of document fields (e.g., title, anchor text, body), λ_f is the weight associated with field f, and $y \in f$ denotes the set of years that occur in field f in document d. Documents are then reordered according to $S'(q, d)$ to produce the final temporally-biased ranking. In our experiments, $\mu = 2008$, $\sigma^2 = 1$, and the title, anchor text, body, and url weight are assigned weights (λ_f) of 2.0, 2.0, 0.5, and 0.5, respectively.

Our evaluation involves a set of 670 web queries. We compare our temporal reordering approach to the ranking of a commercial search engine. Editors judged the relevance of each document retrieved for each query as either Perfect, Excellent, Good, Fair, or Bad.

The results are shown in Table 1. The first thing to notice is that DCG@5 decreases monotonically as the queries become less temporally ambiguous. This suggests that web search engines tend to perform substantially worse on temporal queries, highlighting the need for special temporal query handling. The results also show that our proposed temporal result set reordering tends to improve temporally unambiguous queries more than temporally ambiguous queries.

4. REFERENCES

[1] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *WWW*, pages 2–11, 2005.

[2] W. Dakka, L. Gravano, and P. Ipeirotis. Answering general time-sensitive queries. In *Proc. 17th Intl. Conf. on Information and Knowledge Management*, 2008.

[3] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 18–24, New York, NY, USA, 2004. ACM.

[4] X. Li and W. B. Croft. Time-based language models. In *Proc. 12th Intl. Conf. on Information and Knowledge Management*, pages 469–475. ACM, 2003.