Behavioral dynamics from the SERP's perspective

What are failed SERPs and how to fix them?

Kiseleva, J.; Kamps, J.; Nikulin, V.; Makarov, N.

DOI
10.1145/2806416.2806483

Publication date
2015

Document Version
Final published version

Published in
CIKM'15

Behavioral Dynamics from the SERP's Perspective: What are Failed SERPs and How to Fix Them?

Julia Kiseleva[1] Jaap Kamps[2] Vadim Nikulin[3] Nikita Makarov[3]

[1]Eindhoven University of Technology, Eindhoven, The Netherlands
[2]University of Amsterdam, Amsterdam, The Netherlands
[3]Yandex, Moscow, Russian Federation
[1]j.kiseleva@tue.nl [2]kamps@uva.nl [3]{vnik,nkmakarov}@yandex-team.ru

ABSTRACT

Web search is always in a state of flux: queries, their intent, and the most relevant content are changing over time, in predictable and unpredictable ways. Modern search technology has made great strides in keeping up to pace with these changes, but there remain cases of failure where the organic search results on the search engine result page (SERP) are outdated, and no relevant result is displayed. Failing SERPs due to temporal drift are one of the greatest frustrations of web searchers, leading to search abandonment or even search engine switch. Detecting failed SERPs timely and providing access to the desired out-of-SERP results has huge potential to improve user satisfaction. Our main findings are threefold: First, we refine the conceptual model of behavioral dynamics on the web by including the SERP and defining (un)successful SERPs in terms of observable behavior. Second, we analyse typical patterns of temporal change and propose models to predict query drift beyond the current SERP, and ways to adapt the SERP to include the desired results. Third, we conduct extensive experiments on real world search engine traffic demonstrating the viability of our approach. Our analysis of behavioral dynamics at the SERP level gives new insight in one of the primary causes of search failure due to temporal query intent drifts. Our overall conclusion is that the most detrimental cases in terms of (lack of) user satisfaction lead to the largest changes in information seeking behavior, and hence to observable changes in behavior we can exploit to detect failure, and moreover not only detect them but also resolve them.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation, Relevance feedback, Search process*

Keywords: Query reformulation, Concept drift, Information retrieval.

1. INTRODUCTION

The information seeking behavior of users on the web is inherently sensitive to changes happening in world [25, 28]. As the web reflects the world around us, content is changing in predictable and unpredictable ways, affecting the search intent and queries issued by users. Added to that, searchers express their complex information needs in short queries, causing an inherent ambiguity in their statements of request: the query intent is specific to the context of the user and the point in time. It is a formidable achievement of modern search engines that they manage to keep up to pace with changing content, at equally formidable costs in crawling and updating search engines indexes. In particular, for updating rankers, click through information in interaction logs are crucial [2, 21]

Yet, there remain cases of failure where the organic search results on the search engine result page ($SERP$) are outdated, and no relevant result is displayed. This can be caused by temporal query intent drift, where the desired pages for a query are changing over time, and the historical transaction logs privilege the outdated results. For example, if users were searching for *'Malaysia airlines flight'* in March 2014 they most likely wanted to see news about the *Malaysian flight 370* that disappeared. However, if users issued the same query in July 2014 they mostly likely were searching for information about the *Malaysian flight 17* that is presumed to be shot down. Figure 1 shows daily Wikipedia page views for the MH17 and MH370 pages over 2014, with striking increases from 0 to 100s of thousands of page views when the events happened.

The Malaysian Airlines example can characterized as a "sudden" drift which may cause the $SERP$ to become outdated. Such changes can be associated with the news, and received the most attention in research community [12, 13]. However, changes may happening over a longer period of time and not necessarily bring an increase in the volume of traffic. For instance, if users issued the query *'CIKM conference'* in 2014 they were satisfied with results referring to the page http://cikm2014.fudan.edu.cn/ and this page gets a majority of clicks. However, the conference site has been changed and the same query issued in 2015 should be linked to the different page http://www.cikm-2015.org/. The CIKM example can be characterized as an "incremental" drift where the intent of the original query is changing over a longer period of time.

