Use of Query Similarity for Improving Presentation of News Verticals

Annie Louis*
University of Pennsylvania
Philadelphia, PA 19104, USA
lannie@seas.upenn.edu

Eric Crestan *
Microsoft
81669 Munich, Germany
ericres@microsoft.com

Youssef Billawala
Yahoo! Labs
Sunnyvale, CA 94089, USA
billawal@yahoo-inc.com

Rao Shen
Yahoo! Labs
Sunnyvale, CA 94089, USA
raoshen@yahoo-inc.com

Fernando Diaz
Yahoo! Labs
Sunnyvale, CA 94089, USA
fdiaz@yahoo-inc.com

Jean-François Crespo
Yahoo! Labs
Sunnyvale, CA 94089, USA
jfcrespo@yahoo-inc.com

ABSTRACT

Users often issue web queries related to current news events. For such queries, it is useful to predict the news intent automatically and highlight the news documents on the search result page. An example query would be "election results" issued during the time of elections. These highlighted displays are called news verticals. Prior work has proposed several features for predicting whether a query has news intent. However, most approaches treat each query individually. So on a given day, very similar queries can be assigned opposite predictions. In our work, we explore how a system can utilize query similarity information to improve the quality of news verticals along two dimensions—prediction and presentation. We show via a study of actual search traffic that the accuracy of predicting queries into newsworthy and not newsworthy categories can be improved using query similarity. Further, we present a method to identify a canonical variant for a newsworthy query such that using the canonical query would retrieve better results from the news backend to show in the display. Use of the canonical query also has the advantage of creating a consistent presentation of results for query variants related to the same news event.

1. INTRODUCTION

Consider the query "Michelle Obama visits Spain" issued to web search engines *during the time* of the First Lady's visit to Spain. If this query is issued on a news website, the results would contain all news articles about the event. On the other hand, a user may also issue such a query to a generic web search engine. If the search engine can predict the query as related to currently newsworthy events, latest news can be retrieved for the query and explicitly highlighted on the result page. Such news displays, also called news verticals now appear on the result pages of major search engines such as Google, Yahoo! and Bing. An example news vertical on the search result page of a generic search engine is shown in Figure 1. Prior work has introduced several features for news intent prediction. These features characterize the bursty nature of a query during a particular time period [8, 16, 9]. In this work, we show how to improve the presentation of news displays by making use of query similarity information.

Figure 1: A news display above search results

We consider two properties related to news displays which can further benefit from information about similar queries.

Triggering accuracy: When two queries q and q' are related to the same news event, they would have the same news intent. For example, the queries "Michelle Obama Spain visit" and "Spain vacation Michelle Obama" refer to the same event and both are newsworthy during that time. If a system can identify q and q' as similar, it can make more accurate predictions. Besides accuracy, it is desirable that a consistent system would trigger a news display for both queries regardless of the actual lexical realization.

Retrieval of results for the display: Another use of query similarity is in the case when q and q' are predicted to be newsworthy. The lexical form of q could retrieve very relevant content from the news backend while q' might be a specific query with few matches. Knowledge that they

9*Work conducted at Yahoo! Labs.

are similar can help us choose from q, q' and other similar queries, a variant which would retrieve the most relevant results for the event that they represent. This setup could also lead to uniformity in the results shown for different variants belonging to the same event. So query similarity can help in presentation quality and consistency.

The work on news intent prediction that is closest to ours is by Diaz and Arguello [8, 9] who seek to improve predictions by incorporating user feedback. In their method, user clicks on a query's news display are tracked and used as feedback not only for that query but also for similar ones. They compute similarity between queries based on the similarity in their search results. In this work, we show how to adjust the predictions themselves using similarity information. Their method also requires the results of the search which is expensive to compute. The selection of a canonical variant and the issue of presentation consistency has not been explored so far.

In this paper we augment a news intent predictor with query similarity information.

(a) We show that a simple lexical match method to get similar queries is highly accurate and useful for improving news intent predictions. We show that the improvement is pronounced even in a model which uses a rich set of features for the baseline predictions.

(b) We validate through a user study on actual search traffic, that the predictions get improved when we add query similarity information.

(c) We propose a method to identify a canonical variant for similar queries. We show that we can outperform the baseline approaches significantly for this task and achieve a better display of news results for different query variants.

