Exploring Real-Time Temporal Query Auto-Completion

Stewart Whiting, James McMinn and Joemon M. Jose
School of Computing Science
University of Glasgow
Scotland, UK.
{stewh,mcminn,jj}@dcs.gla.ac.uk

ABSTRACT

Query auto-completion (QAC) is a common interactive feature for assisting users during query formulation. Following each query input keystroke, QAC suggests queries prefixed by the input characters; allowing the user to avoid further cognitive and physical effort if any are acceptable. To rank suggestions, QAC approaches typically aggregate past query popularity to determine the likelihood of a query being used again. Hence, QAC is usually very effective for consistently popular queries. However, as the web becomes increasingly real-time, more people are turning to search engines to find out about unpredictable emerging and ongoing events and phenomena. QAC approaches reliant on aggregating long-term historic query-logs are not sensitive to very recent real-time events, because newly popular queries will be outweighed by long-term popular queries, especially for less-specific prefix lengths (e.g. 2 or 3 characters). We explore limiting the aggregation period of past query-log evidence to increase the temporal sensitivity of QAC. We vary the query-log aggregation period between 2 and 14 days, for prefix lengths of 2 to 5 characters. Experimentation simulates a real-time environment using openly available MSN and AOL query-log datasets. Analysis indicates a linear relationship between prefix length and QAC performance when using different query-log aggregation periods. In particular, we find QAC for shorter prefix lengths is optimal when a shorter query-log aggregation period is used, and vice-versa, longer prefix lengths benefit from a longer query-log aggregation period.

1. INTRODUCTION

For users, cognitively formulating and physically typing queries is a time-consuming and error prone process. As such, query auto-completion (QAC) [3, 10] has been widely adopted by major web search engines to reduce the effort necessary to submit a query.

As a user types their query into the search box, QAC attempts to predict the completed query the user may have in mind. Following each query input keystroke, QAC suggests possible queries (which we refer to as *completion suggestions*) beginning with the already input character sequence (i.e. *prefix*). The goal for effective QAC is to present the user's intended query after the least possible keystrokes, and at the highest rank in the list of completion suggestions.

Conventional QAC approaches rank completion suggestions by aggregating their popularity in past query-logs. Further work has incorporated personal contextual features for short prefixes [3] and time-series modelling of temporal trends [10]. However, with enough past evidence, completion suggestions ranked solely by their popularity in past query-logs provides reasonably effective QAC [3, 10].

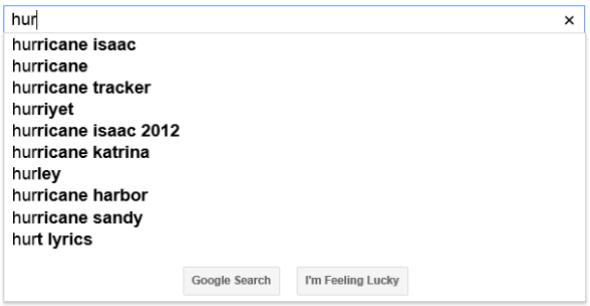

Figure 1: Google auto-completion suggestions for the query prefix '***hur***'. Screenshot taken November 8th 2012, 10 days after Hurricane Sandy made landfall on the East Coast of the USA. Browser cookies were cleared to avoid individual personalization effects.

Figure 1 illustrates the ten completion suggestions offered by Google for the three character query prefix '***hur***' on November 8th, 2012. The list of query suggestions indicates the historically most likely queries to be submitted with the given prefix, possibly in the context of some undisclosed ranking features such as geo-location or the user's past queries. Despite the recency and prominence of Hurricane Sandy, the query ranks very low in the completion suggestions, while '*hurricane isaac*' ranks first, regardless that it occurred many months previously. Aside from this issue, QAC for short and unspecific prefixes (i.e. 1 or 2 common characters) is often unsuccessful as there are usually a huge number of possible completion suggestions [3]. Consequently, it is typically long-term '*head*' queries that are suggested as completions for such short prefixes.

With the web increasingly becoming a platform for real-time news and media, time plays a central role in information interaction. A substantial volume of daily top queries are the result of users turning to search engines for up-to-date information about very recent or ongoing events [2, 6]. 15% of the daily queries to an industrial web search engine have never been seen before[1]; a substantial proportion of these queries may be attributed to real-time events and phenomena, rather than the long-tail of very uncommon queries. Similarly, previously unpopular queries may suddenly become extremely popular because of recent developments. It is therefore important that QAC supports queries which become highly popular only during brief periods of time, which we refer to as *real-time temporal queries*.