In this paper, we examine a generic approach to detect $SERPs$ that become out of sync with the query intent. Specifically, users issue a query Q and a search engine returns search result page ($SERP$) that is a ranked list of URLs:

$$SERP = (url_1, \ldots, url_i, \ldots, url_n).$$

Our users are expected to click on some url_i on the $SERP$ that satisfies their information need, and the order of URLs on the $SERP$ is based on various features and optimized to fit a history of user interactions with a pair $\langle Q, SERP \rangle$. As a result, the $\langle Q, SERP \rangle$ shown at a given point in time will reflect the user preferences over an earlier period of time. However, this gives no guarantee on the quality of the current $\langle Q, SERP \rangle$ as user preferences are sensitive to time and events happening in world. We aim to detect cases of $SERP$ failure due to a significant drift in query intent

CIKM'15, October 19–23, 2015, Melbourne, Australia.

ACM 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806483 .

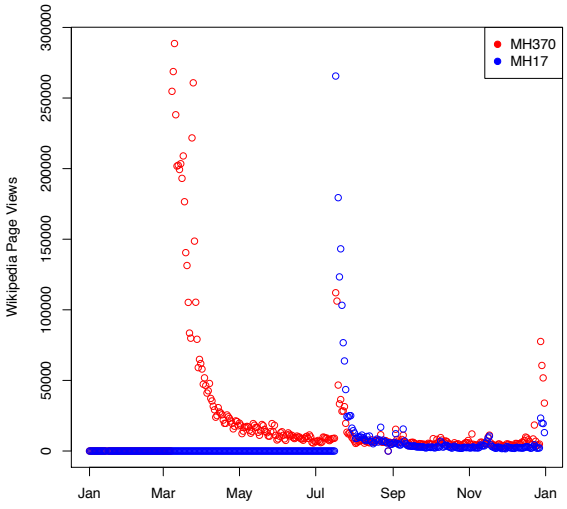

Figure 1: Wikipedia page views per day over 2014 for https://en.wikipedia.org/wiki/Malaysia_Airlines_Flight_17 and https://en.wikipedia.org/wiki/Malaysia_Airlines_Flight_370.

over time. Our aim is to detect failed $SERPs$ due to intent drift in an unsupervised way not relying on signals from other sources than the web traffic, language independent and not relying on rules or templates, independent of volume capturing both head and tail query drift. Hence we use behavioral signals as indicators of user (dis)satisfaction, such as click-through information [2, 22] and in particular query reformulations [1, 17, 24]. Specifically, in this paper, we are trying to answer the following main research question:

By analyzing behavioral dynamics at the SERP level, can we detect an important class of detrimental cases (such as search failure) based on changes in observable behavior caused by low user satisfaction?

We break up the main research problem into three different parts. Our first concrete research question is:

RQ1: *How to include the SERP into the conceptual model of behavioral dynamics on the web? How to identify (un)successful SERPs in terms of drastic changes in observable user behavior?*

We conduct a conceptual analysis of behavioral dynamics from the $SERP$'s perspective, and introduce failure and success at the SERP level, analyzing their behavioral consequences identifying indicators of success and failure. We then analyze success and failure in light of changing query intents over time, and identify an important case of $SERP$ failure due to query intent drift, and suggest an approach to detect a failed $SERP$ due to query intent drift by significant changes in behavioral indicators of failure.

Our second concrete research question is:

RQ2: *Can we distinguish different types of SERP failure due to query intent drift (e.g., sudden, incremental), and when and how should we update the SERP to reflect these changes?*

We study different types of possible query intent drift inspired by the literature on concept drift [15]: sudden, incremental, gradual and reoccurring. It is important to be able to classify the type of changes in user satisfaction because a sequence of actions a search engine should perform to normalize a situation can be different. We identify relevant parameters, such as the window of change, volume or popularity of queries, and relevant behavioral indicators, such as the probability of reformulation, abandonment rates, and click through rates. For the two main categories of intent drift, we define an unsupervised approach to detect failed $SERPs$. We also show how the detected changes can be used to improve a ranking of search results.

Our third concrete research question is:

RQ3: *How effective is our approach on a realistic sample of traffic of a major internet search engine?*

We validate our approach on twelve months of search interaction logs of a major commercial search engine. We run a simplified version of our algorithm on a massive transaction log, and detected pairs of $\langle Q, SERP \rangle$ suspected of failing due to drifting query intents. We investigate the accuracy of drift detection and the accuracy of the clicked URLs of the revision to include on the $SERP$ of the original query. We look at the effectiveness of our approach for both sudden and incremental changes in query intent, by varying the duration of the window to detect failed $SERPs$.

