Web Usage Mining with Semantic Analysis

Laura Hollink[*]
Delft University of Technology/
VU University Amsterdam
The Netherlands
l.hollink@vu.nl

Peter Mika
Yahoo! Research
Diagonal 177
Barcelona, Spain
pmika@yahoo-inc.com

Roi Blanco
Yahoo! Research
Diagonal 177
Barcelona, Spain
roi@yahoo-inc.com

ABSTRACT

Web usage mining has traditionally focused on the individual queries or query words leading to a web site or web page visit, mining patterns in such data. In our work, we aim to characterize websites in terms of the semantics of the queries that lead to them by linking queries to large knowledge bases on the Web. We demonstrate how to exploit such links for more effective pattern mining on query log data. We also show how such patterns can be used to qualitatively describe the differences between competing websites in the same domain and to quantitatively predict website abandonment.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval; I.2 [**ARTIFICIAL INTELLIGENCE**]: Knowledge Representation Formalisms and Methods

General Terms

Measurement, Experimentation

Keywords

Query log, semantic analysis, query session

1. INTRODUCTION

Understanding the information needs of Web users is a crucial task for both search engine providers and site owners on the Web. In search engine development, such insights are used for developing specific tools for common information needs, e.g. in the form of tailored information boxes that show up for specific types of queries (e.g., verticals). Similarly, content publishers are interested in understanding user needs in order to select and structure the content of their properties.

Though user needs may be elicited in other ways, usage mining of Web logs is a widely used alternative for understanding usage patterns. To this end, search engines collect query logs and toolbar data, while content providers log information about search referrals, site search and browsing activity. In both cases, the information is typically aggregated into sessions, which group together the actions performed by the same user (typically identified using IP addresses or cookies) within a predefined window of time.

A key challenge in gaining any insight to usage patterns based on query logs is the notable sparsity of the event space. Baeza-Yates et al. [1] note that 64% percent of queries are unique within a year, and even though some of the variation could be attributed to misspellings, there are surprisingly many syntactic variations of the same information need. Further, it is well known that users with the same need may click on vastly different results, based on the presented ranking, their evaluation of the *perceived* relevance of the search results and user preferences in terms of known websites, types of information etc.

This causes problems for both data mining and machine learning. When looking at sequences of queries and clicks, as they appear in session data, even the most frequent patterns have extremely low support. As a result, these patterns may not be representative of the real use cases. In terms of machine learning, collecting labeled data becomes ineffective since the labeled examples do not generalize to unlabeled data due to the size of the feature space. The situation is even more critical in domains where queries and/or content frequently change. In this paper, we look at one such domain, i.e. movie search.

A site owner or search engine might collect data similar to the example in Figure 1. These are ten sessions that all reference the movie *Moneyball*, issued on June 29, 2011 by ten different users. We want to point out the variety of queries (”moneyball”, ”moneyball movie”, ”moneyball the movie”, ”brad pitt moneyball”), the different preferences for clicked websites for the same query —IMDb, Wikipedia, and Yahoo! Movies all provide basic information—, and that the related queries are interspersed with other searches, making it difficult to see patterns. Still, we can discern that all the above users are essentially performing the same task, i.e. gathering information about this movie. The overall task can be broken down into different activities, such as finding information about the main actor (Brad Pitt), the topic of the movie (the Oakland A’s basketball team), the main characters (Billy Beane, Peter Brand). Further, the users are looking for basic reference information, as well as trailers and news.

Given the above data, we might want to know whether these activities are typical for all movies, or just recently released ones, or typical only for *Moneyball*. We also want to find if there is an order in which users prefer to carry out

[*] Work done partly while visiting Yahoo! Research.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

oakland as bradd pitt movie moneyball movies.yahoo.com oakland as wikipedia.org

captain america movies.yahoo.com moneyball trailer movies.yahoo.com

money moneyball movies.yahoo.com

moneyball movies.yahoo.com movies.yahoo.com en.wikipedia.org movies.yahoo.com peter brand peter brand oakland nymag.com moneyball the movie www.imdb.com

moneyball trailer movies.yahoo.com moneyball trailer

brad pitt brad pitt moneyball brad pitt moneyball movie brad pitt moneyball brad pitt moneyball oscar www.imdb.com

relay for life calvert ocunty www.relayforlife.org trailer for moneyball movies.yahoo.com moneyball.movie-trailer.com

moneyball en.wikipedia.org movies.yahoo.com map of africa www.africaguide.com

money ball movie www.imdb.com money ball movie trailer moneyball.movie-trailer.com

brad pitt new www.zimbio.com www.usaweekend.com www.ivillage.com www.ivillage.com brad pitt news news.search.yahoo.com moneyball trailer moneyball trailer www.imdb.com www.imdb.com

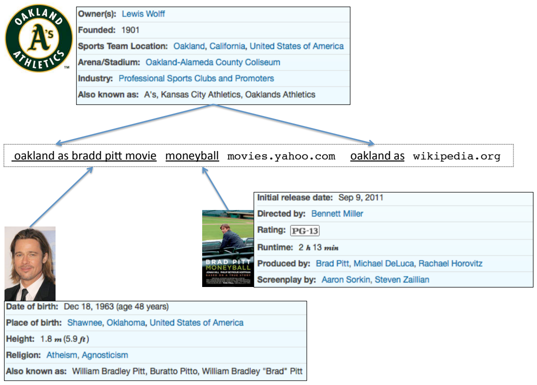
Figure 1: Sample of sessions containing the term "moneyball".

the task, e.g. whether they prefer in general to check out the cast of the movie before watching a trailer or the other way around, so that we can support the users in carrying out their task. Lastly, as the owners of one of the websites provisioning content, we might be interested to find out what are the patterns that lead to our site versus the competitor, information that we could use to improve our services.

In our work, we prove that the above questions can be effectively answered by a semantic analysis of query logs, and in particular by connecting query log data to the growing universe of public Linked Open Data (LOD). Datasets of the LOD cloud provide valuable background knowledge of the entities occurring in log data.