DIR 2013, April 26, 2013, Delft, The Netherlands.

[1] http://www.google.com/competition/howgooglesearchworks.html

Although the common approach to QAC is to rank completion suggestions by their popularity in the historic query-log (i.e. *past query-log evidence*), there has been very little study on the aggregation period necessary to achieve optimal QAC effectiveness, and whether this varies for each prefix length [10]. Thus, the objective of this paper is to investigate this uncertainty by conducting experiments based upon the AOL and MSN query-log datasets. For each prefix length we use an N day sliding window of past query-log evidence to rank completion suggestions, hence making QAC more sensitive to real-time querying distribution changes. We present results and observe overall QAC effectiveness for different periods of N days, at prefix lengths of 2 to 5 characters.

2. MOTIVATION

As time undoubtedly plays a central role in user search behaviour [2], it is important for QAC to suggest completions that become highly popular over very short periods (i.e. real-time temporal queries), while also supporting always popular '*head*' queries.

Relying on a long period of past query-log evidence will ensure QAC is robust for continually popular queries, however, it will also have the effect of smoothing over short-term popular queries. For example, imagine a scenario where query q_1 is consistently popular, appearing 1000 times each day in the query-log. Aggregating query popularity over a past 30 day period would mean that query q_2 which is popular only today would need to be appear 30,000 times before it outweighed the long-term popular query in a probabilistic QAC approach. At the same time, reducing the aggregation period may mean the long-term popular query q_1 is not adequately represented, allowing short-term noise to reduce its ranking.

Ultimately, developing an effective QAC system that can respond to real-time temporal trends is a trade-off between robustness and sensitivity. In this paper we aim to study this trade-off in terms of how much past query-log evidence is optimal for aggregating query popularity, and how this changes for each prefix length. Moreover, as there has been little experimentation on open datasets, this work establishes baseline QAC performance for further studies.

The effectiveness of using a shorter query-log evidence aggregation period has been noted previously, particularly for real-time temporal queries [10]. While time-series modelling for query trends is able to improve QAC for recurring predictable temporal trends, for short-term real-time temporal queries it often proved problematic due to lag and over-fitting [10]. Time-series models were not able to model the increasing trend quickly enough, and likewise, continued to predict increased popularity for some time after the brief period of actual popularity.

2.1 Temporal Query-log Analysis

We quantify the extent to which the query-logs are composed of real-time temporal querying, in order to determine the degree to which QAC must support this behaviour. We define real-time temporal queries as those which appear as a 'spike' - with the vast majority of their occurrence within a short period, e.g. hours to days. Similarly, the queries are unlikely to have been recently popular, or even seen previously.

We analyse the temporal trends contained in two publicly available[2] datasets: the AOL [7] and MSN [1] query-log datasets. Extensive temporal analysis of longer-term and larger proprietary query-log data has been performed previously by others [6, 4, 2].

The AOL query-log contains 36.3M user interactions over a 3 month period from the 1st March 2006 to the 31st May 2006. The MSN query-log contains almost 14.9M user interactions over a 1 month period from the 1st May 2006 to the 31st May 2006.

Query-log entries necessary for identifying result clicks were removed. By extracting all the unique query and timestamp combinations, we obtained only queries directly typed by users. Navigational queries containing the URL substrings: .com, .net, .org, http, .edu or www were removed. We were left with 21.8M and 12.2M queries for AOL and MSN, respectively. Preliminary analysis discovered a sizeable number of short bursts of what we suspect is bot spamming activity in the AOL query-logs. Generic queries such as '*personalfinance*', '*aolcelebrity*', '*computercheckup*', appear in high volume with very uniform spacing (e.g. every 30 or 60 seconds). We manually observed and removed around 10,000 instances of these queries from our analysis.

	Volume of Queries	
Window Size (Days)	**AOL**	**MSN**
1	9.2%	3.5%
3	10.1%	4.5%
5	10.4%	5.1%

Table 1: The volume of queries in each query-log which were used ≥ 4 times, and for which 80% of their overall occurrence is within a window of N days.

In Table 1 we present the volume of queries (i.e. % of the total queries submitted in the query-log) which occur four or more times, and have at least 80% of their use concentrated within a period of N days. In AOL, the most popular 1 to 5 day highly temporal queries include: '*amelia earhart pictures*', '*karl der grosse*', '*the simpsons live action*' and '*leisure suit larry*'. Likewise, in MSN among the most popular are: '*stephen colbert*', '*poison milk*', '*ohio bear attack*' and '*kimberley dozier*'. Investigation shows that many of these queries describe, or are strongly related to significant events.