Our study is insightful for both the quality of news display and the search user experience. Inconsistent triggering lowers accuracy. Recent work [4, 22] recognize that long and rare queries are also important components of user experience but do not always return (informative) results. So approaches have made use of query similarity to address the needs of tail queries. Our work continues along these lines for a different search task. From a user experience perspective, particularly with explicitly highlighted displays, it would be jarring if the choice to show a display and the presentation style were markedly different for closely related queries. It would be particularly difficult for users to find a certain article again using a different but equivalent query (referred to as the task of refinding [6, 7]). We seek to add accuracy and quality in the displays with the aid of query similarity.

2. RELATED WORK

Prior work on news intent prediction have introduced a number of features [8, 16]. Three major resources are utilized—query logs, news index and blogs. When a query appears with greater frequency in the documents belonging to a time period compared to recent past, it can be called newsworthy. Work by Diaz and Arguello [8, 9] extends beyond features and incorporates user feedback as well to adjust predictions. They obtain a prior news intent prediction based on features and then correct it as user feedback is obtained.

In most prior approaches, the vector of features for a query is not dependent on other queries issued the same day. Even when two queries are closely related, they could have varying frequencies in query logs and news documents. So their predictions could be quite different. We want to incorporate query similarity to create more accurate predictions. In Diaz and Arguello's work, they share user feedback from one query to those which are similar, so that similar queries can also take advantage of the feedback information. In this work, we seek to produce more accurate baseline predictions using query similarity. Our approach to do similarity-aware predictions is similar to techniques introduced to handle tail queries for advertisement display [5, 4, 22] and query suggestions [1, 20, 23, 17, 21]. There is also recent work which create specialized ranking models for a query by making use of similar queries in the training data [11, 18].

Our approach to canonical query selection is reminscent of query modification methods to obtain better matches with documents or advertisements. Here, when queries do not match the bid phrases or documents, some substitution [15], deletion [14], and other modification [12] is necessary. Our situation is different in that we desire a concise query which can be issued to get the most relevant results from the news index. So we need to judge the candidate queries based on both their match to original query as well as the news documents they retrieve from the news index. The idea is similar to work by Baeza-Yates et al. [1] who ranked query suggestions not only by similarity but also considering the attractiveness or quality of the suggested query.

3. NEWS INTENT PREDICTOR

We consider a simple news predictor and then show how we add query similarity information to it. We predict newsworthy queries in an offline process. To identify newsworthy queries for day d, we use the query logs from day $d-1$. The queries from immediately previous day are likely to be issued again. (In fact, one-third are repeated on average.) Each query from day $d-1$ is associated with a binary prediction (newsworthy or not) and the newsworthy queries are are entered into a whitelist. On day d, if a query issued is in the whitelist, we trigger a news display. The list is refreshed daily. So the goal of the predictor is to identify queries that will continue to have news intent on the next day.

This situation is obviously incapable of handling new queries that were not seen the previous day. However, for our goal of evaluating the use of query similarity, this setup is simple and appropriate, also enabling us to deploy it for a user study. Although the system is simple, this offline setup enables us to use a rich set of features which would be prohibitively costly to compute at query issue time. So we are evaluating the power of query similarity over an already strong feature set. So we focus on this baseline model for our experiments, further approaches such as adding user feedback and handling new queries will improve the capabilities of the model we use.

3.1 Data

The gold standard news intent judgements for a query was obtained empirically as was done in prior work. We logged query statistics in early 2008 from a commerical web search engine. A small portion of search users were shown news displays for *every* query they issued provided there was any matching content in the news backend. On average, 800,000

2008 long beach grand prix	long beach grand prix (0.57), f1 grand prix (0.03), long beach (0.02)
2008 nfl draft order	2008 nfl draft (0.30), 2008 draft (0.19), nfl draft (0.12), nfl draft 2008 (0.06)
bad beef recall	beef recall (0.15), recalled beef (0.02), massive beef recall (0.01), california beef recall (0.01)
carly smithson controversy	carly smithson (0.10), carly smithson american idol (0.01), carly hennessy smithson (0.01)

Table 1: Lexically similar variants for a query (similarity value within parentheses)

unique queries were issued per day by these users and 30,000 of them had matches in the news index. For these 30,000 queries, displays were shown without any attempt to verify their news intent. The display consisted of the titles of the three most relevant news documents. The click through rate (CTR) on the display for each query was logged and we consider these values as gold standard. A click on any of the three titles within the display is considered as a click on the display. The 30,000 queries per day collected for a two week period form our data set. We train a model to predict the CTR of a query and then apply a threshold to obtain a binary (newsworthy or not) prediction.

