From "Selena Gomez" to "Marlon Brando": Understanding Explorative Entity Search

Iris Miliaraki
Yahoo Labs
Barcelona, Spain
irismili@yahoo-inc.com

Roi Blanco
Yahoo Labs
Barcelona, Spain
roi@yahoo-inc.com

Mounia Lalmas
Yahoo Labs
London, UK
mounia@acm.org

ABSTRACT

Consider a user who submits a search query "Shakira" having a specific search goal in mind (such as her age) but at the same time willing to explore information for other entities related to her, such as comparable singers. In previous work, a system called Spark, was developed to provide such search experience. Given a query submitted to the Yahoo search engine, Spark provides related entity suggestions for the query, exploiting, among else, public knowledge bases from the Semantic Web. We refer to this search scenario as *explorative entity search*. The effectiveness and efficiency of the approach has been demonstrated in previous work. The way users interact with these related entity suggestions and whether this interaction can be predicted have however not been studied. In this paper, we perform a large-scale analysis into how users interact with the entity results returned by Spark. We characterize the users, queries and sessions that appear to promote an explorative behavior. Based on this analysis, we develop a set of query and user-based features that reflect the click behavior of users and explore their effectiveness in the context of a prediction task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; H.1.2 [**User/Machine Systems**]: Human information processing

Keywords

related entity; Yahoo Spark system; log analysis; user click behavior; explorative search

1. INTRODUCTION

Search engines are rapidly evolving from the ubiquitous *ten blue links* that have dominated Web search results for over fifteen years. One main driver for enhancing the capabilities of search systems has been the ability to incorporate structured data in their search results page [4, 12]. In general, when a user submits a query to a search engine their corresponding information need can be categorized at a high level as navigational or informational [5]. In the latter case, there are many situations in which users know exactly what they are looking for and would like immediate answers, whereas in other cases they are willing to explore to extend their knowledge, satisfy their curiosity or simply for the fun of it [13]. This situation happens, for example, when learning about people in the news, following a long term interest in music, movies or sports or when exploring destinations for future travel. As an example, consider a user who submits a search query "Shakira". The user may have a specific search goal in mind (such as the age of Shakira, some recent news about her or her next concerts). At the same time that user may be willing to explore other choices given to her, such as other artists related to Shakira, including those who have performed or sang with her.

In the realm of Web search, knowledge bases (KBs) contain information about different real-world objects or concepts commonly referred to as *entities*. Given that KBs generally store typed relationships between entities (such as "*born_in*" for entities *Shakira* and *Colombia*), they can be represented as a graph using entities as nodes and relations as edges, and are also known as *knowledge graphs* (KGs). Search engines can exploit such KGs not only for displaying additional facts and direct information about the central entity in a query, but also to provide extended suggestions for users who would like to browse. Queries containing entities correspond to the most popular type of queries [35] and are typically referred as *named entity queries* or simply *entity queries*. Given that the system is able to identify the real-world entity that is being referenced in an entity query and link it to a knowledge base, then it can provide recommendations of related entities based on the relationships explicitly encoded in the knowledge base.

As an example, consider the search result page in Figure 1 which displays the results on Yahoo Search for the entity query "barcelona spain". Besides the main search result links representing document results and images, we can see on the right panel suggestions for points of interest in Barcelona. This requires an understanding that this query represents the city of Barcelona, and a ranking over the points of interest, which is performed over the entities neighboring *Barcelona* in a KG. This is exactly what a system called Spark is doing [3]. Spark provides extended browsing capabilities through the knowledge graph exploiting public knowledge bases from the Semantic Web, in combination with proprietary data, to provide related entity suggestions for web search queries.

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741284.

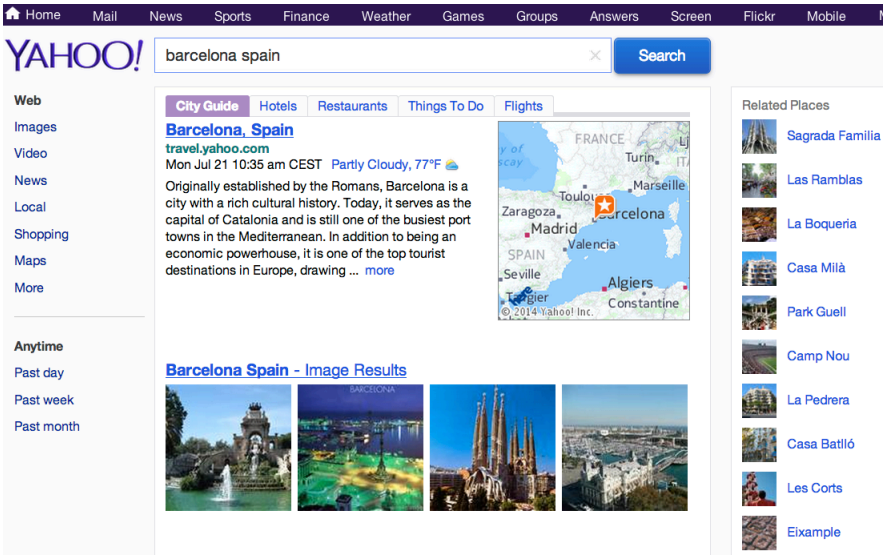


Figure 1: Search result page for the query *"barcelona spain"*. Spark results (related places) appear on the right.

So, what has the young actress "Selena Gomez" to do with "Marlon Brando"? This is a path that can be potentially explored by a user following suggestions made by Spark. Starting from the entity query "selena gomez", Spark is triggered recommending several related entities for exploring, one being *Kevin James*, which itself also triggers Spark to suggest *Henry Winkler*; this process continues with the following sequence of clicks on recommended entities, *Robin Williams*, *Al Pacino*, and *Robert de Niro*, to finally reach *Marlon Brando*. The relationship between each pair of these entities, here persons, comes from them appearing in the same event, belonging to the same industry, involved in the same film, and so on. This sequence of queries and clicks was one that was actually followed by users, and is an example of what we refer to as an *explorative entity search*.

While entity ranking systems have evolved significantly in terms of more effective and efficient algorithms for generating entity recommendations, no previous work has studied how users interact and engage with these systems and under which circumstances they are willing to follow an explorative path not directly related to their information need. This paper attempts to fill this missing gap, which has not been addressed before in the literature, focusing on how users interact with results returned by existing Web scale entity recommendation systems. We perform a large-scale analysis on search logs of $2M$ users on the existing Spark system [3].