These results suggest a reasonable volume of real-time temporal queries in both query-logs, at least in the relatively short periods we are able to study. We suspect that the percentage of real-time temporal queries will have substantially increased in more recent query-logs, given the increase in real-time media available on the internet.

3. RELATED WORK

The majority of research has concentrated on the inherent engineering complexity of providing efficient and scalable QAC, which is resilient to typing errors. There have been relatively few studies on improving QAC effectiveness in search engines; likely due to the fact that there are few suitable query-logs available outside industrial search engine companies for experimentation.

Exploiting the user's personal context, and past query sessions has led to considerable QAC improvement, especially for shorter prefixes [3, 8]. Shokouhi and Radinsky [9, 10] used time-series modelling of past temporal query patterns to improve QAC effectiveness. Popular queries recurring during specific temporal intervals, such as day/night, day of week, month, etc. were modelled so that current query popularity could be predicted based on prior evidence only. Shokouhi and Radinsky [10] propose the short time window technique we experiment with in this paper as a baseline (which they refer to as p_1, etc.). They note its relative effectiveness, particularly for correctly predicting short-term highly temporal and unpredictable queries for which time-series modelling is problematic. However, no detailed analysis on the performance impact of aggregation period for each prefix length is performed.

[2]MSN available on request. We justify our use of AOL as we study the data without identifying individuals.

4. AUTO-COMPLETION APPROACH

The common "standard" approach to QAC is Maximum Likelihood Estimation (MLE), based on past query popularity (i.e. '*most popular completion*') [3]. MLE for a prefix ρ_n (of n characters), with each query q in all past queries Q prefixed by ρ_n, is formalised as follows:

$$MLE(\rho_n) = \arg\max_{q \in Q} P(q) \quad (1)$$

$P(q)$ is the probability of the query appearing in the past query-log. We refer to this method, aggregating all query-log evidence prior to the current time q_t as our baseline **MLE-ALL**.

4.1 Limiting Past Query-log Evidence

We propose using only the last N days of query-log evidence (e.g., $N = 2, 4, 7$ or 14 days) for computing $P(q)$ at q_t (i.e., a *sliding window* of past evidence). We refer to this approach as **MLE-W*N***.

The intuition underlying this approach is that a more recent and limited period of queries may more accurately reflect the current query distribution. Similarly, although consistently popular queries will still be adequately reflected in the distribution, their total frequency will no longer be great enough to outweigh the frequency of popular queries that only spike in shorter periods.

5. METHODOLOGY

The objective of our experiment is to study the trade-off between sensitivity and robustness of QAC, for different prefix lengths. As such, we explore various query-log aggregation periods for each prefix length, and measure the effect on overall QAC performance.

Our experimental methodology simulates a real-time user search scenario; such that the user types a prefix, and receives completion suggestions based only on evidence prior to the time of their query. QAC effectiveness is measured by the presence, and rank of a ground-truth match for each set of suggestions.

A time-ordered query-log provides a stream of ground-truth user queries. We assume that each query present in the query-log is the result of a user having manually typed it into the search box. As such, for each prefix of length n of the query, QAC provides completion suggestions. Each suggestion is matched with the ground-truth of the user's actual query (we discuss matching in the following section).

Evaluation Metric. Similar to past QAC work [3, 10], we rely on Mean Reciprocal Rank (MRR) to observe the effectiveness of each QAC approach. Reciprocal Rank (RR) has typically been used for evaluation in IR situations where there is a single relevant document. For a set of completion suggestions S, RR is computed as:

$$RR(S, q_{intended}) = \frac{1}{S, Rank(q_{intended})} \quad (2)$$

If no match for $q_{intended}$ is present, then a RR of 0 is assigned (avoiding divide by zero errors). MRR is then computed as the arithmetic average of RR for all queries.

MRR reflects the user interaction model of QAC; a higher-ranked completion suggestion is more beneficial, but the difference in ordering of lower-ranked completion candidates is less significant. That is to say, there is less noticeable difference between a correct completion suggestion ranked at either the 3rd or 4th position, compared to the 1st or 2nd position. We consider a literal lower-case string match between completion suggestion and ground-truth as a successful match.

6. EXPERIMENT

We conduct experimentation using the AOL and MSN query-log datasets. By experimenting with millions of queries contained in each query-log, we achieve a representative indication of how each approach would perform in a real-world setting.