3.2 Prediction model

As already outlined, our method predicts newsworthy queries for day d from the query traffic logged for day $d-1$. So the target for training is the *next day CTR* of a query. Only those queries observed on two consecutive days can be used for training. We divide the data described above into training (around 100,000 unique queries) and test (60,000) sets.

Overall, we use around 70 features in our model. Most of them are inspired by prior work by König et al. [16] and Diaz [8] and use query logs and news index. Some of the distinguishing log-based features are: ratio between frequency of the query on this day and that during the previous week/days, number of times the query was issued on a news search engine, the output from a navigational intent predictor, the observed click rate on search results from news domain versus other webpages. The following features are among the most discriminative ones based on the news index: matches with titles and abstracts of news documents, the maximum score of the news documents retrieved, matches in different categories such as business, health and sports, the number of relevant documents in the backend compared to the number observed the previous week.

A Gradient Boosted Decision Trees (GBDT) [10] training approach is used to predict the real-valued next day CTR. Then we apply a cutoff on the predicted value to obtain a binary decision (newsworthy or not). To decide on the cutoff, we consider the set of queries above different points on the predicted value. For each of these sets, we compute the average value of their gold standard next day CTR. The set that has an average value of 20% is taken and the corresponding threshold value is chosen as the cutoff. This choice guarantees that on average if this threshold is applied on the predicted value, the queries chosen as newsworthy will have an average next day CTR value of 20% (a reasonable CTR expectation for news events).

Next we explain how we add query similarity information to the predictions.

4. COMPUTING SIMILAR QUERIES

We use a simple metric that measures the match between bigrams in the two queries. Such a lexical measure is inadequate when we consider query variants such as "Jennifer Lopez babies" and "Marc Antony twins". So we also tried other methods which offer more flexibility for matching queries. These methods included the similarity in the URLs, titles and abstracts of the top results [2, 19, 23, 8], tracking user sessions to find rephrased queries [23, 3] and a dictionary-based identification of similar named entities (Eg. Jennifer Lopez, Mariah Carey). We found that these looser notions (and their combinations) introduce several false positives for query similarity while lexically similar queries (with bigram match) had very high precision. For our task, particularly canonicalization, high precision is necessary and so we choose to use lexical match as our measure.

The words in the query are stemmed and each bigram of stems is a feature. We also include skip bigrams with gap 1, i.e. we form bigrams by ignoring atmost *one* intervening word. The feature value is the frequency of the bigram in the query multiplied by the inverse-document frequency (idf) computed on all the queries seen that day (recall that our processing is offline and all queries on day $d-1$ are available). The vectors from two queries are compared using cosine similarity. A cutoff is applied on the similarity value and we consider those query pairs above the threshold as similar. A threshold value of 0.01 was picked by manually examining the query pairs from different threshold ranges. We also limit the number of similar queries to atmost 10. Some example matches are shown in Table 1.

5. PREDICTION RESCORING

Now, we add query similarity to the news intent predictions using a post-processing approach.

5.1 Algorithm

Let us call the predicted CTR for a query q as *score(q)*. Next, we gather the most similar queries to q, *kNN(q)*, using lexical similarity and applying the threshold we described earlier. Now, we want to modify *score(q)* such that if the majority of neighbours in *kNN(q)* are newsworthy, it is likely that q is also such or vice versa.

Some ways to modify *score(q)* are: assign the weighted average of *score* for queries in $kNN(q)$, or assign the *score* of the most similar query from *kNN(q)*, etc. During development testing, we found that the predicted scores from the baseline model were more accurate for high frequency queries. So setting *score(q)* to be the same as that of its variant in *kNN(q)* with *maximum frequency (was issued most times)* provided the best CTR adjustments which moved queries reliably across the boundary for newsworthy/ not newsworthy decision. (We evaluate this aspect using the true category of queries where the system proposed to make a change in category.) We also found that the queries predicted with high confidence in the newsworthy category were quite accurate and demoting scores led to a number of false negatives. So we only focus on adding queries to the whitelist.

Our rescoring method can be summarized as follows. Let T be the threshold on *score(q)* to mark newsworthy queries.