We analyze the types of queries and entities that users interact with, exploring who are the users interacting with Spark results, the characteristics of their sessions, and the interplay between the typical search results and Spark entity recommendation results. Our analysis clearly indicates that Spark is able to promote an explorative behavior. Examples of such cases arise with long user sessions and entity queries that lack any surrounding context indicating a specific need (e.g., "images"). Based on our analysis, we develop a set of query and user-based features that reflect the click behavior of the users and explore their impact in the context of click prediction on Spark (i.e., predicting future interactions with entity recommendations). We find that the previous history of users, session duration and query-click through rate are the most discriminating features for predicting whether a user will click on a recommended entity or not.

2. RELATED WORK

Often, the user's initial interest can be uniquely linked to an entity in a KB, and in this case it is natural to recommend the explicitly linked entities for further exploration. Many useful facts about *entities* (people, locations, organizations, or products) and their relationships can be found in data sources such as Wikipedia or Freebase, while others need to be extracted from unstructured text. In real world knowledge bases, however, the number of linked entities may be very large and not all linked entities are equally relevant therefore making ranking related entities a must.

All major search engines currently display *entity recommendations* in the context of a web search session. The focus of our study is Yahoo's Spark system [3]. Spark extracts several signals from a variety of data sources, including user sessions, Twitter and Flickr, using a large cluster of computers running Hadoop. These signals are combined with a machine-learned ranking model to produce a final recommendation of entities to user queries, which is currently powering Yahoo Search results pages.

The problem of discovering interesting relations from unstructured text has led to a surge in research on the topic of *entity search* [8, 15, 20, 24, 26]. Entity search is viewed as an ideal paradigm to support explorative search. It provides semantically rich answers, i.e., entities and their relations, which are often considered more suitable for search exploration than individual web pages. Spark [3] falls in this type of systems.

Methods for generating query recommendations have focused on the analysis of query logs at the level of entire query strings or tokens, without any concern for the linguistic or semantic structure of the query strings [18]. However, more recent work has recognized that queries have internal structures and can be classified based on their semantic intent [21, 25, 16]. These observations have led to the development of the area of semantic search [4]. The broad area of

Table 1: Spark input graph

Domain	# of entities	# of relations
Movie	205,197	9,642,124
TV	88,004	17,126,890
Music	294,319	77,673,434
Notability	585,765	89,702
Sport	75,088	1,281,867,144
Geo	2,195,370	4,655,696
Total	3,443,743	1,391,054,990

semantic search in general refers to finding information in knowledge bases using unstructured, keyword queries typical for search engines; Spark makes explicit use of semantic web technologies [2].

Various evaluation efforts like INEX Entity and Linked Data tracks,[1] TREC Entity track,[2] and SemSearch challenge[3] have been carried out to assess the effectiveness of entity search approaches. Their focus was mainly with respect to evaluating how well the proposed techniques worked, in terms of delivering the right results for a given query. To the best of our knowledge, there has not been any large scale evaluation looking at how users engage with such results. This is the focus of this paper studying user behaviors for queries for which Spark returns recommended entities, for users to browse during their search session.

There has been many works studying user behavior with search engines. These studies examined query log data to elicit search patterns and other signals that can inform about various aspects of the search process [9, 17]. For instance, click-through data has been extensively used to assess how satisfied (or unsatisfied) users are with search results returned to them, mostly in terms of their relevance, e.g. [28, 32]. Query intent [17], user frustration [10] and search engine switching [33] have also been studied. Finally, tracking user search behavior over time (so-called historical data) has been shown to lead to better prediction of future search behavior, e.g. [6, 27, 31]. In this paper, we also study user search behavior to gain similar insights. Our context is however different; we look at how users behave when presented search results, here entities, that are served to them to promote exploration. All the above studies studied user behavior when users were presented search results that were served to "solve" their information need.

Explorative search[4] addresses the problem of less well-defined information need: users are unfamiliar with the problem domain, or the search tasks require some exploration [22, 34]. It also includes scenarios where users are enjoying exploring without a specific search objective in mind, they just want to get an update, or be entertained during their spare time. Searching for fun[5] or having fun while searching involves activities such as online shopping with nothing to buy, reading online, watching funny videos, and even clicking on images that have little relation with the original information needs [30]. Various works looked at approaches to promote and evaluate explorative search [14, 36]. This paper is also concerned with explorative search, and focuses on how users interact with entities returned to them *in addition* to the search results, enticing them to explore.

Although recommendations similar to Spark appear on the result pages of Google and Bing, details of these systems and their large-scale evaluation have not been published. Spark has been described in previous work [3, 19], where the focus was largely on ranking. The system has evolved considerably since then, and is now ready to be studied in terms of what type of engagement it promotes, and whether user behavior patterns can be identified. This paper studies how users interact with results returned by Spark, in terms of the submitted queries, the user demographics, and the interplay between typical search results and Spark entity recommendation.

3. SPARK: AN ENTITY RECOMMENDER SYSTEM

Given the large number of related entities that are usually available in a knowledge base, Spark selects the most relevant ones to show based on the current query of the user. Unlike the standard task addressed in entity search, where most related work has focused [8, 15, 20, 24, 26], Spark does not try to find information directly related to the user's *current query* but to recommend possible *future queries* to explore. Hence, Spark aims to promote exploration search within the remit of entity recommendation, which we refer to as *explorative entity search*.

The main component of Spark is its knowledge base, represented as a large entity graph. Spark takes this graph as input, and applies a ranking function to extract a weighted subgraph consisting of the most important entities, their most important related entities, and their respective types. This entity graph is drawn from a larger Yahoo Knowledge Graph, a unified knowledge base that provides key information about all the entities that are deemed worthwhile, and how they relate to each others. Entities, relations, and information about them are automatically extracted from multiple data sources on an ongoing basis. Data sources consist of Web extractions, structured data feeds, and editorial content. Both open data sources and closed data sources from paid providers are leveraged. Reference data sources such as Wikipedia and Freebase provide background information for a wide variety of domains whereas domain-specific data sources provide rich information for domains such as Movie, TV, Music, or Sport.

The knowledge base is modeled as a property graph with a common ontology, which was developed over 2 years by the Yahoo editorial team and is aligned with schema.org. Today's knowledge graph focuses on the domains of interest of key Yahoo sites, including the News domain (various types of entities), the Movie domain (movies, actors, directors, etc.), the TV domain (TV shows, actors, hosts, etc.), the Music domain (albums, music artists, etc.), the Sport domain (leagues, teams, athletes, etc.), and the Geo domain (points of interests, etc.). Overall, the graph that Spark uses as input consists of 3.5M entities and 1.4B direct and indirect relations from the Movie, TV, Music, Sport and Geo domains. See table 1 for details.