Figure 2 illustrates the basic idea. We annotate entity references in user sessions with references to LOD datasets such as Dbpedia[1] and Freebase[2]. This provides us with a wealth of information about the entities, e.g. that Moneyball was only to be released two months later on September 9[3], and that Brad Pitt was the star of the movie playing Oakland A's general manager Bill Beane.

Our paper is organized as follows. Following an overview of related work, in Section 3 we illustrate the challenges that current syntactic methods face when attempting to mine raw data without exploiting background knowledge. In Section 4 we describe our approach to exploiting large knowledge bases for semantic usage mining. Section 5, we detail our implementation and experimental evaluation. In particular, we apply (sequence) pattern mining that helps us to uncover patterns that generalize beyond a time period, set of users and particular movie. By comparing the access patterns of competing websites, we also qualitatively describe the differences in the services they provide. Last, we show how to reuse the same data and apply machine learning methods in order to predict website abandonment. To our knowledge, ours is the first paper to show the benefits of semantic annotations in addressing these problems.



Owner(s): Lewis Wolff
Founded: 1901
Sports Team Location: Oakland, California, United States of America
Arena/Stadium: Oakland-Alameda County Coliseum
Industry: Professional Sports Clubs and Promoters
Also known as: A's, Kansas City Athletics, Oaklands Athletics

oakland as bradd pitt movie moneyball movies.yahoo.com oakland as wikipedia.org

Initial release date: Sep 9, 2011
Directed by: Bennett Miller
Rating: PG-13
Runtime: 2 h 13 min
Produced by: Brad Pitt, Michael DeLuca, Rachael Horovitz
Screenplay by: Aaron Sorkin, Steven Zaillian

Date of birth: Dec 18, 1963 (age 48 years)
Place of birth: Shawnee, Oklahoma, United States of America
Height: 1.8 m (5.9 ft)
Religion: Atheism, Agnosticism
Also known as: William Bradley Pitt, Buratto Pitto, William Bradley "Brad" Pitt

Figure 2: Example of a session annotated with background knowledge from Linked Data.

2. RELATED WORK

Transaction-log analysis is a valuable and often used method to study user's search behavior on the Web. It is popular for being a non-intrusive way to study large amounts of usage data [17]. On the other hand, it does not provide information about the underlying information need of the searchers [24] or how satisfied they were with the result [19]. Jansen and Spink [18] provide an overview of Web-search transaction-log analyses.

In the last decade, several approaches have emerged that use semantics to aid the analysis of Web logs. For example, the recent USEWOD workshop series [4] is a forum for research on the combination of semantics and usage analysis. In this workshop, Hoxha et al. [15] presented work on linking Web resources to linked data entities to enable a semantic analysis of clicks in a search log. In Web retrieval, semantic information such as entities recognized in the queries, along with their types, can be valuable signals for ranking if they are appropriately incorporated into retrieval models [6]. Hofmann et al. [13] express the need for semantic enrichment of queries with annotations of language tags, named entities, and links to structured sources of knowledge. In 2004, Berendt et al. [5] had already argued in general for further integration of the fields of Semantic Web

[1] http://dbpedia.org

[2] http://freebase.com

[3] Sony Pictures released a trailer on June 21 to drum up interest in the movie, which is why it shows up in our data

and Web usage mining. Due to space restrictions, in the remainder of this section we will only discuss related work from this intersection of two fields, that is most relevant to our approach.

Linking queries to entities in structured sources of knowledge (i.e. linked data or offline thesauri) is the first step towards semantic analysis of query logs. Several linking approaches have been proposed: Huurnink et al.[16] utilize clicked results to link queries to an in-house thesaurus. Hollink et al [14] simply look for an exact match between a query and the *rdfs:label* of the entity, if necessary aided by the Porter stemmer. Mika et al. [21] use a similar approach: they match the largest substring common to the query and the label of the entity. Meij et al. [20] have developed an approach that combines language modeling for information retrieval and machine learning to map queries to DBpedia. They use features related to the words in the query, the concept to be matched and the search history. Our approach is different to these approaches as we rely on a Web search engine to select the best matching entity. We have deliberately chosen for an approach that does not require manual input (contrary to a machine learning approach).

In this paper, we use links from queries to entities to observe patterns in user behavior. Similar studies have been undertaken on specialized archives. Huurnink et al. [16] study the behavior of users of an audiovisual archive, and make a categorization of their query terms by linking query words to titles and thesaurus terms from clicked results. In [14] Hollink et al. enrich search queries of a commercial picture portal with linked data to determine the semantic types of the queries and the relations between queries in a session. They show that the semantic approach for analyzing query modifications forms a valuable addition to term-based approaches. Our approach builds on these approaches, but we analyze searches issued to a general purpose Web search engine. Even though we limit the scope by selecting sessions that contain movie-related clicks, the queries are typical Web queries in the sense that they are broad and short, and not by professional users.

We have specifically looked at longer patterns of sequences of entities found in a session. We have applied an out-of-the-box state-of-the-art method for frequent pattern mining, although other approaches exist that could be tailored to extracting meaningful sequences of queries within sessions [12]. Baeza-Yates and Tiberi [2] have used entities in queries to extract semantic relations between entities. Similarly, Benz et al. [3] have used logs to extract more complex semantic structures, so called *logsonomies*. Both, however, have not looked at longer patterns of entities in queries.

To the best of our knowledge, the use of links from queries to entities to detect navigational queries and to predict website abandonment is entirely novel.

3. LIMITATIONS OF SYNTACTIC MINING

Search engines and content providers collect similar usage information. Search logs contain information about user behavior while the user is interacting with the search engine, but not beyond, unless the search engine tracks the user through other means such as browser functionality or a toolbar. Publishers collect similar and complementary information in that they are typically aware of the search query that led to their site, and record additional searches and page views in their domain. The questions that are asked by the service providers are also largely common. The problems we address in this paper: (P1) How can we identify typical use cases in the data? (P2) What are the common user needs that are not satisfied, i.e. use cases that lead to abandonment of a service? (P3) Which other sites do (dis)satisfied users go to and in what cases?

