# Search Query Disambiguation from Short Sessions

Lilyana Mihalkova and Raymond Mooney
The University Of Texas at Austin, Austin, TX
{lilyanam,mooney}@cs.utexas.edu

## 1. Introduction

Personalizing a user's web search experience has become a vibrant area of research in recent years. One of the most actively researched topics in this area is web query disambiguation, or automatically determining the intentions and goals of a user who enters an ambiguous query. This is not surprising, given the frequency of ambiguous searches and the unwillingness of users to enter long and descriptive queries. For example, Sanderson [9] reports that anywhere between roughly 7% and 23% of the queries frequently occurring in the logs of two search engines are ambiguous, with the average length of ambiguous queries being close to one.

Ambiguity exists not only in cases such as the all-too-familiar "jaguar" example, but also in searches that do not appear ambiguous on the surface. Frequently queries that are commonly considered unambiguous become ambiguous as a result of the wealth of Web sources, which examine different aspects of a given topic. For example, a search for "Ireland" may be prompted by at least two different kinds of intentions—either by a desire to visit the country as a tourist, in which case one is interested in hotels and tourist sites, or by the need to complete a geography essay, in which situation objective facts about the country are more appropriate.

Most approaches to web query disambiguation leverage a user's previous interactions with the search engine to predict her intentions when entering an ambiguous query. Typically, user actions over long periods of time are logged, e.g.,[10, 11, 3]. Approaches that require long search histories may raise privacy concerns and may be difficult to implement for pragmatic reasons. After the release of AOL query log data allowed journalists to identify one user based on her searches [1], many people have become especially wary of having their entire search histories recorded by search engines. This has led to increased interest in the ethical issues surrounding user data collection, e.g., [2], and the appearance of search engines that expressly do not store any user activity information, such as Cuil (`http://http://www.cuil.com/`).

However, in order to determine user intent when typing an ambiguous query, at least some information must be available about the user. We present an approach that bases its predictions only on a short glimpse of user search activity, captured in a brief search session. Our approach relates the search session of the current user to previous short sessions of *other* users based on the search activity in these sessions. Because sessions do not record user identifiers, it is impossible to find previous searches by the same user and thus impossible to reconstruct whole search histories.

Our proposed approach is appealing also from a pragmatic standpoint because it does not require search engines to store, manage, and protect long user histories, thus decreasing the amount of processing that needs to be performed on the server side and avoids the difficulty of identifying users across search sessions.

When so little is known about a searcher, the problem of query disambiguation becomes very challenging. In fact, in previous research it has been argued that session-only information is too sparse to allow for any meaningful prediction [3]. In this work, we present evidence to the contrary by developing an approach that successfully leverages this small amount of information about a user to improve the ranking of the returned search results. Our approach is based on statistical relational learning (SRL) [4] and exploits the relations between the current session in which the ambiguous query is issued and previous sessions.

SRL, which addresses the problem of learning models that support probabilistic reasoning about data that involves multiple entities connected by a variety of relationships, is an appealing approach for our problem for two main reasons. First, the data is inherently relational—there are several types of objects: queries, clicked URLs, and sessions, which relate to each other in a variety of ways, e.g., two sessions may be related by virtue of containing clicks to the same URLs or searches for similar queries, queries may be related by sharing words or by being followed by clicks to the same URLs, and so on. SRL techniques allow us to learn *general* models of the ways in which the various types of entities interact, thus overcoming the problem that not much may be known about any particular entity, i.e. a particular URL. Second, data of human interactions with a search engine is likely to be noisy. Because SRL models allow for probabilistic inference, they can be successfully used to reason from noisy data.

We used one particular SRL representation, Markov logic networks (MLNs) [8]. A Markov logic network consists of a set of weighted first-order clauses and defines a Markov network when provided with a set of constants. The probability of a world decreases exponentially in the weight of clauses that are not satisfied by it. We chose MLNs because of their generality, their successful application to other language-related tasks, e.g., [7], and the availability of a well-maintained code base [5].

## 2. Data

We used data provided by Microsoft Research containing anonymized query log records collected from MSN Search in May 2006. The data consists of timestamped records for individual short sessions, the queries issued in them, the URLs clicked for each query, the number of results available for each query and the position of each result. Although some of the sessions may belong to the same users, the data excludes this information. This dataset therefore perfectly mirrors the scenario of disambiguating user intent from short interactions that we address in this research. Because there is a one-to-one correspondence between users and sessions, we will use these two terms interchangeably.

