Measuring Website Similarity using an Entity-Aware Click Graph

Pablo N. Mendes*
Freie Universität Berlin
Garystrasse 21, Berlin, 14195, Germany
pablo@pablomendes.com

Peter Mika, Hugo Zaragoza, Roi Blanco
Yahoo! Research Barcelona
Diagonal 177, Barcelona, 080018, Spain
{ pmika,hugoz,roi }@yahoo-inc.com

ABSTRACT

Query logs record the actual usage of search systems and their analysis has proven critical to improving search engine functionality. Yet, despite the deluge of information, query log analysis often suffers from the sparsity of the query space. Based on the observation that most queries pivot around a single entity that represents the main focus of the user's need, we propose a new model for query log data called the *entity-aware click graph*. In this representation, we decompose queries into entities and modifiers, and measure their association with clicked pages. We demonstrate the benefits of this approach on the crucial task of understanding which websites fulfill similar user needs, showing that using this representation we can achieve a higher precision than other query log-based approaches.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]; H.3.1 [**Content Analysis and Indexing**]

General Terms

Algorithms, Experimentation

Keywords

click graph, query logs, website similarity

1. INTRODUCTION

For both search engine providers and Web site owners, query logs are among the most valuable sources of information on how users interact with online content. For search engines, query logs are indispensable for providing such crucial services as query completion and 'also try' suggestions. For site owners, search referrals and site logs make it possible to discover the information needs of their users and optimize their presence on the web.

Most applications of query log analysis, however, suffer from the notable sparsity of the query space. In the case of Web search, Baeza-Yates [1] shows that 44% of the queries occur only once even when considering a full year of data. Query frequencies follow a power law, which means that a large fraction of the queries that appear more than once have very low frequency in general, and consequently offer a small number of clicks that can be used to determine their relationship to other queries.

As an illustration of the problem, Figure 1 shows the sets of queries that two hypothetical websites may receive in a query log. If we would try to measure the similarity[1] of these websites based on the overlap of these sets, we would find that no queries are shared between the two sites. However, it should be clear that the two websites are related based on the intent of these queries.

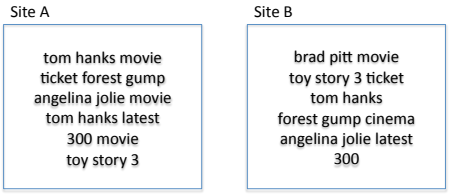

Figure 1: Queries leading to two different sites

In order to alleviate the effects of sparsity, practical studies on the click graph usually exclude low frequency queries or treat queries as bag of words [1], which destroys the semantics of queries resulting in data loss and/or bias [9].

In this paper, we propose the *entity-aware click graph* model for query log representation, which relies on breaking up queries into a named entity and a modifier, i.e. the remaining words that are not recognized as part of a named entity. Based on the manual annotation of 264 randomly selected queries from Yahoo! Search query logs, it was shown in recent work [20] that over 62% of the queries contain the name of an entity or type of entity that the user is trying to locate. In most of these queries, the name of the entity or type is surrounded by additional modifiers that narrow the search context, e.g. by specifying additional characteristic of the entity sought or by expressing the intent of the user with regard to the named entity or type. Context, however, is typically rather short, as users – accustomed to the con-

*Part of this work was done while the author was visiting Yahoo! Research Labs, Barcelona

[1]In this work we interchangeably use the terms *similarity* and *relatedness* to loosely mean semantic relatedness.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

junctive semantics of queries – try to keep the number of query words to a minimum.

We will show that it is indeed possible to capture large portions of query log data using this model and that this treatment preserves the structure that is inherent to the query log data, while reducing sparsity. Figure 2 shows how this relatively simple parsing alleviates the sparsity of the query space.

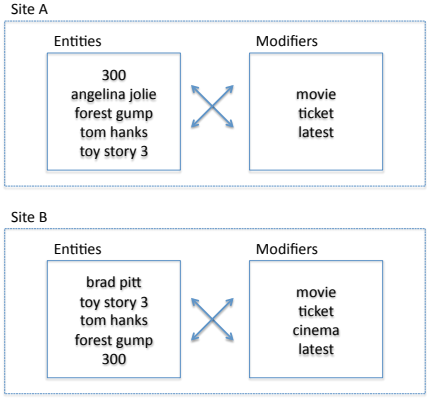

Figure 2: Queries from Figure 1 broken down into entity and modifier

As a demonstration of this model, we consider the problem of finding websites that provide similar services to a user. We will show that a simple entity spotting approach that leverages Linked Data on the Web can be effective in identifying website intent. In our current study we will use a high quality subset of named entities provided by Freebase,[2] a repository of collaboratively managed structured data exposed as Linked Data on the Web. By focusing on the entity and context dimensions separately, we will show that we can also identify two broad classes of websites, i.e. websites centered around particular entities vs. websites providing generic services.