The access logs commonly collected by Web servers can be formally represented as a sequence of events $< e_1, e_2, \ldots, e_n >$ where each event $e_i \in E$ is an atomic action taken by a client of the server. For each event, typically at least a timestamp, the user and the type of action are identified, i.e. $E = T \times U \times Z$, where T is the set of valid timestamps, U is the universe of users, and Z is the type of action. There are various ways in which client requests may be related to users but we do not concern ourselves with this problem here. In the following, we will consider only two classes of events, queries and clicks, i.e. $E = Q \cup C$. We also introduce the notion of an annotation function $\lambda : E \rightarrow L$ that assigns labels from a set L to the events. In access logs, we minimally record the queries and clicked URLs and we will represent this as the functions λ_{query}, λ_{url}, respectively.

Low-level access logs are aggregated into sessions $s =< e_1, ..., e_k >$ where $\forall e_i = (t_i, u_i, z_i), e_j = (t_j, u_j, z_j) : u_i = u_j, t_i < t_j$, by grouping the events by user and segmenting the sequences. The simplest way of segmentation is starting a new session after a predefined period of inactivity. Other methods exist to identify logical sessions, i.e. subsequences of events that likely belong to the same user task [7].

In all of the questions we would like to address, we are looking for frequent sequences of events in the data, which are frequent either because they are repeated in a large fraction of sessions, or alternatively, by a large fraction of users. A pattern may be a full session that is repeated often or a subsequence of events. Note that a (sub)sequence pattern may be also rewritten as an association rule, where the first $n-1$ events are the head of the rule, and the last event is the body of the rule. Formulated this way, problem P2 can be reduced to P1: the difference is that in P2 the user is looking for patterns that lead to the particular condition of abandonment. This may be the event that represents the user canceling a shopping process, abandoning a search, visiting a competitor site after looking at a website etc. It can be formalized by introducing a third type of event, the abandonment event ($E = Q \cup C \cup A$) and inserting this manually in the event sequence at the point where the abandonment condition is met. The problem P2 is then finding association rules with the abandonment as a consequence, i.e. patterns of the form $e_1, \ldots e_n$ where $e_n \in A$.

An alternative formulation of P2 is a decision problem where given a sequence prefix $e_1, \ldots e_{n-1}$ we are to predict whether the next event is an abandonment event or not, i.e. whether $e_n \in A$ or $e_n \notin A$. The problem P3 can be conceived then as a multi-class version of P2 where there are different types of abandonment events, in particular one for each competitor website. Both P2 and P3 can be addressed by supervised machine learning methods the learned model is used to predict abandonment, or to predict to which competitor website the user is going.

Our problems are thus straightforward to map to the problem of frequent subsequence mining by transforming session data to a sequence of labels generated by some annotation function. Methods for extracting frequent patterns are well known in the literature [12]. They produce equiva-

Rank	Session	Frequency
1	netflix `www.netflix.com` netflix `www.netflix.com`	5577
2	netflix login :netflix login member `www.netflix.com`	4283
3	netflix login `www.netflix.com` netflix login `www.netflix.com`	2368
4	netflix login `netflix-login.com` netflix login `www.netflix.com`	1989
5	netflix login netflix login member `www.netflix.com`	1865
...		
12	fandango movies times `www.fandango.com` fandango movies times `www.fandango.com`	866

Table 1: Most frequent sessions in a sample of 1 million sessions from the movie domain

lent results when it comes to finding patterns more frequent (having a higher *support*) than a given threshold. Frequent pattern mining methods, however, fail in a very apparent fashion when applied to the raw, syntactic data as modeled above, due to the sparsity of the event space. In the case of frequent sequence mining they fail to identify sequences that would cover much of the data, while in the case of learning decision trees failure shows in the form of models that are large and do not generalize (i.e. they over-fit the data) or do not substantially differ from the default rule. Given the sparsity of labels that are provided by the λ_{query} and λ_{url} functions, we want to define new annotation functions that generalize user events while preserving their intent. For click events, it is a logical step to generalize URLs in the data to URL patterns or hosts, especially if we are only interested in competitive analysis. This alone is not sufficient to address the sparsity problem.

We illustrate this problem on a dataset of movie-related sessions that we will introduce in more detail in Section 5.1. Table 1 shows the most frequent sessions ranked by support, i.e. the number of sessions. There are several issues at hand. The explanatory power of these patterns is very low: even the most common pattern explains only 0.5% of the data. Second, all the top patterns above rank 12 look very similar and in fact express the exact same user need: a user trying to log in to Netflix. As we will show later, logging in however is not the most important use case in this domain, only the one that users express most consistently in terms of issuing the same query and clicking the same result. The more common use cases do not surface as patterns due to the syntactic variety of queries and clicked URLs.

4. SEMANTIC USAGE MINING

To overcome the limitations of a syntactic analysis of usage data, we propose a semantic annotation of usage logs. We observe that a large number of queries in query logs refer to the same real world object. Recent studies showed that about 70% of Web search queries contain entities [11] and that the intent of about 40% of unique Web queries is to find a particular entity [23]. These queries either contain only the name of an entity (*brad pitt*) or just the entity and an additional term (a *modifier*) that defines the information need or action that the user would like to take (*brad pitt quotes*, *toy story download*) [23].

We make use of the prevalence of these queries in two ways. First, we note that the position of the modifier is largely irrelevant to the query intent, i.e. the query *quotes brad pitt* expresses the same need as *brad pitt quotes.* Also, the results differ only slightly: for our example query *brad pitt quotes*, 7 out of 10 results overlap in Yahoo! Search. The reason is that while search engines do take position information into account, the relative weight of this feature is rather low: in general, the order of query words tend to matter only within parts of queries or in the case of searching for phrases (quotes, lyrics, error messages), which are relatively rare compared to keyword queries. On the other hand, we also do not want to treat all queries as bags-of-words, given that the order of query words matter within named entities. We introduce a new annotation function λ_{dict} that labels queries with modifiers, i.e. the words that are not part of a named entity mentioned in the text.