For every triple (subject, relation, object) in the knowledge base, Spark extracts over 100 features belonging to three main categories, i.e., co-occurrence, popularity, and graph-theoretic ones. Co-occurrence features are motivated

[1] `http://www.inex.otago.ac.nz/tracks/entity-ranking/entity-ranking.asp`
[2] `http://ilps.science.uva.nl/trec-entity/`
[3] `http://semsearch.yahoo.com/`
[4] The term "exploratory" search is also used [22].
[5] A workshop Searching4Fun focusing on pleasure-driven, rather than task-driven, search. See `http://www.cs.nott.ac.uk/~mlw/s4f2014/`.

by the fact that entities that frequently occur together in a given set of observations (sets of short text pieces) are more likely to be related to each other. Spark uses Yahoo Search, Twitter, and Flickr as sources to extract the co-occurrence information. For example, for the query "flight from barcelona to madrid", "Barcelona" and "Madrid" are identified as two entities that occur together. In case of Twitter and Flickr, the occurrence frequencies are extracted from tweets and user tags associated with photos, respectively.

Popularity features represent the frequency of an entity in a given data source. Examples include the number of matching results in Yahoo Search, when the entity string is used as a query, as well as frequency information with respect to queries, query sessions, tweets, and photo tags. Finally, the graph-theoretic features include graphs metrics, for example the number of shared vertices (common neighbors) between two entities. The extracted feature vectors are the sole input to the ranking process. Spark uses learning to rank approaches to derive an efficient ranking function for entities related to a query entity. The system employs Stochastic Gradient Boosted Decision Trees (GBDT) for deriving the learning (ranking) function [11].

Figure 1 shows the related entity recommendations made by Spark in Yahoo Web Search for the query "barcelona spain". This is a place, so Spark returns "related places". These are the typical places that are visited by millions of people, as they correspond to the city's top attractions. After clicking one of the related entities a new query is launched with the related entity leading to a new search result page, which can potentially contain more recommendations by Spark based on the new query.

The effectiveness and efficiency of the Spark system has been described recently in [3]. We study how users interact with the related entity recommendations returned by Spark as it performing a large-scale analysis on search logs of $2M$ users. We characterize the users, queries and sessions that appear to promote an explorative behavior.

4. ANALYSIS

We performed a large-scale analysis exploring the click behavior of users submitting entity queries that trigger Spark. We first examine how query characteristics affect the click probability on related entities focusing on the click-through rate of the query, the type of the entity, and the surrounding context of the entity in the query. Then, we study the users, investigate the impact of their demographics (age and gender) on the observed click behavior, and attempt to characterize sessions (e.g., in terms of duration) that lead to a successful interaction with Spark (i.e., at least one click on a related entity). We also distinguish between different navigation patterns of users while interacting with Spark. Last, we examine whether time has an effect on user behavior (i.e., in terms of the day of visit or time of visit) and report also other interesting trends discovered.

4.1 Dataset, metric and scope

We collected a sample of 2M users focusing on their activity on queries that trigger Spark to recommend related entities. We only consider queries issued at least 100 times.

Spark aims to promote exploration. This means that when a set of recommended entities are returned to the user, a successful outcome is a click by the user on at least one of them; the user is exploring. A key metric to measure this is *click-through rate (CTR)*, which is commonly used to evaluate user satisfaction in search [1] or online advertising [29]. For a given query, we define the *search CTR* (respectively the *Spark CTR*) as the total number of clicks on a search (respectively Spark) result returned as an answer to this query divided by the total number of times that the query was issued (respectively, triggered Spark). The latter is also referred to as a *view*, and corresponds to the event that a particular item has been displayed on the search page (an entity or a document result). Likewise, a user can also be characterized by a click-through rate observed over a time period given the queries she has issued and her performed clicks.

Our analysis could include other metrics, such as click dwell time, to differentiate for instance long versus short clicks or time between consecutive clicks. We leave this for future work, as, in this paper, we want to understand what makes users click on the entities returned to them by Spark. We therefore focus on CTR, as our metric to study user exploration with Spark results.

We also restrict ourselves to sessions where Spark was triggered to recommend entities for users to explore as part of their search session. As discussed in the previous section, not all queries submitted by users lead to entities returned by Spark. Although it would be interesting to compare users' "explorative behavior" when returned or not returned Spark results, our focus is to gain an understanding on what makes users explore the entities recommended to them by Spark, and whether this can be predicted.

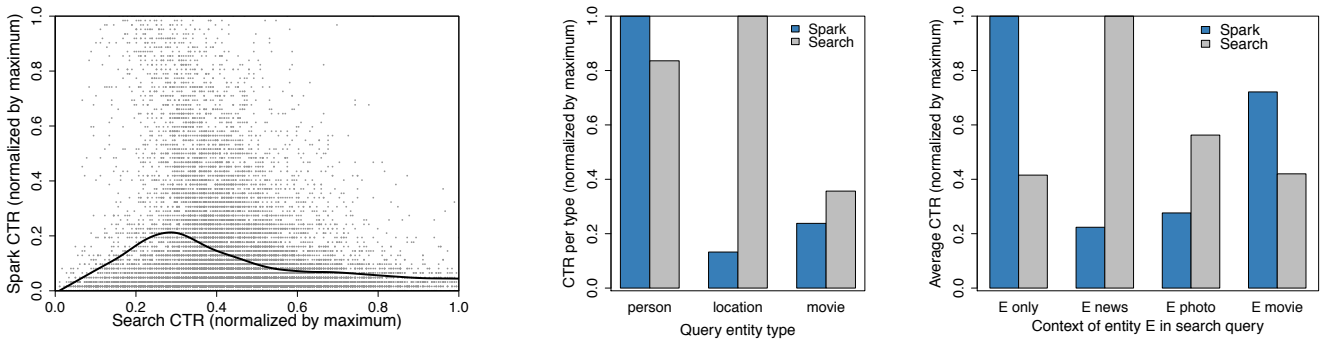
Finally, for confidentiality reasons we normalize all raw CTR values via a linear transformation and report only relative CTR values.[6]

4.2 Query-based analysis

Consider a user issuing an entity query which leads to a search result page like the one in Figure 1. We distinguish between the following user actions: (1) the user clicks on one or more search results situated in the central page panel (*search click* on organic results), (2) the user clicks on one or more related entities suggested by Spark on the right page panel (*Spark click*), (3) the user does not click. We look at these actions with respect to CTR, entity type and entity context.