	Added (%)	Removed (%)
Very newsworthy	101 (40.4%)	78 (31.2%)
Newsworthy	48 (19.2%)	55 (22.0%)
Somewhat newsworthy	23 (9.2%)	35 (14.0%)
Not newsworthy	78 (31.2%)	82 (32.8%)
Total	250	250

Table 2: Human newsworthiness judgements

$$\begin{aligned} \text{modScore}(q) &= \text{score}(q), \text{ if } |\text{kNN(q)}| = 0 \text{ or score(q)} > T \\ &= \text{score}(\underset{q' \in \text{kNN}(q)}{arg\,max}\ \text{frequency}(q')), \text{ otherwise} \end{aligned} \quad (1)$$

5.2 Evaluation

We employed both human judgements and system deployment for evaluating our approach. The human evaluation was done on a sample of our data. The deployment was done for a fraction of users of a commerical search engine and serves as a large scale study in the target setting. Both cases confirm the usefulness of our approach.

5.2.1 Human evaluation

We ran our rescoring method on one day of full web traffic in 2010 containing over 20 million unique queries. The default whitelist had 33,000 queries. When the scores were modified, the size increased to 57,000. We selected a random sample of 250 queries from those added to the whitelist and obtained annotations for their newsworthiness. To also confirm our choice to not remove queries from the whitelist, we also created a list of queries that would be demoted had we kept that option. A random sample of 250 queries from this list was also included for annotation.

The queries were presented on the next day to annotators. We did not use the same click data as for development because it is difficult for people to judge the newsworthiness of events from the past. We asked them to rate the news intent of queries on a 4-level scale a) very newsworthy, b) newsworthy, c) somewhat newsworthy d) not newsworthy. The division that the judges provided are shown in Table 2.

From the results, we see that 60% of queries added to the whitelist are clearly newsworthy. An additional 10% of the added queries are on the borderline. So a number of queries missed by the default predictor are added to the whitelist. We also see from the annotations for the removed queries, that removing queries from the whitelist seems not to be a good option. 30% of those removed are actually rated 'very newsworthy'. However, it should be remembered that these judgements are only on a small sample. We also had 50 queries simultaneously annotated by 3 judges. The pairwise annotator agreement (Kappa) is only 0.29 to 0.36. Collapsing the levels into two categories—'very newsworthy' and 'newsworthy' into one and the other two into a second category—increases the agreement to only 0.44 to 0.60 range. So it is rather difficult for human annotators to agree on the news level of events even for the most recent time frame. So we use this evaluation to validate our design and focus on system deployment to test if there are any significant improvements in the user experience.

5.2.2 User study

Measure	baseline	rescored
CTR	10.43	10.67
Coverage	2.32	2.39

Table 3: Results from user study

We studied two small fractions of search users. There were 37 million unique queries issued by each set of users on a given day. During a two week period, news verticals were shown for one set of users as per baseline predictions. For the other, the rescored predictions were used. It is important to note that both systems were employed simultaneously during the two weeks, so users would be querying for the same events in both tracks. The CTR and coverage (proportion of queries that were shown displays) for the two branches are shown in Table 3.

After rescoring, there was a 2.36% improvement in CTR compared to the baseline approach. At the same time, the system was also able to show displays for 3.13% more queries than the baseline. Approximately same number of users interacted with the baseline and new system during the two week period. The click rate of each user was used as a data point and a Wilcoxon test was used to compare these values between the two systems. Both CTR and coverage are significantly different with p-values lower than 0.0001.

These results show that after rescoring, we were able to show news displays for more queries at the same time improving the average user engagement on the displays.

6. QUERY CANONICALIZATION

Making accurate predictions is only a first step towards display quality. The results retrieved for newsworthy queries should also be informative. Here again we can make use of information from similar queries. So we propose an approach to query canonicalization. For *each newsworthy query*, we identify a canonical variant which retrieves the most relevant results from the backend. This canonical query might be the original query itself or a semantically equivalent variant. In this way, quality results can be shown regardless of the query variant actually issued. Further this approach would ensure that similar queries have some uniformity and consistency in what results get highlighted for that news event.

6.1 Data and setup

This experiment is based on the data described in Section 3.1. We try to find a canonical variant for each query in the whitelist. Those queries not predicted as newsworthy will not trigger news displays and need not be considered. For each query in the whitelist, we obtain a set of candidates for canonicalization. We limit these candidates to other queries *in the whitelist* which are highly similar (using the same threshold on lexical similarity as we have done so far). The original query is always added as a candidate.

The task is to pick out the candidate that will have highest success as a canonical variant. The whitelist queries are compiled to be used on the next day. So we define the most successful query as the one with highest next day CTR.[1] However, we should avoid choosing very specific and rare queries as canonical. For example, a query such as "Obama Texas debate opposition response" is rather specific and could have a CTR value of 1, but might have been

[1] Only queries overlapping between consecutive days were used.