The data has two main limitations. First, it does not state which search queries are ambiguous. We employed a simple heuristic to obtain a (possibly noisy) set of ambiguous queries, using DMOZ (`www.dmoz.org`): a query string is considered ambiguous if, over all URLs clicked after searching for this exact string, at least two fall in different top-level categories, according to the DMOZ hierarchy. We limited ourselves to strings containing up to two words, thus obtaining 6360 distinct ambiguous query strings. Second, the data does not list all the URLs presented to the user after a search but just the clicked ones. This presents a difficulty during testing. To overcome this, we assumed that the set of all URLs clicked after searching for a particular ambiguous query string, over the entire dataset, was the set of results presented to the user.

We used the first 25 days of data for training and validation and the last 6 for testing. Sessions stretching across these time periods were discarded. As training/testing examples, we used sessions that contained an ambiguous query, temporally preceded by clicks to at least 5 distinct hostnames. As a result, in the training sessions, the

average number of distinct previous clicks was 8.02 and the average number of previous searches was 5.76. Distinct ambiguous searches in the same session are treated as separate examples. During testing, *only* the information regarding user activity preceding the ambiguous query is provided as evidence. The set of hostnames of the possible results for this ambiguous query string is given, and the goal is to rank them based on how likely it is that they represent the intent of the user. In cases when the user clicks on more than one result after searching for a string, we accept all as relevant.

# 3. Query Disambiguation with MLNs

Our general approach follows that of previous applications of MLNs to specific problems, e.g., [7]: we hand-coded the structure of the model as a set of first-order clauses and learned weights on this structure from the data. For weight learning we used the discriminative perceptron-like contrastive divergence weight learner [6]. Because of the large amount of data, which would not fit in memory, we adapted this algorithm to proceed in an on-line fashion—at each step, only a single training session and its relevant background information are presented to the learner and a single weight update is carried out for each clause.

We defined the following predicates (in the descriptions, `as` refers to the current active session and `aq` to the current ambiguous query):

| | |
|---|---|
| `possR(r)`: | `r` is a possible result for `aq` |
| `srchAndCl(s,r)`: | session `s` clicked on `r` after search for `aq` |
| `willCl(r)`: | `as` will click on result `r` |
| `connViaCl(s,d)`: | sessions `s` and `as` are related via shared click to `d` |
| `connViaCC(s,k)`: | sessions `s` and `as` are related via a keyword `k` shared between a click in `as` and a click in `s` |
| `connViaCS(s,k)`: | sessions `s` and `as` are related via a keyword `k` shared between a click in `as` and a search in `s` |
| `connViaSC(s,k)`: | sessions `s` and `as` are related via a keyword `k` shared between a search in `as` and a click in `s` |
| `connViaSS(s,k)`: | sessions `s` and `as` are related via a keyword `k` shared between a search in `as` and a search in `s` |

The goal is to predict the `willCl(r)` predicate, given as evidence the values of the remaining ones. We used the following clauses:

$\exists$`r willCl(r)`.

`possR(r)` $\wedge$ `connViaCl(s,d)` $\wedge$ `srchAndCl(s,r)` $\Rightarrow$ `willCl(r)`

`possR(r)` $\wedge$ `connViaCC(s,k)` $\wedge$ `srchAndCl(s,r)` $\Rightarrow$ `willCl(r)`

`possR(r)` $\wedge$ `connViaCS(s,k)` $\wedge$ `srchAndCl(s,r)` $\Rightarrow$ `willCl(r)`

`possR(r)` $\wedge$ `connViaSC(s,k)` $\wedge$ `srchAndCl(s,r)` $\Rightarrow$ `willCl(r)`

`possR(r)` $\wedge$ `connViaSS(s,k)` $\wedge$ `srchAndCl(s,r)` $\Rightarrow$ `willCl(r)`

The first clause states that at least one of the results will be clicked and is hard, i.e. it is always required to hold. The remaining clauses, for which we learn weights, state that the user will choose the result chosen by background sessions that are related to it via one of the `connVia` predicates. The more clicks or keywords the current session shares with background sessions in which a particular possible result was clicked, the more likely it is, according to the model, that this result will be clicked in the current session. Thus, this is an entirely collaborative-based model. We also considered a mixed collaborative-content-based approach, in which we additionally used the keywords appearing in previous searches and hostnames in the *current* session to provide evidence about which possible result may be chosen. However, we found that this extra information did not lead to an improvement.