The remainder of this paper is organized as follows. Section 2 introduces earlier work on behavioral dynamics on the web, and behavioral indicators of user satisfaction focusing on the $SERP$ level. Then, Section 3 introduces the concept of $SERP$ success and failure, and outlines behavioral cues for their detecting $SERP$ becoming out of sync over time. Followed by Section 4 zooming in on different types of query drift causing failed $SERPs$, and outlining practical ways of detecting them. Finally, Section 5 reports on extensive experiments demonstrating the real-world utility of our approach.

2. RELATED WORK

In this section we will study related work, focusing on research on topic and concept drift, on the behavioral dynamics of the web, and on user satisfaction signals on the $SERP$ level.

2.1 Topic and Concept Drift

Topic or query drift has been studied for long in IR, usually in the context of evolving information needs as may happen in routing tasks [4], or the opposite negative effect of retrieving off-topic documents lower in the ranking [33]. In particular in adaptive filtering, topic models are continuously updated when new data comes available [5]. The focus is on a general topic or standing profile that monitors a stream of data and selects relevant documents. Our focus is on the SERP, serving results to a population of users with subtle or less subtle variation in query intent, taking into account changes in the query intent over time.

Topic drift is distinct from concept drift [15, 30, 36] which, in a machine learning setting, refers to changes in the conditional distribution of the output (i.e., target variable) given the input (i.e., input features), while the distribution of the input may stay unchanged. We will use a concept drift approach in the next sections, to model changes in features indicating lack of search satisfaction, and for determining thresholds for drift detection.

2.2 Behavioral Dynamics

The changes in query popularity over time have been studied extensively in prior work. Moreover, researchers have also examined the relationship between query behavior and events [27]. There are algorithms for identifying queries that are related to breaking news and for blending relevant news results into core search results [12, 26].

Prior work on behavioral dynamics is based on three factors: (1) on changes in query dynamics and in this case authors are concentrated on the 'head' queries [25, 28, 29, 31]; (2) on changes in web content dynamics and user interaction with dynamic content [25]; and (3) how information about changes can be used:

- to improve the ranking on the $SERP$ [10, 12, 13, 26]; and
- to improve query auto-completion [32].

Additionally, Kulkarni et al. [25] explored how queries, their associated documents, and the intents corresponding to the queries change over time. Radinsky et al. [29] have done an extensive studies how time-series analysis methods can be applied to predict dynamics on the web. Shokouhi [31] proposed using time-series decomposition techniques for identifying seasonal queries.

In summary, the prior studies cited above examine how general changes in content, specific content features, or query volume can be used to improve web search experience. Although much has been done to understand user web search behavior over time, few efforts have sought to construct underlying models to understand changes $\langle Q, SERP \rangle$ and even used to automatically fix the observed problems. We present the construction of models for behaviors over time that can explain observed changes in user satisfaction with $\langle Q, SERP \rangle$.

2.3 User Satisfaction

User satisfaction with the $SERP$ has been researched extensively. It is widely adopted as a subjective measure of search experience. User clicks are reasonably accurate on average to evaluate user satisfaction with pairs $\langle Q, SERP \rangle$ [2, 22], using click-through information. This user satisfaction scenario is successfully applied to navigational queries. It is called *query-level satisfaction*. However, we have to take into account the fact that user clicks are biased:

1. to the page position in the $SERP$ [9, 21];
2. to the quality of the page's snippet [37]; and
3. to the domain of the returned URLs [19].

Al-Maskari et al. [3] claim that the search scenario for informational queries is different. Users can run follow-up queries if they are unsatisfied with the derived results, and reformulations can lead users to the desired information. This scenario is called *task-level satisfaction* [12]. On the one hand, earlier research proposed different methods for identifying successful sessions. Hassan et al. [18] used a Markov model to predict success at the end of a task. Ageev et al. [1] exploited an expertise-dependent difference in search behavior by using a Conditional Random Fields model to predict a search success. On the other hand, separate researches are interested in situations when users are frustrated. Feild et al. [14] proposed a method for understanding user frustration with the pair $\langle Q, SERP \rangle$ based on query log and physical sensor features. Kiseleva et al. [24] showed how to automatically detect changes in user satisfaction using reformulation signal.