2. RELATED WORK

Query log analysis is an important task in information retrieval and web mining research [16, 19, 5, 8].

Graph representations of query logs. One of the most widely used approaches for query log analysis is to model the query log as a graph. In a *query similarity graph*, two queries are connected by an undirected edge weighted by a given query similarity function (e.g. keyword overlap). As an alternative, Baeza-Yates and Tiberi [2] define a similarity metric based on common clicked URLs that has been shown to outperform keyword overlap. A second class of approaches models both queries and clicks. Craswell and Szummer introduce the *click graph* as a bipartite graph between queries and URLs to improve search [8].

In practice, all of the above approaches suffer from the sparsity of the query space. In order to obtain a more connected graph, researchers often perform cleaning steps (e.g. removing all infrequent queries) resulting in a loss of information and a bias towards frequent queries. An alternative is to treat queries as a bag of words [2], which destroys the semantics of many queries. In comparison, our entity-aware click graph model leads to less loss of information and preserves query structure, while considerably attenuating the sparsity problem.

[2]`http://freebase.com`

Query interpretation. The most common form of query interpretation is classification against a generic or domain specific taxonomy. Jansen, Booth and Spink [12] expanded on previous work in query intent classification (Broder, 2002 [6]; Rose and Levingson, 2004 [21]) to detail query intents in three hierarchical levels, and approached manually developed rules to classify 1.5 million queries. Li et al. [13] also target the problem of query intent classification. They model the query logs as a click graph and infer class memberships of unlabeled queries from those of labeled ones according to their proximities in a click graph. Hu et al. [11] present an alternative methodology to intent classification that uses the Wikipedia graph instead of the click graph.

We do not perform explicit query classification in our work. We annotate queries with entity identifiers from well-known Linked Data sources, separating mentions of entities from modifiers, and use this new model to classify websites. A key insight of our work is the orthogonality of these dimensions which we will illustrate with examples (see Section 3). This leads to potential interpretation of queries – and by proxy, clicked websites — across two dimensions: the entities and the modifiers found in a query. We will show that by separating these dimensions, we are able to outperform alternative methods that treat queries as a whole or as a bag of words on the task of computing website similarity.

Named Entity Recognition. A key aspect of our approach is the recognition of the internal composition of queries containing named entities and context words. Several approaches exist for the task of Named Entity Recognition (NER)[17] in the context of natural language text and in the context of query logs [18, 14, 10]. Determining the best entity extraction technique is out of the scope of this work. We focus on how to use the extracted entities and context words for analyzing the query logs.

3. ENTITY-AWARE CLICK GRAPH

A typical representation of a query log is the click graph, a bi-partite graph where the nodes are queries $Q = \{q_1, ..., q_{|Q|}\}$ and URLs – or, simplifying, the sites $S = \{s_1, ..., s_{|S|}\}$ hosting the URLs. An edge connects a query q_a to a site s_c if there has been at least one search session in the data where a user clicked on a URL in that site after issuing the query, but before issuing another query.

The entity-aware click graph models relationships between entities and modifiers appearing in queries and clicked sites, and can be defined as the union of two bipartite-graphs. Let $E = \{e_1, ..., e_{|E|}\}$ be the set of all entities and $M = \{m_1, ..., m_{|M|}\}$ the set of all modifiers.

We define $CG_{entity} = (E \cup S, (e_i, s_j))$ where an edge (e_i, s_i) exists if a user searched with keywords containing the entity e_i and visited site s_j. Analogously, $CG_{modifier}$ has edges (m_k, s_l) relating modifiers and sites. The entity-aware click graph is then defined as $CG = CG_{entity} \cup CG_{modifier}$.