Original query	Lexical variants	Canonical query
1. jennifer lopez marc anthony	marc anthony, jennifer lopez, jennifer lopez baby, jennifer lopez babies, jennifer lopez twins	jennifer lopez twins
2. presidential candidates	republican presidential candidates, 2008 presidential candidates	2008 presidential candidates
3. shooting down satellite	satellite shoot down, US shoots down satellite, spy satellite shot down	satellite shoot down
4. nfl draft combine	nfl draft combine, nfl combine, nfl draft, 2008 nfl draft,nfl draft 2008, nfl scouting combine, nfl mock draft	nfl combine

Table 4: Example original and canonical queries

issued only once. We do not wish to canonicalize queries such as "Obama debate" and "Obama Clinton texas debate" to such a specific query. So we filter out candidates that were issued fewer than two times that day. (The original query is always retained as a candidate regardless of its frequency.) Then we also include the number of clicks with the CTR value while ranking the candidates: {next day CTR * log(next_day_clicks+1)}. The top variant in this ranking is considered canonical.

Some examples from our data are shown in Table 4. We can see that the canonical variant chosen in this way is a crisp characterization of the event. Since we use lexical similarity, we can expect the candidate set to be of high precision. The candidates have at least one bigram match with the original query and so these matching keywords will still appear in the search results. We divide the data into a training set of 2876 examples and a test set of 1659. The breakdown of the number of candidates for different queries in our test set is below.
(a) 2 candidates: 800
(b) 3 to 5 candidates: 636
(c) 6 and above: 223

Our definition of canonical queries holds for a period of *one day only*. For the whitelist queries, we identify and store the mappings of original-canonical queries. On the next day, when a query from this list is issued, the canonical variant can be used for retrieving news results in place of the original one. The list is refreshed at the end of the day. It is also worth noting that we wish to present canonical results for the news display only so that the relevant news updates are provided for all query variants. The regular search results would still provide more fine-grained distinctions (if any) as per the original query.

6.2 Canonicalization methods

We test several *baselines* for this task: a) a random query from the candidate set, b) the original query itself (no canonicalization), c) the query with maximum frequency that day, d) the query with maximum score from the GBDT model (the model predicts the next day CTR and high CTR queries could be considered as more representative ones), e) maximum and f) minimum length queries.

We also propose a ranking approach to canonicalization. The target value {next_day_CTR * log(next_day_clicks + 1)} is used to train a ranker to predict the ordering of queries within a candidate set. For this purpose, we employ SVM-Rank [13], whose training approach seeks to minimize the number of pairwise wrong orderings over the training set. The regularization parameter was chosen using cross validation on the training set. The output of the ranker is a real-valued score for each candidate which can be used to rank them. The highest ranked query is marked as the canonical one. Our ranking approach combines a number of features with the baseline metrics.

	All	2 cand.	3 to 5	above 5
Random	0.43	**0.59**	0.30	0.10
Self	0.43	0.59	0.32	0.17
Max. frequency	**0.53**	0.58	**0.53**	**0.34**
Max. offline score	0.44	0.59	0.33	0.19
Max. length	0.34	0.49	0.23	0.13
Min. length	0.42	0.56	0.32	0.16
Ranking approach	**0.56**	**0.60**	**0.55**	**0.44**

Table 5: Accuracies for canoncial query selection from different candidate set sizes

Simple metrics. Frequency of the query, offline model score, length of the candidate

Relationship to original query. Is the original query?, is substring/superstring of original query?, is subsequence/ supersequence of original query?

Content-based. These features are based on the relevance of the news documents that are present for a candidate query in the index. In addition, the readability of the titles from these news documents is also an important factor for quality of the news display. We retrieve the top 3 titles for the query from the news index and compute the following features: length of the titles, query is subsequence/substring of one of the titles (similarly for first title), whether there is a title string that is spelled fully in capital letters.

Informativeness of the query. Words used in queries could make some queries more informative compared to others. So we add some features based on query word frequency and word types. We compute the frequency of each query word and bigram over all the candidates in a set. Then for each query, we indicate whether it contains the top, second or third most frequent unigrams and bigrams of that cluster. Numbers or year also get mentioned in some queries, for example, "grammy awards 2008". To check the usefulness of such information, we add the features: query contains number/year, query's prefix/suffix is a number/year. We also indicate the presence of prepositions.