Earlier, White and Dumais [35] gave users difficult information seeking assignments and evaluated their level of dissatisfaction via query log features and physical sensors. They demonstrated that the prediction model gets the highest quality when it is built based on query log features. One type of user behavior that can be clearly associated with frustration is search engine switching. Guo et al. [16] showed that one of the primary reasons users switched their search engine was due to dissatisfaction with the results on the $SERP$. A recent study [20] shows a method to predict finer-grained, graded satisfaction levels. This paper significantly extends earlier work [24], analyzing behavioral dynamics at the $SERP$ level, and explaining how and why the changes are happening. In this work we propose a methodology to define a type of changes in user satisfaction and how this information can be used to improve a ranker.

Summarizing, in this section, we presented an overview of prior work on behavioral dynamics and user satisfaction on the web, with a special focus on the $SERP$ level. In the rest of the paper, we will study variations in user satisfaction with $\langle Q, SERP \rangle$ pair over time, starting with a conceptual analysis of success and failure at the $SERP$ level in the next section.

3. SUCCESS AND FAILURE AT THE SERP

In this section we will study RQ1: *How to include the SERP into the conceptual model of behavioral dynamics on the web? How to identify (un)successful SERPs in terms of drastic changes in observable user behavior?*

3.1 (Un)successful SERPs

We first introduce the notions of successful and unsuccessful SERPs as a conceptual model. Recall from the above that we looks at the pair $\langle Q, SERP \rangle$, with a *query* Q and a *search engine result page* ($SERP$) consisting of a ranked list of URLs in response to query Q. That is,

$$SERP_q = (url_1, \ldots, url_i, \ldots, url_n).$$

Let us further assume that queries are issued for a purpose and that the *intent* of query Q can be represented as a non-empty set of desired pages $INTENT_q$. For example, conceptually speaking a navigational query will have a singleton set $INTENT_q$, and an informational query will have a larger set of desired pages. Over a population of users there may be a distribution of intents, each giving rise to a different sets of desired pages, and it is straightforward to incorporate this into the conceptual model, but for simplicity and clarity we use a single set of desired pages here.

We define a successful and unsuccessful $SERPs$ in the following way:

DEFINITION 1. *(a) A $SERP_q$ is a* successful SERP *for query Q if and only if $\exists url_q \in INTENT_q$ such that $url_q \in SERP_q$.*
(b) A $SERP_q$ is a failed SERP *for query Q if and only if $\forall url_q \in INTENT_q$ such that $url_q \notin SERP_q$.*

A user issuing query Q may respond to the $SERP$ in different ways. One of the possible scenario of user interaction with the $SERP$, which is widely studied, is an event when users do not click on presented results. This case is called *search abandonment* that is known as a metrics of how successful a $SERP$ is. Research on search abandonment [7, 8, 11, 34] studied two primary abandonment cases: *bad abandonment* indicating user frustration and dissatisfaction; *good abandonment* suggesting satisfaction without needing to click. Assume we have a successful SERP in the sense of the conceptual definition above, and observe no clicked result, this suggests a case of good abandonment. Good abandonment is quite common in modern search engines because direct answers such as weather and stock quotes are returned for queries with explicit intent. Moreover, snippets can also satisfy users' information needs directly. However, if we assume a failed $SERP$, then a lack of clicked results suggests bad abandonment. Diriye et al. [11] report roughly equal fractions of good and bad abandonment, hence the abandonment rate is a secondary indicator of $SERP$ success or failure.

Table 1: SERP Success and failure

Behavior	Failed $SERP$	Successful $SERP$
No clicks	Bad abandonment	Good abandonment
Clicked result	$DSAT$ clicks	SAT clicks
Revised query	Negative reformulation	Positive reformulation

The other possible scenario is for users to interact with a retrieved $SERP$. Web search users often click on the $SERP$ and/or follow up with other queries. Many researchers have showed that clicks and reformulations can be used for a variety of tasks. However, clicks are usually considered to be as a positive sign [2, 22] to detect user satisfaction with the pair $\langle Q, SERP \rangle$. In the conceptual model, we can distinguish between satisfaction (SAT) and dissatisfaction of clicks based on the desired pages:

DEFINITION 2. *(a) A click on $url_i \in SERP_q$ for query Q is a* SAT click *if and only if $url_i \in INTENT_q$.*
(b) A click on $url_i \in SERP_q$ for query Q is a DSAT click *if and only if $url_i \notin INTENT_q$.*

It is an immediate corollary that there cannot be SAT clicks on a failed $SERP$, and that we can expect SAT clicks, but cannot exclude $DSAT$ clicks, on a successful $SERP$. A practical approximation to detect the difference in satisfaction is the use of dwell time, either with simple thresholds such as 30 seconds, or by advanced classification models [23].