Search versus Spark CTR. We look into the relationship between search and Spark click-through rate per query. The results are plotted in Figure 2(a), where the CTR values are plotted after eliminating outliers in the upper deciles (i.e., points referring to less than 2% of the observations) and normalizing the values by the remaining maximum one. We can observe the following cases. First, queries having a relatively low CTR for search results tend to also have a low CTR for Spark results. Then, we identify a mutual growth area where we find queries with relatively high search and Spark CTR. Finally, queries with the highest search CTR values tend to have the lowest values of Spark CTR. Overall this indicates that queries with an average search CTR (neither low nor high) are those for which the returned Spark recommended entities are more likely to "attract" user attention. However, as the plot depicts, the search CTR does not directly indicate its Spark CTR value.

[6]Unless otherwise stated, we normalize by the maximum value observed in each plot.



(a) CTR per query

(b) CTR per entity type

(c) CTR per query context

Figure 2: Query-based analysis

CTR across entity types. In Spark, each entity is associated with a type, such as "person", "location" and "movie". We focus on these types in this study since these are the most common ones. We study now how the click-through rate varies across these entity types. The results are reported in Figure 2(b). We note how some types of entities (persons, movies) are more prone to receive clicks than others (locations). This might be due to the fact that some entity types are intrinsically more likely to spur an explorative intent than others. All reported average values are significantly different (t-test, $p < 0.001$).

Effect of entity context on query. We study how the surrounding context of an entity in a submitted query affects the click behavior of users (see Figure 2(c)). The notion of context in an entity query was used in [35] to build a taxonomy of web search intents for entity queries. As an example, consider the query "jennifer aniston photo", with the intent to discover pictures of Jennifer Aniston (i.e., the surrounding context is "photo"). In this case, Spark may be triggered recommending a set of related entities, as in the case of a query containing only the entity "jennifer aniston". As we observe in Figure 2(c), the user who submits an entity query without any surrounding context is more likely to click on Spark results (denoted as "E only"). However, a user who submits an entity query with the surrounding context such as "news" or "photo" is less likely to click on a Spark result. This might be explained by the fact that users in the latter case are already looking for a specialized set of results within the context of that particular entity (i.e., the actress) and the odds of being willing to explore other related entities (i.e., other actors) are lower. The same holds for the context "news"; returning related entities when users are interested in reading about news does not seem to trigger an explorative behavior through Spark entities. Last, the context "movie", aiming most probably to identify the entity of a query (e.g., "walk the line movie"), leads to a relatively higher CTR on Spark. All reported average values are significantly different (t-test, $p < 0.001$).

To conclude, queries with average CTR are more likely to have their Spark recommended entities clicked, the entity type (person versus location) affects whether the corresponding Spark recommended entities are clicked, and finally, queries associated with a genre (e.g., news) or media context (e.g., image) are less likely to have their Spark recommended entities clicked.

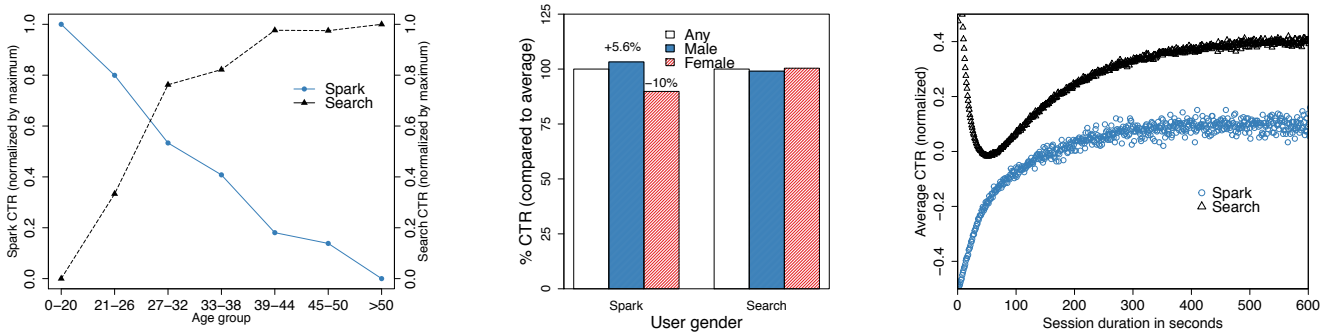
4.3 User-based analysis

We continue our analysis from the user's perspective. First, we study whether the demographics of the users affect their click behavior on Spark and also explore whether specific characteristics of a user session such as its overall duration affect this behavior. In the former case, we consider only a subset of users who have disclosed this information (age and gender) as part of the registration process while obtaining an account with Yahoo.

Spark & search CTR by demographics. In Figure 3(a), we depict the average CTR values for search and Spark results across users of different ages. The plotted values are normalized by the maximum average value separately for search and Spark CTR values. As we observe, there is a different effect on search versus Spark click-through rate as the age increases. In case of Spark, users of a younger age tend to have a relatively higher CTR than the users of the other groups. The results for Search CTR are complementary to this; users from the older age groups tend to click more on search results. Figure 3(b) depicts the difference of the Spark and search CTR rates between male and female users. While for search no difference is observed, for Spark, male users tend to exhibit a higher CTR than female users.

Session duration effect. Figure 3(c) shows how the duration of a user session affects the average Spark CTR and the average search CTR of the corresponding session. Spark and search CTR values are normalized by their standard score and shifted by a constant value for visualization purposes. Shorter sessions have the higher search CTR; users come, find what they are looking for and leave. They are satisfied with the results. As the session length increases, search CTR diminishes, likely because users are trying various queries to find the information they are looking for. After a while, the CTR again increases slowly, which may suggest either a stubborn user, a complex information need, or users who are simply exploring.