Using two different query-logs validates the approach across two query samples of varying characteristics, and different user populations of the two industrial search engines. The exact sampling and construction of each query-log is unknown. MSN has been filtered for privacy (e.g. clearing known number patterns, such as phone numbers), appears to contain fewer adult queries, and is more in-depth as it is only for a one month period. In contrast, the AOL query-log contains more queries, but has greater breadth as it covers a three month period. However, as noted in [2], the sampling of AOL may not be truly representative of normal querying distributions because of re-finding behaviour.

6.1 Experiment Settings

We report results in Section 7 for two QAC settings: MLE-ALL using all query-log evidence prior to q_t (we treat this as the baseline), and MLE-W*N*, with 2,4,7 and 14 days of past query-log (characterising short and medium-term event/evidence periods). With only sampled query-log datasets, reducing the evidence period further leads to relatively sparse querying data. To emulate a real user interface scenario, we assume the user would see 4 highest-ranking completion suggestions for each prefix they input.

We run each approach for all query prefixes of 2 to 5 characters. Experiments for each prefix length were run independently, hence a successful completion suggestion at a prefix of 2 characters had no effect on the later evaluation for 3 characters.

Learning Period. We report the MRR of MLE QAC computed over the period of the query-log, minus the first N days which we treat as the learning period. Doing this makes the MRR obtained from MLE-ALL and MLE-W*N* directly comparable as both are computed over exactly the same set of queries. In any case, QAC performance during this early period will be extremely low as there is very little query popularity evidence (i.e. the 'cold-start problem'), and wouldn't reflect a real-world scenario where a QAC system would almost always be trained on past evidence.

7. RESULTS

Table 2 presents the overall MRR observed for MLE QAC experiments on the AOL and MSN and AOL query-logs; using the past 2, 4, 7, 14 days as well all past query-log evidence, for prefix lengths of 2 to 5 characters. The MLE-ALL MRR reported beside each MLE-W*N* corresponds to the baseline using all queries prior to q_t, but with the first N days of queries excluded for comparison.

The aggregated statistical power of 21.8M and 12.2M RR measures (i.e. each query) provided by the AOL and MSN experiments, respectively, means that the results we report are statistically significant according to standard t-tests [5]. Therefore, our analysis concentrates on the effect size of each window period over the baseline - that is, change in MRR over the corresponding MLE-ALL.

Firstly, for all runs and both query-logs it is clear that QAC is considerably more effective with a longer (i.e. more specific) prefix. This is expected, given that each extra character in the prefix reduces the space of possible completion suggestions, thus increasing the chance of a completion suggestion match [3].

QAC is almost always more effective for MSN than for AOL, especially for prefix lengths of 4 or less characters. Using a sliding window of evidence has a significant effect on overall QAC performance in almost all cases.

For AOL there is a sliding window of evidence which can im-

ρ	MLE-ALL	MLE-W2	MLE-ALL	MLE-W4	MLE-ALL	MLE-W7	MLE-ALL	MLE-W14
AOL								
2	0.090	**0.091 (1.11%)**	0.090	0.091 (1.11%)	0.090	0.091 (1.11%)	0.090	0.091 (1.11%)
3	0.143	**0.147 (2.80%)**	0.143	0.146 (2.10%)	0.143	0.145 (1.40%)	0.143	0.145 (1.40%)
4	0.185	0.189 (2.16%)	0.184	**0.189 (2.72%)**	0.184	0.188 (2.17%)	0.184	0.187 (1.63%)
5	0.217	0.215 (-0.92%)	0.216	0.217 (0.46%)	0.217	0.218 (0.46%)	0.217	**0.219 (0.92%)**
MSN								
2	0.112	**0.117 (4.46%)**	0.111	0.115 (3.60%)	0.111	0.113 (1.80%)	0.110	0.111 (0.91%)
3	0.164	0.163 (-0.61%)	0.164	**0.165 (0.61%)**	0.164	**0.165 (0.61%)**	0.164	0.164 (0.00%)
4	0.197	0.188 (-4.57%)	0.197	0.193 (-2.03%)	0.197	0.196 (-0.51%)	0.197	**0.197 (0.00%)**
5	0.215	0.197 (-8.37%)	0.216	0.205 (-5.09%)	0.216	0.211 (-2.31%)	0.218	**0.216 (-0.92%)**

Table 2: MRR observed for QAC when using all prior query-log evidence, and the past 2, 4, 7 or 14 days of query-log evidence. Prefix (ρ) lengths of 2 to 5 characters are reported for the AOL and MSN query-logs. The best performing sliding window setting is highlighted for each prefix length (although in some cases this is still outperformed by the baseline, at least for the reported window periods).

prove QAC performance over MLE-ALL for all prefix lengths, albeit relatively marginally for 5 characters. Using a shorter 2 day window of evidence improves QAC performance by up to nearly 3% for shorter prefixes of 2 or 3 characters. For a prefix of 4 characters, using a little more evidence, e.g. 4 days is optimal. Similarly, the best performance for a 5 character prefix is obtained when using 14 days of evidence.