The second idea we pursue is that we can link entity mentions in queries to semantic resources in the Linked Data cloud such as Dbpedia and Freebase. These knowledge bases provide permanent identifiers (URIs) for well-known entities and are rich sources of facts about the entities, i.e. their types (classes), attributes and relationships (properties). This will provide us the labeling function λ_{entity}. Once we have linked the entity in a query to its URI in one of these knowledge bases, we can generalize the intent of the query using either the types or attribute values of the instance. We will introduce the function λ_{type} to denote the function that maps query events to one or more types of entity referenced in the query. There are two problems that we address in detail in our experiments: the method for mapping from queries to entities and selecting the most appropriate type(s) from the ontology, given that the data source may contain many types for a given entity.

Using the semantic annotations of queries and sessions, we can also address another source of sparsity in session data, i.e. navigational queries. A navigational query is a query entered with the intention of navigating to a particular website [9]. For our analysis, we want to identify them, and treat them as a separate category. While this is common in any kind of query log analysis, in our case it is especially relevant: if we would not do this, an analysis of *types* of queries would falsely suggest a lot of user interest for databases (e.g. IMDb is of type database), or organizations (e.g. Fandango is of type organization). Further, as we have seen in the above examples, navigational queries such as *netflix login* are common, but hide the real user need: we learn that the user is trying to log in to a website, but do not learn what her intent is, and most likely we can not follow her to the private part of a website. Yet, navigational queries are frequent and easily surface as (parts of) patterns.

Identifying navigational queries is surprisingly tricky in practice. One common heuristic is to consider navigational queries where the query matches the domain name of a clicked result. This produces both false positives and false negatives. The heuristics works for the queries *netflix* or *netflix.com*, but the query *netflix login* does not match the

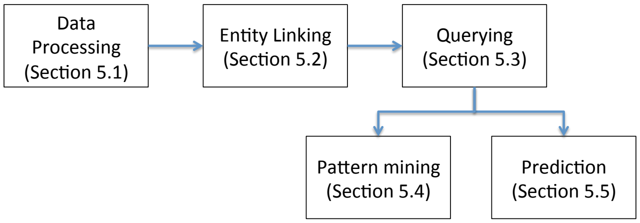

Data Processing (Section 5.1)
Entity Linking (Section 5.2)
Querying (Section 5.3)
Pattern mining (Section 5.4)
Prediction (Section 5.5)

Figure 3: Workflow for semantic query log mining.

URL of the top result `movies.netflix.com/WiHome`, i.e. the word login does not appear.

In the following, we will provide a new definition for navigational queries, which takes advantage of the semantic annotation of query logs.

Definition 1 (Navigational Query). *Given a query q that leads to a click c_w on webpage w, and given that q is linked to entity e, q is a 'navigational query' if the webpage w is an official homepage of the entity e.*

Lastly, a robust semantic annotation method can be used to address the problem of repeat queries and misspellings. For example, the user may first issue the query *moreen ohera* followed by *maureen o'hara*. A semantic annotation method should be able to reliable map both queries to the entity *Maureen_O'Hara* in Dbpedia and the Freebase resource *maureen_ohara*. This does not address typos in the modifiers (e.g. *michael keaton boi* repeated correctly as *michael keaton bio*), but spelling mistakes in modifiers hurt much less as modifiers are less sparse in general. In other words, there are many more cases where users issued the same query with the correct modifier.

5. METHODS AND EVALUATION

Our proposed workflow for semantic usage mining goes through the steps of data collection and processing, entity linking, filtering, pattern mining and learning (see Figure 3). In the following sections, we describe our method for each of these tasks and evaluate them on a real dataset collected from Yahoo! Search, a commercial search engine. We choose to perform our evaluation separately for each task because our methods may be reused in other contexts. For example, determining the main entity in a query is also a key task in triggering entity-based displays, for example for the related entity suggestions that currently appear in Yahoo! Search.

5.1 Data Processing

We have collected a sample of server logs of Yahoo! Search in the United States from June, 2011, where this service has been already powered by Bing at the time of data collection. The logs record time-stamped query and click events where the user is also identified by the means of browser cookies. We use the previously developed query-flow graph [7] for aggregating low level events into sessions as shown in Figure 1. Note that we do not use information about the user or the timing of queries and clicks.

For our experiments, we choose the movie domain as our application domain, because it is one of the most challenging ones in terms of the data, given that the information needs vary constantly based on new film releases and life events of celebrities. We limit the collected data to sessions about the movie domain by keeping only sessions that contain at least one visit to any of 16 popular movie sites[4].

In total, we have collected 1.7 million sessions, containing over 5.8 million queries and over 6.8 million clicks. The median session length is 6 queries and/or clicks, lower than the average session length of 7.1. The distribution of the session length, number of queries and clicks per session show a typical power law. Note that despite the initial filtering the data set contains clicks on a huge variety of websites, both movie-related and not movie-related, given that users often change topics during a session. We do not attempt to perform any logical segmentation of the sessions. We can expect that this noise is much less consistent than the signal and we will see that in fact it does not interfere with our analysis.

We also apply the filtering of navigational queries as proposed above. To collect the official homepages of entities, we query the combined Dbpedia and Freebase datasets for the values of `dbpedia:homepage`, `dbpedia:url`, and `foaf:homepage` for each entity of the type */organization/organization* in Freebase. In this way we identify 117.663 navigational queries, which makes it the 12th most frequent category of queries from all other semantic types (cf. Section 5.3.1). Thanks to our method, we are able to not only recognize frequent navigational intent queries (e.g. *netflix*) that could be trivially identified by other means such as matching the query string with the domain name of the clicked site. We also find frequent syntactic variations (e.g. *netflix login member*, *netflix login member queue*) that are not easily recognizable by other means. At the same time, we avoid false positives. For example, the query *banana* leads to clicks on the site `banana.com`. In this case, the query matches exactly the clicked domain, yet it is not a navigational query: banana.com is not the homepage of an organization with the same name. In fact, there are domain names for most common words in English and a click on these sites does not mean that the user wanted to navigate there instead of searching.