# 4. Preliminary Results and Ongoing Work

We compared two MLNs to three baselines. **MLN1** contains the first two clauses described above. **MLN2** contains all clauses. The first baseline, **Random**, randomly ranks the possible results. The second baseline, **Click-Sim** uses the same information as **MLN1** as follows. Given an ambiguous query $Q$, issued in a session $S$, and a set of possible results $\mathcal{R}_Q$, it assigns to each $R \in \mathcal{R}_Q$ a rank that equals the average similarity of $S$ to all background sessions that clicked on $R$ after searching for $Q$. The similarity between two sessions equals the number of hostnames (of the clicks) that the two sessions have in common. The third baseline, **Click-KW-Sim** uses the same information as **MLN2** in an analogous way, except that the similarity between two sessions equals the number of hostnames *and keywords* they share.

We used two metrics, the Mean Average Precision (MAP), which is identical to the area under the precision-recall curve, and the area under the ROC curve (AUC-ROC). If the user starts scanning the page of returned results from the top, the AUC-ROC intuitively represents what percentage of the irrelevant results were *not* seen by the user. Thus, a random ranker would obtain an AUC-ROC of 0.5. Unlike MAP, AUC-ROC is sensitive to the number of possible results that are to be ranked and thus, more informative in our case where we have differing numbers of possible results for each ambiguous query. The results are listed in the following table.

| | **Random** | **Cl-Sim** | **MLN1** | **Cl-KW-Sim** | **MLN2** |
|---|---|---|---|---|---|
| **MAP** | 0.304 | 0.329 | 0.348 | 0.332 | 0.364 |
| **AUC-ROC** | 0.504 | 0.526 | 0.544 | 0.534 | 0.567 |

Although the differences may appear small, most of them are statistically significant. All significance claims are at the 99.5 level or better (i.e. p-value $\leq$ 0.005). All approaches give significant advantages over **Random**. The other statistically significant results are as follows: **MLN2** is better than all others; **MLN1** is better than **Cl-Sim** on both measures, and better than **Cl-KW-Sim** on MAP; **Cl-KW-Sim** is better than **Cl-Sim**. However, **Cl-KW-Sim** is not as successful as **MLN2** at taking advantage of the additional information that is provided to it. One reason for this is that **Cl-KW-Sim** considers all ways in which two clauses may be related to be equally important, whereas the learned weights on the MLN determine the relative importance of different types of relations. These initial results demonstrate that the use of MLNs allows for better rankings than hand-coded baselines that use roughly the same information. A second advantage of the MLN approach is that using already existing general technology for learning and inference with MLNs we were able to obtain competitive performance after hand-coding only a few simple rules. In on-going work, we are experimenting with more complex models.

# Acknowledgment

This research is supported by a gift from Microsoft Research. Some of the experiments were run on the Mastodon Cluster, provided by NSF Grant EIA-0303609, at The University of Texas at Austin.

# 5. References

[1] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. New York Times, August 2006. http://www.nytimes.com/2006/08/09/technology/09aol.html?ex=1312776000.

[2] G. Conti. Googling considered harmful. In *New Security Paradigms Workshop*, Dagstuhl, Germany, 2006.

[3] Z. Dou, R. Song, and J. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW-2007*.

[4] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2007.

[5] S. Kok, P. Singla, M. Richardson, and P. Domingos. The Alchemy system for statistical relational AI. Technical report, Dept. of Comp. Sci. and Eng., Univ. of Washington, 2005. `http://www.cs.washington.edu/ai/alchemy`.

[6] D. Lowd and P. Domingos. Efficient weight learning for Markov logic networks. In *PKDD-07*.

[7] H. Poon and P. Domingos. Joint inference in information extraction. In *AAAI-07*.

[8] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[9] M. Sanderson. Ambiguous queries: Test collections need more sense. In *SIGIR-08*.

[10] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW-2004*.

[11] J. Sun, H. Zeng, H. Liu, Y. Lu, and Z. Chen. CubeSVD: A novel approach to personalized web search. In *WWW-2005*.