Apart from consulting the results on the $SERP$, users may also decide to revise the query. Query reformulations have been used as indicator of search satisfaction [1, 17, 24]. Query revision may happen both in case of successful and unsuccessful $SERPs$. In case of the a successful $SERP$, for example after interacting with some relevant results, a user may refine her query to explore a further sub-topic or aspect of the query. In want of a better term, we call this type of revisions a *positive reformulation*. In case of an unsuccessful $SERP$, our frustrated user may opt to formulate her query for example by spelling out her information need more explicitly, in the hope to arrive at a successful $SERP$. We call this type of revision a *negative reformulation*.

Table 1 summarizes the relation between the concept of successful and failed $SERP$ and indicators of user satisfaction such as search abandonment, (dis)satisfied clicks, or query revisions. While the detection of failed and successful $SERPs$ in practice is non-trivial, the conceptual analysis allows us to simply assume the existence of abstract concepts like the set of desired pages, and clear up the exact meaning of core concepts and their dependencies and consequences.

3.2 Behavioral Dynamics of SERP Failure

We now look in detail at the impact of changing query intent over time on the $SERP$, and how this affect the pair $\langle Q, SERP \rangle$. Specifically, we look at the transition between the some time point or period t_i and a later time t_{i+1}:

$$\langle Q, SERP_q \rangle_{t_i} \rightarrow \langle Q, SERP_q \rangle_{t_{i+1}}.$$

Assume that at t_i we have a successful $SERP$, hence it contains at least one page satisfying the intent of query Q at that point in time. Due to a satisfaction click on a result, the ranker will reinforce the $SERP$'s content and like present the same organic results at t_{i+1}. Many queries such as navigational requests, are very stable and resulting in a successful $SERP$ at time t_{i+1}. However, there is also an important fraction of queries that has a changing intent due to something happening in the world, which may cause the $SERP_q$ to become unsuccessful at time t_{i+1}.

This requires detecting when a $SERP$ becomes out of sync due to changes in the query intent. There are of course subtle changes in query intent over time, leading to small changes in the click distribution with the $SERP$ for a Q as studied in previous work for updating rankers. But these do not lead to an unsuccessful $SERP$ as defined in the paper. Hence we aim to distinguish cases where the desired page is not part of the $SERP$, at least not part of top n of the organic ranking (e.g., the top 10 results).

There are cases when users are looking for a desired page that does not exist, either no longer exists or was not created or updated yet. For example, a newly created page with a winner or an outcome of an election, users are looking for the next version of iPhone, etc. Although even in these cases, there is usually a surrogate desired page that explains that the page doesn't exist, and may inform or speculate on the time when the information will become available.

3.3 Detecting Failed SERPs

Our analysis leads to the following scenario when user satisfaction at time t_i with $\langle Q, SERP \rangle$ turn into user frustration at t_{i+1} with the same $\langle Q, SERP \rangle$. In other words we aim to detect situations when at time t_i users were satisfied with a pair $\langle Q, SERP \rangle$ and at some moment in time t_{i+1} users are no longer satisfied with the same pair $\langle Q, SERP \rangle$, due to changes in the query intent for example due to some event happening in world.

We consider the following types of behavior BF_j on the $SERP$ as a sign of user frustration (lack of search satisfaction) with the $SERP$:

- BF_1: search abandonment;
- BF_2: query reformulation;
- BF_3: $DSAT$ clicks on a top-10 search results; and
- BF_4: SAT clicks on a low ranked search result (> 10).

The intuition of our approach is that, over a population of users issuing a query, if we see a *sufficient* amount of negative reformulations, $DSAT$ and low-ranked clicks, and bad search abandonment, then we flag the $SERP$ as *failed*, and use information about the ultimately clicked page to update the $SERP$ for the original query —hence avoid failure for future requests.