5.2 Entity Linking

In the following, we detail our methods for obtaining the functions λ_{entity}, λ_{type} and λ_{dict} that provide the entity, type and dictionary labels for queries in our query log.

5.2.1 *Linking Queries to Entities*

We link the queries to entities of two large semantic resources: Wikipedia and Freebase. To link queries to URI's of these websites, we take advantage of the strengths of Web search engines, i.e. ranking, handling of spelling variations and misspellings. We run each query in our dataset on a large commercial search engine, adding the "site:" operator to limit the results to the site we are interested in. We found Freebase to be more complete than Wikipedia regarding movie related-information, while still general enough to work also for disambiguating non-movie related queries. In our analysis in sections 5.3 and 5.4 we have therefore focussed on the structured information in Freebase. However,

[4] movies.yahoo.com, imdb.com, fandango.com, rottentomatoes.com, www.movies.com, trailers.apple.com, blockbuster.com, comingsoon.net, netflix.com, flixster.com, teaser-trailer.com, movies.about.com, themovieinsider.com, redbox.com, traileraddict.com, amcentertainment.com

since commercial search engines rely on PageRank [8] and click logs to rank search results, they are generally more reliable for frequently visited sites with a high PageRank score. We have therefore chosen to first link to a Wikipedia entity by adding "site:wikipedia.org" to the query. Freebase entities (called *topics*) have a wikipedia *key* that links to the corresponding Wikipedia URI, which makes the translation from a Wikipedia entity to a Freebase entity straightforward.

5.2.2 Linking Entities to Types

To allow for an analysis of queries on a higher level of abstraction, we use the Freebase API to get the semantic "types" of each query URI, i.e. the classes of which the query entity is a member. Freebase entities can be associated to several types. In our dataset, the number of Freebase types ranges from 1 to 100. To be able to compare and visualize which types of queries lead to websites we need a single type for each query URI. Freebase has an API for this purpose, called the 'notable types API'[5]. However, this API is not officially supported and the algorithm it uses to select one type for each entity (or *topic*) is only loosely explained. It is based on a set of heuristics such as the schema to which each type belongs and the frequency of occurrence of the type. Although the service seems to give generally good results, a quick investigation showed some strange cases, e.g. for the entity Arnold Schwarzenegger the type bodybuilder is chosen as the most notable, rather than the more intuitive types politician or actor.

For transparency, we have created our own set of dataset-specific and domain-specific heuristics to select one type for each entity:

1. disregard internal and administrative types, e.g. to denote which user is responsible
2. prefer schema information in established domains ('Commons') over user defined schemas ('Bases')
3. aggregate specific types into more general types
   - all specific types of location are a *location*
   - all specific types of award winners and nominees are an *award_winner_nominee*
4. always prefer the following list of movie related types over all other types: $/film/film$, $/fim/actor$, $/artist$, $/tv/tv\_program$, $/tv/tv\_actor$ and $/location$ (order of decreasing preference).

Heuristics 1 and 2 are data-set specific, in the sense that they relate to the Freebase data model. Heuristics 3 and 4 are movie-domain specific and would be different in other domains.

5.2.3 Dictionary Tagging

We also label queries with a dictionary created from the top hundred most frequent words that appear in the query log before or after entity names as modifiers. These terms are particularly interesting because they capture the intent of the user regarding the entity. The top twenty terms that appear in our dictionary are as follows:

- movie, movies, theater, cast, quotes, free, theaters, watch, 2011, new, tv, show, dvd, online, sex, video, cinema, trailer, list, theatre . . .

5.2.4 Evaluation

When evaluating links, two approaches are possible. (1) We can ask a rater to inspect each automatically created link and judge it according to some scale, e.g. correct or incorrect. This is comparable to approaches in information retrieval benchmarks like TREC, where raters are presented with $< query, document >$ pairs [25]. Measures for inter-rater agreement such as Cohen's κ or Cronbach's α can be used to quantify how well multiple raters agree on the judgements, and how appropriate the scale is. In our case, this would mean presenting a rater with a pair $< query, entity >$ and asking him/her whether this is the correct entity for the query. With large, broad, and irregularly structured resources like Wikipedia and Freebase, however, we cannot expect a rater to reliably judge whether or not a better entity exists for the query. We expect the results to have a (positive) bias, as several entities could be rated as correct because the rater is unaware of a better fitting concept. (2) Alternatively, we can provide a rater with the queries and ask him/her to manually create links to LOD concepts. This is comparable to the creation of a reference alignment in the field of Ontology Alignment. This approach does not have the bias of the first approach, but it is more time consuming. Also, the standard measures for inter-rater agreement cannot be directly applied as the number of categories that a rater can chose from is virtually unlimited (i.e. all entities in Wikipedia/Freebase).

We chose option (2) above, in which we compare manually created $< query, entity >$ and $< entity, type >$ pairs to automatically created links. We take two samples of queries - the 50 most frequently occurring queries and 50 random queries - and ask a human judge to manually create links between them and Wikipedia entities. The rater was free to search and browse Wikipedia in any way (s)he liked. Similarly, we take the 50 most frequent entities and 50 random entities and ask the rater to manually select the most fitting Freebase type from all types associated to the entity.

Table 2 shows the results of both evaluations. In 73% of the cases, the human judge agreed with the detected links between queries and entities. An inspection of the cases where the judge did not agree showed that queries are often (11% of the cases) ambiguous and both the manually and the automatically selected entity could be correct. Examples are the query 'Green Lantern', which could be the movie or the fictional character, and 'one week movie' which could be the movie from 2008 or original one from 1920. In 3 cases the human judge seemed to have made a mistake. In 13 cases the automatically created link was wrong. The latter happened mostly in situations where there exists no entity to represent the complete query. For example, the query *charlie brown shapes in the sky* was incorrectly linked to the Wikipedia page List_of_The_Charlie_Brown_and_Snoopy_Show_episodes, and *i cant get playstation network to reset my password* to PlayStation_Network_outage. Our results are slightly lower than those of other approaches such as Meij et al.[20] and Huurnink et al. [16], who report precision up to 88 and 90% respectively. However, Meij et al. use training data and Huurnink et al. use clicked results. Comparable approaches in Mika et al.[21] and Hollink et al. [14] do not provide a direct evaluation of their linking algorithm.