The behavior is different for Spark CTR. The longer the session, the higher the CTR. This suggests that when a user clicks on a Spark result she is willing to explore the



(a) CTR across age groups

(b) CTR across gender

(c) Session duration effect

Figure 3: User-based analysis

Query entity	Entity type	Pattern type	Navigation pattern
Tennis	Sport	star	{Serena Williams, Maria Sharapova}
Barcelona	Location	star	{La Sagrada Familia, Park Guell, Tibidabo}
Marlon Brando	Person	star	{Elizabeth Taylor, Al Pacino, Sophia Loren, Charlie Chaplin}
Scent of a woman	Movie	path	Al Pacino ⇒ Marlon Brando
Basketball	Sport	path	Carmelo Anthony ⇒ Lebron James ⇒ Dwight Howard
Samuel Jackson	Person	path	Bruce Willis ⇒ John Travolta ⇒ Nicolas Cage
Catherine Zeta Jones	Person	path	Julia Roberts ⇒ Natalie Portman ⇒ Scarlett Johansson

Table 2: Example of user navigation patterns using Spark

recommendation, and may continue to do so for a while. The fact that we observe a difference in the curves of search CTR and Spark CTR clearly shows that Spark is indeed returning results that entice users to explore; when users do so, they become more engaged as they interact with the Spark recommendations; it is not anymore about satisfying an information need, but satisfying curiosity.

User navigation patterns. Finally, we studied how users interact with Spark in terms of navigation. We distinguish between *star* and *path* behavior where the former refers to the case where a user tends to click on many related entities for a given entity query (increasing *breadth*) and the latter refers to the case where a user tends to follow a path of related entities issuing different successive queries (increasing *depth*). Different example navigation patterns are shown in Table 2. An example of a star navigation pattern begins from the entity query "Barcelona" and continues with three different related entities, i.e., "La Sagrada Familia", "Park Guell" and "Tibidabo". An example of a path navigation pattern begins with the query entity "Catherine Zeta Jones", continues with "Julia Roberts", "Natalie Portman", and finally ends with "Scarlett Johansson".

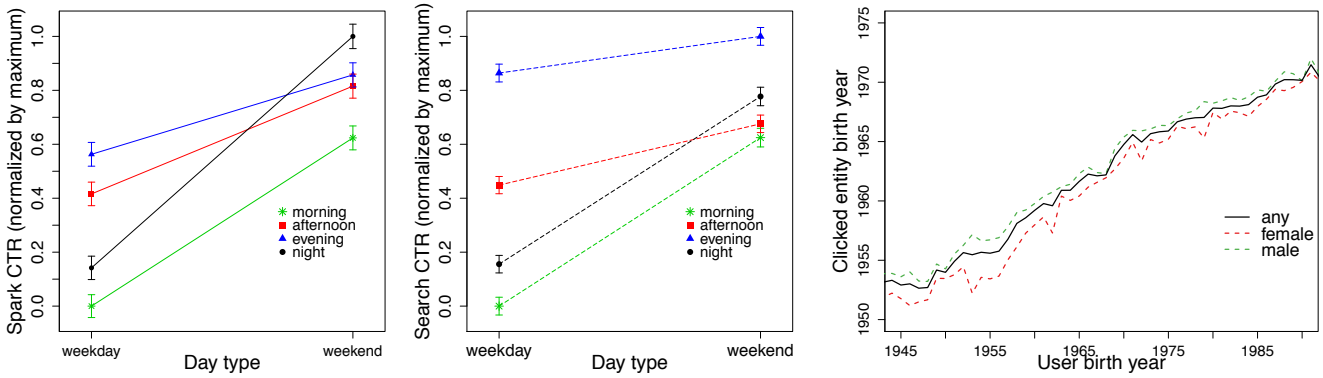
The distribution of users' navigation patterns according to their breadth and depth properties are summarized in Tables 3 and 4. We also include the separate distributions for entities of type person and type location. In general, there is a tendency towards a path navigation pattern from users in 13% of the cases compared to only 4% for star navigation patterns. This suggests that users, when returning to the initial search result page after clicking on a Spark result, are less likely to engage in further exploration. They were enticed to click, but now they are back to what they were doing, i.e., a search. On the other hand, when they continue clicking on further entities, they are engaged in exploration. This difference could help to classify whether users are ready to explore or are more focused on solving their information needs.

To conclude, demographics have some effect on CTR. However, in the context of gender, it is not clear if this comes from the type of queries triggering Spark (to return entities) or from the user gender itself. We return to age when we discuss trends. However, what is clear is that Spark results have the potential to lead users in explorative behavior; when they start engaging with Spark search results, it is more likely that the session is longer and the navigation deeper.

4.4 Day of the week effect

We now study whether Spark and search CTR varies for different days of the week or different times of the day. The results are shown in Figures 4(a) and 4(b). For both search and Spark, CTR is significantly higher on weekends (t-test, $p < 0.001$). We recall here that our dataset only contains sessions where the query triggered Spark to recommend entities. Therefore, this indicates that the types of queries that triggered Spark are those more likely to lead to users interacting with the results over the weekend than weekdays. This is not surprising as the entities, and their relationships, covered by Spark (i.e., stored in Spark knowledge base) relate to persons, locations, movies and sports, many of which have an "entertainment" or "leisure" nature.

Moreover, the difference in CTR between weekdays and weekends is more accentuated with Spark CTR in the afternoon and at night. Users are more likely to click on Spark results during these parts of the day on weekends. For the morning, it is the inverse (a stronger difference is observed with Search CTR). Interestingly, both the Spark CTR and the search CTR do not increase significantly from weekdays and weekends in the evening (there is hardly any increase for Search CTR).



(a) Spark CTR (normalized) (b) Search CTR (normalized) (c) Birth year correlation

Figure 4: Time effect (time of day & day type) and *birth year* correlation

Breadth	All types	Person	Location
1	95.36%	95.46%	95.27%
2	4.10%	3.63%	4.21%
≥ 3	0.54%	0.91%	0.53%

Table 3: Distribution of *star* navigation patterns

Depth	All types	Person	Location
1	72.2%	67.9%	86.6%
2	13.3%	15.1%	7.5%
≥ 3	14.5%	17%	5.9%

Table 4: Distribution of *path* navigation patterns

Overall, the "entertainment" or "leisure" nature of the queries that trigger Spark (e.g. actress, movie) results in users who are more likely to engage with the Spark results (and search results) when their entertainment and leisure activities are planned (a movie trip on the weekend) or happening (browsing the web in the evening). Weekday mornings seem to be a "no-go" for explorative search.