In order to satisfy a requirement about *sufficient* number we use the phenomenon of *concept drift* [15, 30, 36]. The *real* concept drift refers to changes in the conditional distribution of the output (i.e., target variable) given the input (input features), while the distribution of the input may stay unchanged. For our problem we can formally define *concept drift* between time point t_i and time point t_{i+1} as:

$$\exists BF_j : p_{t_i}(BF_j, \langle Q, SERP \rangle) \neq p_{t_{i+1}}(BF_j, \langle Q, SERP \rangle),$$

where p_{t_i} denotes the joint distribution at time t_i between the set of input variables $\langle Q, SERP \rangle$ and the target variable BF_j. The approach is explained in detail in the next section.

We not only intend to detect failure, but also to find and to inject the missing page to the $SERP$. In case of revisions with following SAT clicks, or low-ranked clicks, we have a clear indication of the "missing" page and boost it's ranking so that it will surface on for the original query's $SERP$ for future users issuing the same query.

Summarizing, in this section, we introduced the concept of a successful and failed $SERP$ and analyzed their behavioral consequences identifying indicators of success and failure. We then analyzed success and failure in light of changing query intents over

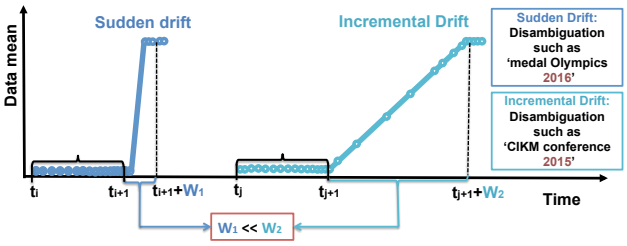

Figure 2: A representation of sudden and incremental types of the drifts.

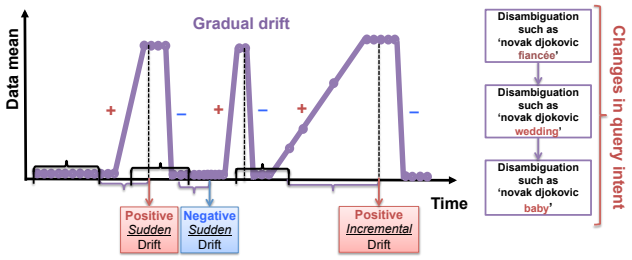

Figure 3: A representation of the gradual drift.

time, and identified an important case of $SERP$ failure due to query intent drift. This suggests an approach to detect a failed $SERP$ due to query intent drift by significant changes in behavioral indicators of failure. Our general conclusion is that more detrimental cases in terms of user satisfaction lead to larger changes in observable user behavior and hence more handles to detect them.

4. TYPES OF DRIFT IN USER SATISFACTION

In this section we will study RQ2: *Can we distinguish different types of SERP failure due to query intent drift (e.g., sudden, incremental), and when and how should we update the SERP to reflect these changes?*

4.1 Classifying Drift Type

This section proposes a method to detect the *types* of changes in user satisfaction. The detection of the type of change is important because it defines a strategy to fix a failed $SERP$ that is a result of changes in user satisfaction. We focus on two main types of the drifts: sudden and incremental, and will argue that the other types (i.e. gradual and reoccurring) can be represented as a combination of sudden and incremental types.

We use an increase in the query reformulations as a sign of user frustration with a shown $SERP$. The main criteria to distinguish between sudden and incremental drifts is the size of the testing window: (1) the sudden drift should be detected during the short period of time and (2) the incremental drift can be characterized by much longer testing period. An example is shown in Figure 2. Moreover, we are proposing a list of *secondary metrics* that can be used to characterize the drifts:

1. if the drift is related to the query popularity (i.e., 'head' or 'tail' queries);
2. if the volume of initial queries is changing a lot;
3. if search abandonment is observed frequently on the initial $SERP$s. In the context of an increasing number of query revisions, we will observe predominantly bad abandonment cases.

Let us characterize in details the types of changes we are studying in this work:

Sudden change. This kind of change gets the most attention in the literature [10, 12, 25, 26, 28] because they bring a most harmful and visible effect to user experience. This drift can be characterized by a growth of query popularity over a short period of time (e.g., 'breaking news queries'), as shown on the left hand side of Figure 2. In order to detect sudden drifts we use a short duration of the testing window (e.g., a couple of hours until a couple of days). Using the secondary metrics the sudden drift can be defined as:

1. the sudden drift is likely concerning more popular or 'head' types of queries;
2. the volume of an initial query Q is changing during the testing period (we also can detect a drift in an increase of Q volume);
3. search abandonment is a frequent behavior on the initial $SERP$ for Q indicating $SERP$ failure, i.e., the required url_q is missing (as presented in Definition 1).