A key feature of the entity-aware click graph is that entities and modifiers represent two distinct dimensions of the query space. As an illustration, we show the difference in which content vs. service-oriented websites behave along these two dimensions. The most common queries in our dataset (see Section 4 for more details) include “yahoo mail”, “facebook login”, “google search engine”, “bank of america online” and “kelly blue book”. When breaking queries into entities and modifiers, the most common entities include “yahoo”, “google”, “wells fargo”, “bank of america” and “face-

book", while the most common modifiers include "lyrics", "online", "games", "mail" and "bank". From the examples we can see that entities represent the topical content of the queries and the clicked websites, while modifiers commonly specify some service provision aspect with regard to the entity searched for and the website clicked.

In order to quantify the range of entities and modifiers that a website interacts with, we observe for each site the entropy of the probability distribution over entities and modifiers. Entropy is a measure of the 'informativeness' of a probability distribution. The more concentrated is the distribution, the less is its entropy; the more diffuse it is, the greater is its entropy. Formally, for a specific website s, we compute the negative entropies as: $H_e(s) = H(E|s) = \sum_i P(e_i|s) \log P(e_i|s)$. We compute H_m analogously for modifiers.

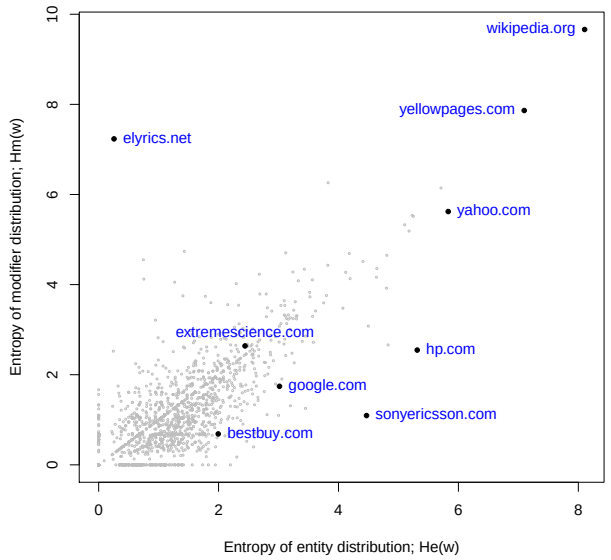

Figure 3: Plot of website relative entropies

Websites that are associated with several entities will have a high H_e, while those that are linked to few entities will have a low H_e. In this sense, H_e may be seen as a measure of entity specificity of a website: a more specialized content provider exhibits a lower negative entropy with regard to entities. Conversely, H_m can be seen as a measure of modifier specificity and, given the commonly observed modifiers, possibly indicating the range of services provided by a website.

In Figure 3, we show a two-dimensional plot where each point represents a website. The vertical axis marks the entropy of the entity-conditional distribution $P(C|w)$ and the horizontal axis marks $P(E|w)$ the conditional distribution with regard to modifiers. Figure 3 shows that websites cover a wide range of values on both scales and there is a spread of sites across all four quadrants. For example, `hp.com` and `sonyericsson.com` are both on the bottom-right quadrant, not because they co-occur with the same entities or modifiers, but because they exhibit a similar behavior with regard to our query model. They both represent brands and therefore relate to a small range of entities (names of the brand and models) while offering a wide-range of information from technical specifications to multimedia.

3.1 Measuring Website Similarity

Website similarity analysis is useful for understanding the structure of the Web at a macro level and can answer questions typically asked in the context of competitive analysis. Using a similarity measure, search engines can group thematically similar sites and also extend the suggestions to other sites that are not present in the search result, but contain similar content (for informational queries) or provide similar services (for transactional queries).

Given a bipartite click-graph, computing a website similarity graph is analogous to computing the query similarity graph [1]. This can be done, for example, by measuring the overlap of queries received by two web sites. In graph terms, this can be achieved by *folding* the bipartite click graph into a similarity graph, a one-mode graph where nodes are web sites and there is an edge between a pair of nodes if there was a path of length two connecting them in the bipartite graph. It is common to weight the resulting edges by the number of such paths.

As the example in figures 1 and 2 illustrate, website similarity graphs obtained from click graphs may fail to capture all relevant relationships. We propose to address this problem using the aforementioned entity-aware click graph, as partitioning queries reveals relationships that were obscured by the monolithic treatment of queries in the regular click graph.