4.5 Other trends

We also looked at many other relationships, between demographics and types of queries (e.g., male users tend to have a higher CTR on sports-related entities), and report one particularly interesting and significant relationship. In Figure 4(c), we plot the age of the users against the age of the person entities that they click on. Values on y-axis are averages of the ages of the entities clicked by users of a specific age (x-axis). To eliminate outliers and potential noise, we only consider ages for which we have at least 100 observations. We observe that a strong correlation exists, clearly showing that users are enticed to explore people of a closer age to them (Pearson correlation is equal to 0.859 with $p < 0.0001$). For instance, younger users tend to click less frequently on results returned to them about persons older than them. This suggests that users, when exploring the recommended entities, seem to choose people with which they can relate, because of their "comparable" ages. We also note the same pattern is observed across genders. If we include user ages with at least 10 observations (introducing potentially more noise), correlation drops to 0.62.

Returning to Figure 3(a), we saw that users of a younger age tend to have a relatively higher CTR than the users of the other groups. It would be interesting to see whether increasing the coverage of Spark, by including entities for which older users would relate, could increase their interaction with Spark (e.g., increasing the CTR on Spark results from older users).

4.6 Discussion

In this section we studied user behavior, with respect to user clicks on related entities suggested by Spark. Users submit a query to solve an information need. Spark results are not there to fulfill this need, but to entice users to explore entities that are related to their initial query. Our analysis clearly shows that Spark is able to promote this explorative behavior. Users are more inclined to navigate through the recommendations for certain types of queries, especially when no specific context is specified (such as "pictures"), with a higher CTR observed during weekends. When users decide to engage with a Spark recommendation, they often end up navigating through the recommendations, clearly engaging in explorative search. Contrary to standard search behavior, where a satisfactory experience is when users find the information they are looking for as soon as possible (the search sessions are short), users interacting with Spark entity recommendations seem to happily explore through these results, leading to longer sessions. Next, we look at how these insights can be used to build a model that can predict whether users will click on Spark results.

5. PREDICTION TASK

We studied the click behavior of users after issuing an entity query for which Spark recommends a set of related entities. Our analysis focused on CTR and how it is affected by the characteristics of the user, the session and the issued entity query. We now consider the problem of predicting whether a user will click or not on a related entity after issuing an query. We tackle this task by exploiting features, user- and query-based ones, inspired by our previous analysis.

5.1 Experimental setup

Dataset. We use a sample of 100k users from Yahoo search logs, from which we collect their actions over a period of 6

Feature	Description	Type
Q1	Total views of entity query	Numeric
Q2	Total clicks on related entities of entity query	Numeric
Q3	Click-through rate of entity query on Spark	Numeric
Q4	Category of search and Spark click-through rate (*low, medium, high*)	Ordinal
Q5	Type of entity (e.g., person, location, movie)	Categorical
Q6	Whether any context surrounds the entity in query string	Binary
U1	Click-through rate of user on Spark	Numeric
U2	Previous user actions ($h = 1, 2, 3$)	Categorical
U3	Session length (*short, medium, long*)	Ordinal

Table 5: Feature sets (Q contains query-based features, U contains user-based features)

months. We only consider actions related to entity queries triggering Spark. To capture the notion of "new users" interacting with Spark for the first time, we only consider users that do not have any action related to Spark during a period of one month and then at some point perform a click on a related entity suggested by Spark. After identifying such users we consider all their succeeding actions. For this task we collect up to $h = 4$ succeeding interactions per user, aiming to exploit at most $h = 3$ succeeding interactions as indicators of the user's future behavior and attempting to predict the last. An interaction consists of the query issued by the user and the action that followed (a click on a Spark result, a click on a search result, or no click at all). We consider two different samples of 100k users varying the number of *required* historical actions from 0 to exactly 3. In the first dataset, denoted as $D1$, we may include users without any or with up to 3 historical actions. In the second, denoted as $D2$, we only consider users for which 3 of their previous actions are known.

Prediction task & method. Given a user, her previous interactions, and an issued entity query, we want to predict whether the user will interact with the Spark module or not. Since our focus is on predicting Spark interactions, we consider this as the "positive event" and any other interaction is considered as the "negative event" (e.g., a click only on organic search results). We do not distinguish between a single click on a related entity or multiple clicks and consider an interaction leading to at least one click as a positive instance. We focus only on interactions concerning entity queries triggering Spark and ignore any other activity of the user. We make use of logistic regression [23] for learning and perform 5-fold cross-validation averaging all reported metrics.

Evaluation metrics. For performance evaluation we use the metrics of precision or positive predictive value (i.e., the ratio of correct positive predictions, denoted as PPV), negative predictive value (i.e., the ratio of correct negative predictions, denoted as NPV), recall or sensitivity (i.e., the ratio of positive observations correctly predicted, denoted as REC), specificity (or else recall for negative labeled instances, denoted as $SPEC$), accuracy (denoted as ACC) and area under the curve (AUC) of a ROC curve. Since our dataset is unbalanced, we use the metric of macro-averaged accuracy ($ACC = PPV \cdot 0.5 + NPV \cdot 0.5$) allowing us to give equal weight to the prediction ability of our method for each class.

5.2 Features

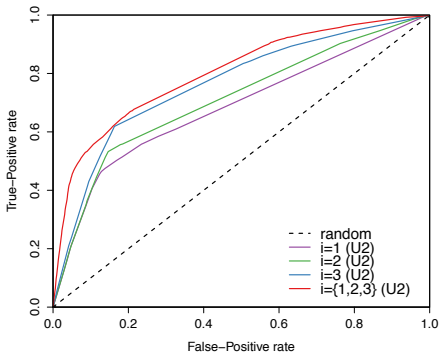
The set of features used for our prediction task are listed in Table 5. Recall that given a user issuing an entity query we want to predict whether the user will click or not on the Spark module (positive vs. negative class). We categorize our features into two sets, query-based ($Q1 - Q6$) and user-based ($U1 - U3$). The set Q of query-based features includes two groups of features related to the issued query. The first group contains query-specific historical features such total number of views (how many times the query has been issued), number of clicks and CTR. These are standard features used in many search-related prediction tasks. The second group includes the query type and the context, as these were shown to have an effect on CTR in Section 4.2. The set U contains features related to the user and includes her past behavior with respect to the Spark module such as her past click-through rate, and previous interaction when returned Spark results. U also includes the current session length (feature $U3$), which, as discussed in Section 4.3, is a good indicator of users being in "explorative mood". $U3$ captures the length of the session up to the time of the action of the user which we aim to predict (inspired by Figure 3(c)).