For the links between entities and types, the human judge agreed with the automatically selected types in 81 % of the cases. The selection of the most notable type from the list of

[5] http://wiki.freebase.com/wiki/Notable_types_API

	Query to Entity	Entity to Type
Correct	73%	81%
Ambiguous	11%	13%
Human error	3%	0%
System error	13%	6%

Table 2: Precision of chosen entities and types

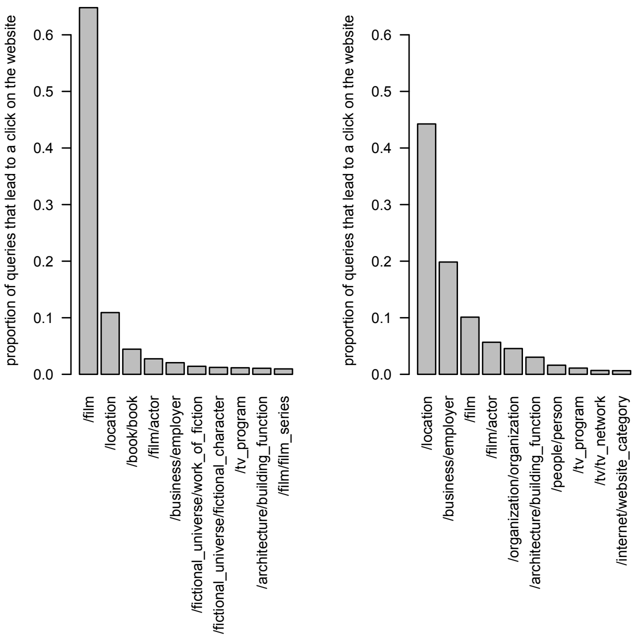


Figure 4: Top 10 most frequent query types that lead to a click on two websites

Freebase types associated to an entity is highly ambiguous. Many of the entities appear as book, film and tv series, or as film actor and tv actor. We identified 6 cases where the type selection was actually wrong. Examples are Oil_peak as a Serious_game_subject and Hillary_clinton as a Tv_actor. We found no significant differences between random and frequent queries/entities.

5.3 Semantic Pattern Mining

In this section we perform an analysis of queries at the level of individual queries and sessions, both using the entity and type information.

5.3.1 Single-query patterns

We compare queries that lead to a click on two different movie-related websites, namely those that occur most frequently in our dataset. For commercial reasons we will not disclose their domain names. Figure 4 shows the types of queries that users performed before they clicked on the two sites. This allows a qualitative comparison of the focus of the websites. Website 1 appears to be very specialized, with 70% of its queries being for movies. Website 2 attracts a broader range of queries, including TV related entities and entities of the general type Person. Business/Employer is the type chosen for any company, business or organization that employs people.

Level	Pattern	Support
L1	$movie$	0.396
	2011	0.15
	$trailer$	0.103
	$movies$	0.095
	$moviedict : 2011$	0.063
	$cast$	0.053
	new	0.049
	dvd	0.035
	$release$	0.032
	$reviews$	0.028
L2	$movie \longrightarrow movie$	0.165
	$2011 \longrightarrow 2011$	0.042
	$movie \longrightarrow 2011$	0.04
	$2011 \longrightarrow movie$	0.038
	$movies \longrightarrow movies$	0.038
	$trailer \longrightarrow trailer$	0.027
	$movie \longrightarrow movie\ 2011$	0.026
	$movies \longrightarrow movie$	0.025
	$movie \longrightarrow trailer$	0.024
	$movie\ 2011 \longrightarrow movie$	0.023

Table 4: Frequent patterns for recent movies (λ_{dict}).

5.3.2 Multi-query patterns

Sequential pattern mining is the problem of finding frequent patterns of labels in sequences of data items (see [12] for a survey). The problem of sequential pattern mining is defined in a way that each item may be labeled by multiple labels. The algorithms for solving this problem limit the search space by looking for patterns that are more frequent than a given threshold and return all such patterns with the computed support. In our work we use the PrefixSpan algorithm [22] and its implementation in the open source SPMF toolkit.[6]

Table 3 shows the top patterns of length 1, 2 and 3 according to λ_{type} that are present in more than 1% of the sessions. The patterns are particularly revealing in terms of the use cases in this domain. By looking at patterns of length one (L1), we can see that the two major use cases are searching for actors and films, remotely followed by other types of activity such as searching for show times, quotes etc. Comparing this with patterns of length two and three (L2, L3), we see the same use cases. In fact, both film and actor queries are frequently part of longer sequences of length two and three, in fact a similar percentage (38%) of the film and actor queries are followed by another film or actor query in the same session.

This analysis alone does not give an indication whether the users are looking for the same or different actors and movies, respectively, that is whether they are asking different information about the same entity (pivoting on the modifier), or the same information about different entities (pivoting on the entity). To determine which is the case, we can look at the actual entities that form part of these sessions to see if the entities are repeated across queries. In fact, we find evidence of both kinds of behavior when looking at the actual entities and modifiers in queries. By looking at frequent sequence of modifiers within the sessions that con-

[6] `http://www.philippe-fournier-viger.com/spmf/`

Level	Pattern	Support
L1	$/film/film$	0.379
	$/film/actor$	0.256
	$/common/topic$	0.107
	$/tv/tv\_program$	0.106
	$/organization/organization$	0.094
L2	$/film/film \longrightarrow /film/film$	0.145
	$/film/actor \longrightarrow /film/actor$	0.098
	$/film/film \longrightarrow /film/actor$	0.04
	$/tv/tv\_program \longrightarrow /tv/tv\_program$	0.036
	$/organization/organization \longrightarrow /organization/organization$	0.035
L3	$/film/film \longrightarrow /film/film \longrightarrow /film/film$	0.041
	$/film/actor \longrightarrow /film/actor \longrightarrow /film/actor$	0.03
	$/organization/organization \longrightarrow /organization/organization \longrightarrow /organization/organization$	0.011
	$/tv/tv\_program \longrightarrow /tv/tv\_program \longrightarrow /tv/tv\_program$	0.011

Table 3: Top 5 frequent sequence patterns using λ_{type} above the threshold of 1%.