In the following, we define separate website similarity graphs based on the similarity between queries, entities, modifiers or the individual keywords that led to a click on two websites. Formally, we will represent each website using the vectors $V_{query}(s)$, $V_{word}(s)$, $V_{entity}(s)$ and $V_{modifier}(s)$ in the four different vector spaces spanned by queries, keywords, entities and modifiers, respectively. In each case, we apply a *tf-idf* weighting to the coordinates instead of using simple counts as in the example. Then we compute the similarities between each pair of vectors in each of these spaces, e.g. $Sim_{query}(s_i, s_j) = cos(V_{query}(s_i), V_{query}(s_j))$

Based on the similarities, we define four website similarity graphs SG_{query}, SG_{entity}, $SG_{modifier}$, SG_{word} where the nodes represent websites and edges represent similarities based on similarity of queries, keywords, entities and modifiers, respectively, weighted by the corresponding similarity measure (e.g. Sim_{query} for SG_{query}).

Since SG_{entity} and $SG_{modifier}$ contain only partial information from each query-click pair, we also created similarity graphs containing the union of nodes and edges from entities and modifiers. We argue that for the task of website similarity, if we knew that a website focused on certain entities, it would be more important to find other websites focusing on the same entity, relegating the modifiers to a more auxiliary role. Conversely, for service-focused websites it would be more important to give more weight to modifiers.

For websites that have been very entity-specific in the query logs, the fact that two websites share the same entity is more perplexing (therefore more informative). Thus, while building a union of SG_{entity} and $SG_{modifier}$, we can use a measure of perplexity to weigh individual entity/modifier components of a query. In Information Theory, *perplexity* is defined in terms of entropy as 2^H. Thus, we model the dynamics between the query parts (entity, modifier) by the ratio between their perplexities: $pRatio_e(s) = \frac{2^{-H_e(s)}}{2^{-H_e(s)}+2^{-H_m(s)}}$. Similarly, we define $pRatio_m$ by substituting the numerator

by $2^{-H_m(s)}$. A higher $pRatio_e$ results in a lower $pRatio_m$ and vice versa.

We then generate SG_{ratio} where each website is represented by a vector resulting from the union of V_e and V_m after applying a scalar multiplication of each by the corresponding $pRatio(s)$ weight, resulting in: $V_{ratio}(s) = V_e(s) \times pRatio_e(s) \cup V_m \times pRatio_m(s)$.

For comparison, we also tested a simple union of SG_{entity} and $SG_{modifier}$, which we reference as SG_{union}.

4. EVALUATION

For our experiments, we created a query log dataset by sampling 45,815,323 successful query sessions from the January 2009 query logs of of Yahoo! Search. We have considered a query session successful when the session has ended with a click [7]. We obtained a list of entities from Freebase data from May 2009 containing 5,600,250 named entities. We created our dictionary E from all entities provided, including both the high-quality curated entities and the user-defined ones, which tend to be less trustworthy.

From our query log, we have collected query hits consisting of the last query and the clicked URL. We then parsed the query keywords and collected every query hit that contained one of the entities in E followed or preceded by a modifier. When faced with ambiguity, we pick the most frequent entity. The most frequent sense is a competitive baseline for word sense disambiguation, hard to beat in non-specialized domains by far more sophisticated approaches [15]. We indexed E in main memory and performed the extraction process in a distributed fashion using Hadoop.[3] After the extraction step we were left with 6,703,821 query hits where the query included at least one entity.

We used Pig over Hadoop to generate counts and simple statistics over this dataset. Although computing similarities between all pairs of click-graph nodes scales quadratically in complexity, we have used a similarity engine similar to Bayardo et al. [3] to speed up the computation. Distribution over a cluster allows us to scale linearly with the number of computers, allowing Web scale analysis.

Baselines. Based on the related work, we have two baselines for our evaluation. We compute website similarities by treating entire queries as units (SG_{query}), or breaking up queries into individual query words (SG_{word}).

We also compare our results with a third baseline based on social tagging metadata. User provided tags from Delicious, for example, have been already shown to result in 8% better accuracy in classifying pages against the Open Directory Project (ODP) taxonomy than representations based on the HTML content of Web pages [22]. Compared to our query log data, Delicious is also a strong baseline in that many websites have attracted significant tagging data over the many years of the existence of service, while our query log data covers only one month of activity.

Using the Yahoo! BOSS API[4] we retrieved the top public Delicious tags $T(s)$ for a site and the counts associated with each tag. Hence, we define $Sim_{delicious}(s_i, s_j)$ as the Jaccard coefficient between the tagsets of s_i and s_j in Delicious: $Sim_{delicious} = |T(s_i) \cap T(s_j)|/|T(s_i) \cup T(s_j)|$.