Besides Spark CTR (see feature $Q3$), we also consider a more coarse-grained representation (see feature $Q4$). We assign both the search and the Spark click-through rate of the query into three bins (low, medium, and high) according to the distribution of their values. This is based also on Figure 2(a), where we saw that queries with an average search CTR (neither low nor high) are those for which the returned Spark recommended entities are more likely to get clicked. As such, each entity query can be characterized by two *CTR classes*, e.g., a *medium* Spark CTR and a *high* Search CTR. These classes are then exploited to characterize the previous actions of the users (see feature $U2$ in Table 5). In total, we can distinguish between 18 types of actions (i.e., 3*3*2, where Spark/search CTR have 3 different labelings each and there are two possible user events, a click on Spark or not). The intuition behind this is to be able to differentiate between a user who clicks on a related entity after issuing a query with a high click-through rate on Spark and the user who clicks on Spark when the click-through rate of the query is low. A user who is likely to click on Spark more often than the average user can be considered a more engaged user and we aim to capture such cases.

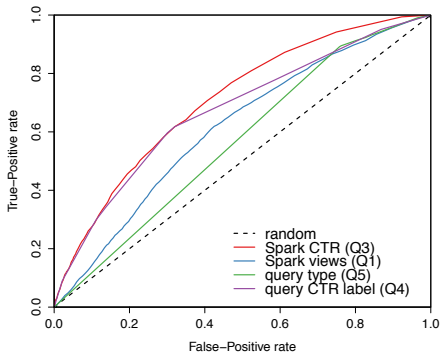
5.3 Results

The results for the different sets of features (Q, U, and $Q + U$) are summarized in Table 6. We include the performance results for D1, where the minimum number of previous available actions per user is 0 ($h \geq 0$), and for D2, where exactly 3 actions are available for each user ($h = 3$).

Overall performance per feature set (D1). In dataset D1, the user-based features (U) achieve a higher performance, increasing both in terms of precision and recall, compared to the query-based features (Q). Precision is 19.1% higher when we use the user-based features against the query-based ones.



(a) AUC curves for features $U2$



(b) AUC curves for features $Q1$,$Q3$,$Q4$,$Q5$

Figure 5: Features performance for prediction task

Feature set	Dataset	PPV	NPV	REC	SPEC	ACC	AUC
Q	D1	0.512	0.809	0.054	**0.987**	0.660	0.695
U	D1	0.703	**0.875**	0.445	0.954	0.789	0.803
Q+U	D1	**0.707**	**0.875**	**0.447**	0.954	**0.791**	**0.811**
Q	D2	0.616	0.799	0.512	0.858	0.707	0.783
U	D2	0.701	0.874	0.719	0.864	0.787	0.856
Q+U	D2	**0.704**	**0.875**	**0.722**	**0.865**	**0.789**	**0.862**

Table 6: Results for different sets of features (Q, U, $Q+U$) and datasets D1, D2 (minimum actions $h \geq 0$ and $h = 3$)

The recall achieved by Q only is extremely low (<1%) exhibiting the difficulty of predicting the positive instances when no user-based information is available. Exploiting user-based features leads to an increase of 39%. In general, predicting negative instances is an easier task and as such the query-based approach performs relatively well (80% NPV for Q vs. 87.5% NPV for U). The macro-averaged accuracy metric is relatively high, around 79% for U and $Q+U$ features sets, while it drops to 66% for the case of Q. Exploiting both user-based and query-based features ($Q+U$) does not yield any additional gains in terms of performance, exhibiting a similar performance as when only features of U are used. We report, although we do not include the separate measurements, that the individual contribution of the historical features $U3$ with respect to the other user-based features ($U1$ and $U2$), is around 18% improvement in PPV and 5% improvement in NPV. This suggests that session length is a good indicator of whether users will click or not on Spark.

Overall performance per feature set (D2). Continuing with dataset D2, we observe a similar trend where U outperforms Q in terms of all metrics (even for specificity where in $D1$, U's performance was 95.4% vs. 98.7% for Q). On this dataset all feature sets achieve a higher recall (ranging from 51% for Q to 72.2% for $Q+U$). This is most likely due to the inclusion of less variance in the dataset since we only allow users with exactly 3 actions in their past behavior. Precision remains around 70% for both Q and $Q+U$ feature sets.

Effect of individual features (U). To quantify the effect of the inclusion of user's previous actions as indicators of their future behavior we plot the corresponding AUC curves for $i = 1$, $i = 2$, and $i = 3$, where $i = 1$ refers to the first available action of the user, $i = 2$ refers to the next, and $i = 3$ refers to the most recent action of the user. Each line in Figure 5(a) shows the AUC performance when each of these actions is used as a single feature for prediction. A larger area under the ROC curve corresponds to a better performance. We also include the AUC curve when all actions are included ($i = 1, 2, 3$). We can observe that the more recent an action is ($\text{AUC}_{i=1} = 0.690$, $\text{AUC}_{i=2} = 0.719$, and $\text{AUC}_{i=3} = 0.761$), the more accurate the prediction. This suggests that users who are more recently engaged with the Spark module are more likely to continue engaging with it. In other words, users who have clicked on Spark are easier to be enticed to explore its recommended entities when triggered. Alternatively, this may be related to the relatively frequent path navigation patterns observed by the users (see Table 4).

Effect of individual features (Q). We also dig further into the effect of specific features in Q, specifically Spark CTR ($Q3$), Spark views ($Q1$), query CTR category ($Q4$) and query type ($Q5$). The results are shown in Figure 5(b). Interestingly, we see that features related to Spark CTR are the most discriminating to predict the future click behavior of the user.

Demographics. We also experimented with demographic features (gender and age). The addition of these features did not add any significant improvement and as such we do not include them in our analysis. This was particularly surprising as we observed an effect of age in CTR. Note that in other contexts, such as in the case of predicting click-through rate for sponsored ads [7], demographics have been proven to be important indicators. The extent to which the fact that demographics do not help in the prediction task is caused by the coverage of Spark knowledge base should be investigated.

5.4 Discussion

We developed features based on the characteristics of queries and users that reflect the click behavior on related entities recommended by Spark given a query and a user who issues that query. These features were inspired by our analysis of CTR reported in Section 4. Using logistic regression, our results demonstrate that user-based features improve significantly the accuracy for the prediction task compared to using only query-based features. The main contribution for this improvement originates from the historical information on past user click behavior. We also demonstrate that the most significant action is the most recent one. Session length can help in the prediction, indicating that users who spend time on the search application may be in "explorative mood" and are more likely to interact with entities recommended to them by the Spark engine. However, in all cases, recall is relatively low showing that overall the particulars under which a user will engage with the Spark module and a recommended entity are diverse and cannot be captured easily. In all cases, predicting the negative instances is a relatively easier task than predicting the positive ones.