Level	Pattern	Support
L1	*movie*	0.391
	cast	0.096
	movies	0.091
	quotes	0.038
	trailer	0.034
	new	0.027
	2011	0.027
	free	0.023
	soundtrack	0.021
	watch	0.021
L2	$movie \longrightarrow movie$	0.169
	$movies \longrightarrow movies$	0.038
	$cast \longrightarrow cast$	0.028
	$movies \longrightarrow movie$	0.025
	$movie \longrightarrow movies$	0.023
	$quotes \longrightarrow quotes$	0.019
	$movie \longrightarrow cast$	0.018
	$movie \longrightarrow trailer$	0.012
	$movie \longrightarrow moviecast$	0.011
	$new \longrightarrow new$	0.01

Table 5: Frequent patterns for older movies (λ_{dict})

tain at least three actors (not depicted), we can see evidence of incremental query construction, i.e. the user starting out with the name of an entity (e.g. an actor name) and later adding modifiers such as *movie*, *pics* or *bio* and possibly additional disambiguation terms. Pivoting around entities is also present.

As suggested, we can also filter our data using our indices to interesting subsets of sessions. For example, Tables 4 and 5 compare the frequent patterns in λ_{dict} for sessions that reference recent movies (with initial release date in 2011 according to Freebase) versus older movies. As posited, we can see that the needs differ: e.g. for newer movies users are more interested in the *trailer* while for older movies they are looking for the *cast* as the primary information to determine whether they would like to watch the movie.

5.4 Predicting Website Abandonment

When the goal is to keep users on a particular website, one can speak of users being lost when they navigate away from the website. A frequently used definition of lost user behavior in the context of Web search is when a user returns to the search engine and clicks on another search result in the list, without altering the query. This is taken as an indication that the previously visited page was not satisfactory. We can speak of a user being gained from the perspective of one website when they are lost for another website.

The linking of queries to URI's allows us to make a slightly broader definition of users being lost, namely to include users who send out a new, syntactically different query, but which is semantically the same, i.e. is linked to the same URI.

Definition 2 (Loosing query). *Given a query q that leads to a click c_w on website w, q is a 'loosing query' if one of the following two session patterns occur:*

1. q_1 - c_w - q_2 - c_o

2. q_1 - c_w - c_o

where website o is different from website w, and q_1 and q_2 are linked to the same entity.

We cast the task of predicting abandonment as a binary classification problem, where learning is performed using Gradient Boosted Decision Trees (GBDT) [10] which brings state of the art classification performance in a number of domains, including Web search [26]. We have also tried different alternatives, including Support Vector Machines with different kernels, but the results brought inferior performance, and for brevity, we leave those experiments out. Table 6 lists the features used, which include signals from the query itself and its extracted semantic information, as well as click features. We investigated richer representations of the data, including features such as previous and next query as well as entities in the session and their types, but this resulted in a negligible performance impact, so we omit them from the analysis.

5.4.1 Evaluation

In this set of experiments, our aim is to predict that a user will be gained or lost for a particular website. There are three tasks addressed using supervised learning:

Task 1 predict that a user will be gained or lost for a given website. We use all features, including the click on the loosing website.

Task 2 predict that a user will be gained or lost for a given website, excluding the loosing website as a feature.

Task 3 predict whether a user will be gained or lost between two given websites.

Query Features:
1: query string
2: query entity freebase id
3: query entity freebase type
4: query keyword
Click Features:
5: first clicked site (loosing site)

Table 6: Features used for predicting abandonment

The ground truth is collected using the information present in the sessions of the search engine (this is, the user clicked on site B after being on site A). Note that for tasks 2 and 3 the click is not included as a feature because it directly encodes the label to be predicted.

We report results in terms of area under the curve (AUC) although the performance using micro/macro averaged precision follows a similar pattern. We break down the experiments by learning different classifiers using an increasing number of features. These experiments use a total amount of around 150K sessions, where the ground truth is extracted automatically from the sessions data. The training and testing is performed using 10-fold cross-validation.

Table 7 show the classification results. For task 1, the highest accuracy and AUC were obtained using all the query features, that is, using the query, its entity and type but not the loosing site. For tasks 2 and 3 using all the features gave the best accuracy/AUC, again showing that indeed the entities and types contribute to the classification. Overall task 2 is more difficult than the other two task, whereas task 1 and 3 provide a moderately high accuracy.

	Feature combination				
Task Number	1	1-2	1-3	1-4	1-5
1	0.75	0.76	0.77	0.81	0.79
2	0.59	0.59	0.61	0.60	-
3	0.75	0.77	0.80	0.82	-

Table 7: Area under the curve for the different tasks and feature combinations.

The experiments show that entity and type contribute to enhance the classification performance for all three tasks of predicting website abandonment.

6. DISCUSSION AND CONCLUSIONS

We have shown how we use semantic knowledge to aid several types of analyses of queries on a commercial Web search engine. First, we have linked queries to entities in the linked open data cloud by running the query on a Web search engine. We have argued that these entities can be used to more broadly tag queries as being navigational or being a 'loosing query', i.e. a query upon which a user abandons a website. Second, we have used the entities – including the semantic type of the entity and temporal information about the entity – to study patterns of usage behavior. By studying single queries we observed that different websites attract queries of a different type. This suggests that generalizing queries to types could provide valuable insights to website owners or others who want to compare or qualify website use. Similarly, an analysis of sessions of sequential queries showed that users frequently keep searching for the same type of entity. These results are potentially useful for disambiguation of entities in queries: if the type of one query in a session is known, this is a clue for the type of other queries in the same session. Third, we have used entities and their associated information to learn to predict whether a user will be lost, i.e. navigate away. The experiments showed that semantically enriched information helps to improve abandonment prediction.