Gold standard. We used the ODP taxonomy as a gold standard, given that it has been extensively employed in evaluations [2, 9, 4]. ODP can provide a grounding to measure website relatedness: websites that are in the same ODP categories are considered related. We use ODP to judge if each predicted edge in our similarity graphs connects related websites.

Let S_{odp} be the set of websites from ODP that have been clicked more than 10 times in January 2009 in our logs. We will note as $C(s)$, $\forall s \in S_{odp}$, the set of categories for each website s in the gold standard (ODP), and $Rel_{odp}(s_i, s_j)$ the function that assign a relevance score according to the gold standard. One natural choice would be to assign 1 if $\exists c \in C(s_i) \cap C(s_j)$, i.e. the two sites share a category in ODP and 0 otherwise. However, such a measure ignores the taxonomy nature of ODP. Since ODP categories are defined as paths (e.g "`Regional/Europe/Spain`"), Baeza and Tiberi defined an ODP similarity function as the length of the common prefix between the categories of c_i and c_j divided by the longest path [2]. We considered two websites related if their categories overlapped by at least 2/3.

Results. We evaluate one website similarity graph at a time. For each website s_i in S_{odp}, we collect all edges (s_i, s_j); $s_i, s_j \in S_{odp}$ from the similarity graph, and assign a score with regard to the gold standard Rel_{odp}.

Graph	*P@5*	*Avg(|E|)*
SG_{query}	0.4380	1.75
SG_{entity}	0.3140	1.94
$SG_{modifier}$	0.4500	2.01
SG_{word}	0.3840	2.26
SG_{ratio}	**0.4640**	**3.22**

Table 1: Precision at 5 ($P@5$) results and average number of edges returned $Avg(|E|)$ for each similarity graph.

We use P@n as defined by Deng et al. for query similarity evaluation [9] and apply it to our website similarity evaluation, so that: $P@n = \sum_{i=1}^{n} Rel_{odp}(s_i, s_j)/n$. We compute P@5 and average the values over all websites. This assesses the performance on our main intended task, i.e. suggesting a small number of similar websites to show to users in an online scenario.

Table 1 shows the performance of each similarity function. The SG_{ratio} graph provides a statistically significant improvement over the SG_{query} and SG_{word} baselines. The SG_{ratio} graph is also a significant improvement over its components SG_{entity} and $SG_{modifier}$. Significance was tested with the Wilcoxon Matched-pair Signed-Ranks Test.

Table 2 focuses on the strongest similarity edges identified for each similarity graph ($Sim > 0.9$). The table shows the percentage of edges returned that were correct (P) and on average how similar were the categories of the websites connected by those edges ($Avg(Sim(s_i, s_j))$). The difference between the two measures is that the latter also captures partial matches. The results show that SG_{ratio} returns less exact matches, i.e. pairs of sites whose categories are exactly the same, but shows higher overall quality of the returned results based on partial matching.

E	*Avg(Sim(s_i, s_j))*	*P*
SG_{entity}	0.9141	0.5556
$SG_{modifier}$	0.8581	0.6206
SG_{query}	0.9497	0.6562
SG_{word}	0.9505	**0.7533**
SG_{ratio}	**0.9600**	0.6190
SG_{union}	0.9528	**0.7500**
$SG_{delicious}$	0.7399	0.3994

Table 2: Precision of top correctly identified edges (Sim>0.9).

[3] `http://hadoop.apache.org`

[4] `http://developer.yahoo.com/search/boss/`

Recall was less than 0.05% for all networks. We consider this a natural result, as edge recall is limited by the query log data, i.e. it only measures the proportion of ODP we cover with a sample of one month of query log data. We note that, our objective is not to reconstruct ODP. Instead, we focus on finding, *amongst websites being searched by users*, those that are similar to each other. ODP here figures solely as a previously assigned, manually generated judgment of each website similarity edge.

Discussion. There are two major ways in which a correct similarity edge was not counted in this study. Many edges were found by our method but there was no classification available for (at least one of) the websites. We call this the 'unclassified websites' problem. For those unclassified sites we could expect the precision to behave as in the subset of hosts that is in ODP.

Furthermore, since ODP is a manually created resource, it is a known fact that its classification is not exhaustive: users may use different category names, and possibly fail to include a website into a relevant category. Some websites were found by our methods but not computed as correct since they were not in overlapping categories in ODP. We call this the 'insufficient classification' problem.