For MSN, shorter prefixes (e.g. 2 or 3 characters) can outperform the baseline when using a window of evidence. Specifically, we see the best performance improvement of nearly 4.5% when using 2 days of evidence for a 2 character prefix. However, using between 2 and 14 days of query-log evidence always impairs QAC performance compared to the baseline for 4 or 5 character prefixes. Notably, the detrimental effect on performance is reduced as the sliding window of evidence is increased.

8. DISCUSSION AND CONCLUSION

The baseline QAC performance, and the following sliding window QAC improvement characteristics for each query-log are considerably different between the query-logs. AOL QAC is marginally but consistently improved in almost all cases, whereas MSN QAC is only improved for shorter prefixes, albeit much more so than for AOL. This suggests that the two query-logs have different temporal characteristics. In part, this may be caused by a couple of factors. Firstly, although AOL has more queries, it is spread more sparsely over a three month period, in contrast, MSN queries are concentrated in a 1 month period. Additionally, AOL has a day of missing data [2] which will harm QAC effectiveness following the affected period. Secondly, there may be underlying demographic differences between users of the two search engines that lead to changes in query distributions.

Although the performance improvement characteristics for each prefix and sliding window are different for each query-log, there is a clear overall linear relationship emerging in the results between prefix length and optimal sliding window period. As such, QAC for shorter prefixes performs optimally with a shorter sliding window of evidence, and conversely, QAC for longer prefixes performs best with a longer sliding window of evidence.

This relationship probably arises from the uncertainty posed by short and non-specific prefix lengths [3], where the space of possible completion suggestions is large. In these cases, using a shorter-period of evidence will still reflect long-term popular queries, but also be sensitive to temporal variation. Longer prefixes are more specific and thus narrow possible completion suggestions considerably. In these cases, real-time temporal factors are probably less likely to play a significant role in the already reduced set of possible completion suggestions. Moreover, for less common prefixes (and therefore rarer queries), relying on a longer query-log period is more likely to include the evidence necessary to rank them effectively as they were less likely to be used recently.

Conclusion. In this paper we examined the trade-off between QAC robustness and real-time temporal sensitivity. We found that QAC effectiveness can be improved by up to nearly 5%, simply by selecting the optimal time period of query popularity aggregation for each prefix length. The period necessary to achieve optimal QAC effectiveness varies by prefix length; shorter prefixes (e.g. 2-3 characters) perform best with only short-term evidence (e.g. 2-7 days), whereas longer prefixes (e.g. 4-5 characters) require more long-term evidence (e.g. 7-14 days, or more). Results also indicate the need to train per query-log, in order to capture intrinsic temporal and demographic characteristics. Care must also be taken with the sampling of queries used for training.

Further work will experiment with larger, more recent query-logs and perform cross-validation to verify the preliminary findings we present in this paper. Moreover, we will investigate alternative modelling techniques to improve QAC effectiveness for real-time temporal queries, which are problematic for time-series modelling as they spike so briefly in time.

Acknowledgments. This work was partially supported by the EU LiMoSINe project (288024).

9. REFERENCES

[1] *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, New York, NY, USA, 2009. ACM.

[2] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. Why we search: visualizing and predicting user behavior. WWW '07, pages 161–170, New York, NY, USA, 2007. ACM.

[3] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *WWW '11*, pages 107–116, 2011.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal analysis of a very large topically categorized web query log. *J. Am. Soc. Inf. Sci. Technol.*, 58(2):166–178, Jan. 2007.

[5] P. Ellis. *The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results*. Cambridge University Press, 2010.

[6] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *ACM WSDM '11*, pages 167–176, New York, NY, USA, 2011. ACM.

[7] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. InfoScale '06, New York, NY, USA, 2006. ACM.

[8] C. Sengstock and M. Gertz. Conquer: a system for efficient context-aware query suggestions. WWW '11, pages 265–268, New York, NY, USA, 2011. ACM.

[9] M. Shokouhi. Detecting seasonal queries by time-series analysis. In *SIGIR '11*, pages 1171–1172, 2011.

[10] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *SIGIR '12*, pages 601–610, 2012.