Generalizability of the Approach.

Our method depends on the availability of Linked Open Data on the topics of the queries. While Wikipedia and Freebase are a natural choice for a wide range of topics, we believe the core of our approach is independent from the LOD source that is used. To analyze query patterns and predict website abandonment we first linked queries to entities and then generalized them to types. This type of generalization does not rely on the structure of Wikipedia or Freebase as subsumption hierarchies are found in the majority of Linked Open Data sources. The same holds for our approach to the detection of navigational queries. We rely on the availability of Linked Open Data about the official website of companies and organizations, but the particular LOD source is not important. To detect navigational queries we used (among others) the $foaf$: *homepage* property, which is widely used in the LOD cloud.

Further research is needed to verify whether other domains benefit from this type of analysis in the same way as the movie domain. Where the majority of movie related queries fall in a limited number of classes (movies, actors, tv programs, etc.) the results of the analysis might be less clear in broader domains, where queries are distributed over a larger number of classes. We have used a domain-specific property in one case: to analyze patterns of queries we divided movies in two groups based on the *release_date* property. While there is a time dimension in many domains, applicability in other domains is not self-evident. With this approach we have shown that the rich background knowledge that is available in the LOD cloud can be used to easily filter queries based on domain-specific properties.

Future Work.

Our results show promising directions for the inclusion of semantic information in usage analysis and are currently being implemented at Yahoo! as part of an interactive toolkit. Apart from using type and date information, our approach could be extended with more domain specific relations between queries – in our domain e.g. movie-actor relations, or actor-spouse relations. Similarly, we see possibilities for domain independent information such as locations or categories of entities. Semantic usage mining has taken us one step closer to understanding what users do on the Web.

7. ACKNOWLEDGEMENT

The authors would like to thank Tim Potter (Yahoo! Research) for his contributions in implementing the system described.

8. REFERENCES

[1] Ricardo Baeza-Yates, Aristides Gionis, Flavio Junqueira, Vanessa Murdock, Vassilis Plachouras, and Fabrizio Silvestri. The impact of caching on search

engines. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR '07, pages 183–190, New York, NY, USA, 2007. ACM.

[2] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '07, pages 76–85, New York, NY, USA, 2007. ACM.

[3] Dominik Benz, Beate Krause, G. Praveen Kumar, Andreas Hotho, and Gerd Stumme. Characterizing semantic relatedness of search query terms. In *Proceedings of the 1st workshop on explorative analytics of information networks (EIN2009)*, Bled, Slovenia, September 2009.

[4] Bettina Berendt, Laura Hollink, Vera Hollink, Markus Luczak-Rösch, Knud Möller, and David Vallet. Usage analysis and the web of data. *SIGIR Forum*, 45(1):63–69, May 2011.

[5] Bettina Berendt, Gerd Stumme, and Andreas Hotho. Usage mining for and on the semantic web. *Data Mining: Next Generation Challenges and Future Directions*, pages 461–480, 2004.

[6] Roi Blanco and Paolo Boldi. Extending bm25 with multiple query operators. In *Proceedings of the 35th annual internatinal ACM SIGIR conference on research and development in information retrieval*, pages 921–930, 2012.

[7] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM.

[8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107 – 117, 1998. Proceedings of the seventh international World Wide Web conference.

[9] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.

[10] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378, February 2002.

[11] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *SIGIR*, pages 267–274, 2009.

[12] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, August 2007.

[13] Katja Hofmann, Maarten de Rijke, Bouke Huurnink, and Edgar Meij. A semantic perspective on query log analysis. In *Working Notes for CLEF*, 2009.

[14] Vera Hollink, Theodora Tsikrika, and Arjen P. de Vries. Semantic search log analysis: A method and a study on professional image search. *Journal of the American Society for Information Science and Technology*, 62(4):691–713, 2011.

[15] Julia Hoxha, Martin Junghans, and Sudhir Agarwal. Enabling semantic analysis of user browsing patterns in the web of data. *CoRR*, abs/1204.2713, 2012.

[16] Bouke Huurnink, Laura Hollink, Wietske van den Heuvel, and Maarten de Rijke. Search behavior of media professionals at an audiovisual archive: A transaction log analysis. *Journal of the American Society for Information Science and Technology*, 61(6):1180–1197, 2010.

[17] Bernard J. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library and Information Science Research*, 28(3):407 – 432, 2006.

[18] Bernard J. Jansen and Amanda Spink. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1):248 – 263, 2006.

[19] Martin Kurth. The limits and limitations of transaction log analysis. *Library Hi Tec*, 11(2):98–104, 2002.

[20] Edgar Meij, Marc Bron, Laura Hollink, Bouke Huurnink, and Maarten de Rijke. Mapping queries to the linking open data cloud: A case study using dbpedia. *Journal of Web Semantics*, 9(4):418–433, 2011.

[21] Peter Mika, Edgar Meij, and Hugo Zaragoza. Investigating the semantic gap through query log analysis. In *proceedings of the 8th international semantic web conference*, volume 5823, pages 441–455. Springer Berlin / Heidelberg, 2009.

[22] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering*, pages 215–224, 2001.

[23] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World Wide Web*, WWW '10, pages 771–780, New York, NY, USA, 2010. ACM.

[24] Ronald E. Rice and Christine L. Borgman. The use of computer-monitored data in information science and communication research. *Journal of the American Society for Information Science*, 34(4):247–256, 1983.

[25] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval.* Digital Libraries and Electronic Publishing. MIT Press, September 2005.

[26] Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon Sun. A general boosting method and its application to learning ranking functions for web search neur. *Advances in neural information processing systems*, 20:1697–1704, 2007.