6.3 Evaluation

The accuracies for predicting the canonical query is shown in Table 5 for different baselines and the ranking approach. We also provide a breakdown of the results depending on the size of the candidate sets (columns 3-5): choosing the canonical variant from a larger set of options is bound to be a harder task.

Among the baselines, using the maximum frequency query produces the best results. It has an accuracy of 53% on all test examples, 10% higher than random choice. The maxi-

mum frequency heuristic is also consistently better performing than the other baselines for larger sets. When a set contains only two candidates (original query and one other variant), random choice, the original query or using the top offline score query are equally efficient, giving 60% accuracy. The accuracy of these baselines degrades much more as the candidate set size increases.

The ranking approach outperforms the baselines for candidate sets of all sizes. There is a 3% improvement when all the test examples are considered. In closer look, one can see that the benefit of the ranking approach is more pronounced when there are more candidates. It gives a 10% improvement over the maximum frequency baseline for sets of size above 5 queries. In fact, the very popular queries are likely to have several variants and therefore a large candidate set. So the ranking approach would be the preferred setup for canonicalization in this case.

7. CONCLUSION

In this work, we have shown that our system which used query similarity had better user engagement and also increased the recall of newsworthy queries. There are several issues that we can address in future work. One is purging non-newsworthy queries from the whitelist. Since similarity-only rescoring was not accurate to do this, we have not addressed this aspect. Perhaps, a combination of query similarity to add queries and user feedback to quickly learn how to demote queries would be a good approach. We will explore these ideas in future.

In our approach, we have not only considered the prediction accuracy but also the presentation of the display. We have introduced the notion of a canonical query for news intent. As a simple baseline for canonical query, maximum frequency query is a good choice and the selection is further improved using query match and content features. But we have only worked with CTR data in this paper. We plan to conduct a user study to strengthen the result and help us to understand if canonical results are preferred by users.

8. REFERENCES

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query recommendation using query logs in search engines. In *Current Trends in Database Technology - EDBT 2004 Workshops*, pages 395–397. 2005.

[2] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proc. of KDD*, pages 407–416, 2000.

[3] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *Proc. of SIGIR*, pages 515–522, 2010.

[4] A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, D. Metzler, L. Riedel, and J. Yuan. Online expansion of rare queries for sponsored search. In *Proc. of WWW*, pages 511–520, 2009.

[5] A. Z. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *Proc. of SIGIR*, pages 231–238, 2007.

[6] H. Bruce, W. Jones, and S. Dumais. Information behaviour that keeps found things found. *Information Research*, 10(1):10–1, 2004.

[7] R. Capra III and M. Pérez-Quiñones. Using web search engines to find and refind information. *Computer*, 38(10):36–42, 2005.

[8] F. Diaz. Integration of news content into web results. In *Proc. of WSDM*, pages 182–191. ACM, 2009.

[9] F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In *Proc. of SIGIR*, pages 323–330, 2009.

[10] J. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[11] X. Geng, T.-Y. Liu, T. Qin, A. Arnold, H. Li, and H.-Y. Shum. Query dependent ranking using k-nearest neighbor. In *Proc. of SIGIR*, pages 115–122, 2008.

[12] J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In *Proc. of SIGIR*, pages 379–386, 2008.

[13] T. Joachims. Training linear svms in linear time. In *Proc. of KDD*, pages 217–226, 2006.

[14] R. Jones and D. C. Fain. Query word deletion prediction. In *Proc. of SIGIR*, pages 435–436, 2003.

[15] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proc. of WWW*, pages 387–396, 2006.

[16] A. C. König, M. Gamon, and Q. Wu. Click-through prediction for news queries. In *Proc. of SIGIR*, pages 347–354, 2009.

[17] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *Proc. of CIKM*, pages 469–478, 2008.

[18] J. Peng, C. Macdonald, and I. Ounis. Learning to select a ranking function. In *Proc. of ECIR*, pages 114–126, 2010.

[19] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of WWW*, pages 377–386, 2006.

[20] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of WWW*, pages 377–386, 2006.

[21] Y. Song and L. He. Optimal rare query suggestion with implicit user feedback. In *Proc. of WWW*, pages 901–910, 2010.

[22] I. Szpektor, A. Gionis, and Y. Maarek. Improving recommendation for long-tail queries via templates. In *Proc. of WWW*, pages 47–56, 2011.

[23] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *Proc. of WWW*, pages 1039–1040, 2006.