The separation in entity and modifier can cause over specialization in some graphs. In SG_{entity}, for `lufthansa-usa.com` there is only an association with the website `lufthansa.com` with a similarity greater than 0.9, and these sites are associated because both are highly related to the entity "Lufthansa" (and that is not the case for any other site). In the $SG_{modifier}$ graph the site `lufthansa-usa.com` is similar to 46 other sites with a similarity greater than 0.9 and only one these sites are not an actual airline (but a travel site). In the SG_{word} graph there is only one related site (`lufthansa.com`) with a similarity greater than 0.9, and there are only 7 sites with a similarity greater than 0.3 (six airline sites and one travel site).

As expected, Sim_{word} yielded mistakes related to the distortion of semantics in queries. For example, for the websites `all-about-halloween.com` and `catlitterboxes.net` we observe $Sim_{word} = 0.78$, since there were queries searching for recipes of "Cat Litter Cake", a type of cake commonly baked for Halloween. Breaking up that entity into words will make it seem closely related to websites that are focused on pets.

5. CONCLUSION

In this paper we have proposed an entity-aware representing of query log data by interpreting search queries as consisting of an entity and modifiers. The method is scalable to the entire Web. It is an unsupervised approach of query log interpretation, where computation can be done offline, distributed over a Hadoop cluster. We showed that this representation alleviates the problems of query sparsity in comparison to other models of query log data, while preserving enough of the semantics of queries to be useful in applications. We showed that even taken in isolation, entities and modifiers are good predictors of website similarity. Moreover, the entity-aware models offer an insight into why those websites are related, namely provision of similar content or similar services. This kind of analysis can be useful to several applications including query intent detection, search result diversification, vertical selection, amongst others.

By recognizing named entities originating from the Semantic Web, in the future we will also be able exploit the fact that background knowledge not only provides names of entities, but also provides the types of entities as well as the relationships among entities. As an example, moving from entities to types will allow characterizing websites at higher level of abstractions while the inter-relationships of entities could be exploited to improve measures of query similarity.

6. REFERENCES

[1] R. Baeza-Yates. Relating content through web usage. In *HT '09*, 2009.

[2] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *KDD '07*, 2007.

[3] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *WWW '07*, 2007.

[4] S. M. Beitzel, E. C. Jensen, D. D. Lewis, A. Chowdhury, and O. Frieder. Automatic classification of web queries using very large unlabeled query logs. *ACM Trans. Inf. Syst.*, 25(2):9, 2007.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *CIKM'08*, 2008.

[6] A. Broder. A taxonomy of web search. *SIGIR'10*.

[7] D. Ciemiewicz, T. Kanungo, A. Laxminarayan, and M. Stone. On the use of long dwell time clicks for measuring user satisfaction with application to web summarization. *Yahoo Research! Tech. Report*, 2010.

[8] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07*, 2007.

[9] H. Deng, I. King, and M. R. Lyu. Entropy-biased models for query representation on the click graph. In *SIGIR '09*, 2009.

[10] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *SIGIR '09*, 2009.

[11] J. Hu, G. Wang, F. Lochovsky, J.-t. Sun, and Z. Chen. Understanding user's query intent with wikipedia. In *WWW '09*, 2009.

[12] B. J. Jansen, D. L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Mgmt.*, (3), 2008.

[13] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *SIGIR '08*, 2008.

[14] X. Li, Y.-Y. Wang, and A. Acero. Extracting structured information from user queries with semi-supervised conditional random fields. In *SIGIR '09*, 2009.

[15] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. Unsupervised acquisition of predominant word senses. *Comput. Linguist.*, 33:553–590, 2007.

[16] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *CIKM '08*, 2008.

[17] Nadeau, David, Sekine, and Satoshi. A survey of named entity recognition and classification. *Linguisticae Investigationes*, (1), 2007.

[18] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In *CIKM '07*, 2007.

[19] B. Poblete, C. Castillo, and A. Gionis. Dr. searcher and mr. browser: a unified hyperlink-click graph. In *CIKM '08*, 2008.

[20] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *WWW'10*, 2010.

[21] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW'04*, 2004.

[22] A. Zubiaga, R. Martínez, and V. Fresno. Getting the most out of social annotations for web page classification. In *DocEng'09*, 2009.