6. CONCLUSIONS

Many entity search systems have been proposed and evaluated through several evaluation initiatives in which the focus was on whether the proposed approaches were effective in returning the most relevant entities given a query. In this paper we also focus on evaluation, but with two main differences. First, we study which relationships promote explorative search exploiting the Spark system, which has been deployed large-scale as part of Yahoo Search, as a use case. Given a query submitted to Yahoo search engine, Spark displays related entity suggestions exploiting Yahoo knowledge graph. Secondly, we perform a large-scale evaluation of Spark, with real information needs that then translate into queries by actual users of Yahoo Search. Our focus is on the queries that trigger Spark to recommend entities.

Analyzing a sample of $2M$ users, we studied how users interact with results returned by Spark, in terms of submitted queries, user demographics, investigating also the interplay between typical search results and Spark recommendations. We looked at how users behave when presented search results, here entities. This resulted in a very different picture compared to studying user behavior when users were presented search results that were served to "solve" their information need. Our results show that Spark is able to promote an explorative behavior. Often, when users decide to engage with a Spark recommendation they end up navigating through the recommendations engaging in explorative search as our introductory example suggests, where users navigated from "Selena Gomez" to "Marlon Brando". Among other findings, we show that longer sessions result in a higher explorative activity and engagement, and that when users navigate through Spark they favour following paths of *different* entities rather than exploring multiple entities related to a *single* entity.

Based on our previous analysis, we design a prediction task aiming to predict when users will click on recommended entities and engage in explorative behavior. We develop a set of features based on the characteristics of queries and users that reflect the click behavior on related entities recommended by Spark given a query and a user who issued it. We showed that user-based features improves significantly the accuracy compared to using only query-based features. The main contribution for this improvement originates from the historical information on past user behavior related to their interaction with Spark results. We also demonstrated that the most significant feature, for predicting of a future click, is the most recent interaction. However, recall is relatively low showing that the specifics under which a user will interact with Spark are diverse and cannot be easily captured easily.

Future work. In this work, we focused on the interplay between search results and Spark recommended entities. Given the content-rich result pages of modern search engines, users will interact with other components of the page like its ads and other results such as images and so on. All these compete for user attention and it will be important to situate the explorative search experience promoted by such systems like Spark within the full search context, and not only with respect to the standard search results, as done in this paper. Also, Spark returns entities from a regularly updated KB. However, we do not study the effect of the coverage, i.e., which entities and relationships are stored in the KB. Future work will look into this issue and into the impact of displaying a more diverse or a larger set of entities to the user and how these elements impact their explorative behavior. Finally, our work focused on queries for which Spark returned entities as recommendations for users to explore. It will be important to study how doing so promotes exploration, compared to not returning any such recommendations.

7. REFERENCES

[1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[2] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[3] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity Recommendations in Web Search. In *The Semantic Web–ISWC 2013*, pages 33–48. Springer, 2013.

[4] R. Blanco, P. Mika, and S. Vigna. Effective and efficient entity search in RDF data. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I*, ISWC'11, pages 83–97, 2011.

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

[6] B. Carterette, E. Kanoulas, and E. Yilmaz. Simulating simple user behavior for system effectiveness evaluation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 611–620, 2011.

[7] H. Cheng and E. Cantú-Paz. Personalized click prediction in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 351–360. ACM, 2010.

[8] T. Cheng, X. Yan, and K. C.-C. Chang. Entityrank: searching entities directly and holistically. In *VLDB*, 2007.

[9] D. Downey, S. Dumais, and E. Horvitz. Models of searching and browsing: Languages, studies, and applications. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2740–2747, 2007.

[10] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 34–41, 2010.

[11] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[12] K. Haas, P. Mika, P. Tarjan, and R. Blanco. Enhanced results for web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 725–734, New York, NY, USA, 2011. ACM.

[13] M. Harvey, M. Wilson, and K. Church. Workshop on searching for fun 2014. In *Fifth Information Interaction in Context Symposium, IIiX '14, Regensburg, Germany, August 26-29, 2014*, 2014.

[14] A. Hassan and R. W. White. Task tours: Helping users tackle complex search tasks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1885–1889, 2012.

[15] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP*, 2011.

[16] L. Hollink, P. Mika, and R. Blanco. Web usage mining with semantic analysis. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 561–570, 2013.

[17] B. J. Jansen, D. L. Booth, and A. Spink. Determining the user intent of web search engine queries. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1149–1150, 2007.

[18] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, editors, *WWW*, pages 387–396. ACM, 2006.

[19] C. Kang, S. Vadrevu, R. Zhang, R. van Zwol, L. G. Pueyo, N. Torzec, J. He, and Y. Chang. Ranking related entities for web search queries. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, *WWW (Companion Volume)*, pages 67–68. ACM, 2011.

[20] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *KDD*, 2009.

[21] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman. Active objects: actions for entity-centric search. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *WWW*, pages 589–598. ACM, 2012.

[22] G. Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46, Apr. 2006.

[23] P. McCullagh and J. A. Nelder. Generalized linear models. 1989.

[24] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, 2007.

[25] P. Mika, E. Meij, and H. Zaragoza. Investigating the semantic gap through query log analysis. In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, *International Semantic Web Conference*, volume 5823 of *Lecture Notes in Computer Science*, pages 441–455. Springer, 2009.

[26] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM*, 2008.

[27] K. Radinsky, K. Svore, S. Dumais, J. Teevan, A. Bocharov, and E. Horvitz. Modeling and predicting behavioral dynamics on the web. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 599–608, 2012.

[28] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 43–52, 2008.

[29] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

[30] M. Slaney. Precision-recall is wrong for multimedia. *IEEE Multimedia*, 18(3):4–7, 2011.

[31] Y. Song, X. Shi, and X. Fu. Evaluating and predicting user engagement change with degraded search relevance. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1213–1224, 2013.

[32] K. Wang, T. Walker, and Z. Zheng. Pskip: Estimating relevance ranking quality from web search clickthrough data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1355–1364, 2009.

[33] R. W. White and S. T. Dumais. Characterizing and predicting search engine switching behavior. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 87–96, 2009.

[34] R. W. White and R. A. Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.

[35] X. Yin and S. Shah. Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th international conference on World wide web*, pages 1001–1010. ACM, 2010.

[36] X. Yuan and R. White. Building the trail best traveled: Effects of domain knowledge on web search trailblazing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1795–1804, 2012.