Incremental change. This kind of change is less often studied and can be characterized by a slow change in query intent over a long period of time, as shown on the right hand side of Figure 2. An example it the reformulation of the query *'CIKM conference'* to include the specific year or location. This drift is a more difficult to detect because it does not necessary require an increasing query volume. However, changes in the fraction of query reformulations [24] can be used to detect incremental drift. Using the secondary metrics the incremental drift can be defined as:

1. the incremental drift is likely concerning less popular or 'tail' types of queries;
2. the volume of initial queries is not changing much during the testing period (we hardly can detect changes in an increase of volume);
3. search abandonment is a frequent behavior of initial $SERPs$ that means the initial $SERP$ is failed. Also in case a required URL is present at the $SERP$ but at a low rank, users tend to reformulate their query rather than explore further results on the $SERP$.

We identify two other types of query intent drift that can be represented as a combination of sudden and incremental types.

Gradual change. This is a different type of change that is presented in Figure 3. It can be viewed as a combination of the sudden and (or) incremental types of changes happening over time. For example, we consider the query *'novak djokovic'* (the famous tennis player) that may change its intents over time. For example, it has a drift in September 2013 on the term *'fiancée'* because the tennis player got engaged. Therefore, $SERP$ for the initial query is missing information about his fiancée and users tend to reformulate because they are interested in this topic.

At some moment in time the couple celebrate their wedding and users' interest changes again. Therefore the news about an engagement of famous tennis player becomes outdated and users start to reformulate the query *'novak djokovic'* using the reformulation *'wedding'* (if this information is missing from the $SERP$). As a logical continuation of this story the couple have a baby and users

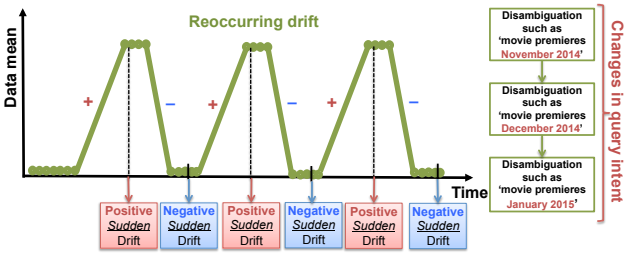

Figure 4: A representation of the reoccurring drift.

are interested to see information about this news, etcetera. It is important to note that the described changes are happening without any pattern, so some of these drifts may be sudden and some may be incremental.

Reoccurring change. The final special case of drifts is presented in Figure 4. A specific characteristic of the the reoccurring change is that it has a regular type of behavior [31]. The example in Figure 4 shows that users reformulate the query *'movies premieres'* regularly according dates. It is important to note that the reoccurring change is a combination of the same type drifts (sudden or incremental).

It is important to track changes for queries over time in order to understand if the $SERP$ is fixed when it is needed. A *positive* drift is typical for those cases where the reformulation signal is growing over time. Basically, the detected positive drift is a sign of a failing $SERP$ (shown as "+" in Figure 3). A *negative* drift is typical for those cases where the probability to reformulate a query is decreasing dramatically (shown as "-" in Figure 3). The *negative* drift may be interpreted in the two following ways:

1. the system has reacted to the positive change first and changed the $SERP$. Therefore a number of query revisions dropped down that means the problem with user satisfaction has been fixed;

2. the system has not reacted to the positive change in the reformulation signal but the moment has passed and users are no longer interested in a revision.

A detailed algorithm how to identify if a detected drift has positive or negative signs is presented in the next section.

4.2 Detecting Sudden and Incremental Drifts

Let us first define formally a set of features $\{F_j\}_{j=1}^{4}$ we use to detect changes in user satisfaction with the pair $\langle Q, SERP\rangle$: (1) a reformulation signal (RS), (2) a search abandonment signal (AS), (3) a query volume signal (VS), and (4) an average clicked position signal (CS):

- F_1: $RS(Q,Q')_{[t_i,t_{i+1}]}$ for the query Q and its reformulation Q' is a probability to reformulate Q to Q' within a particular time period $[t_i, t_{i+1}]$:

$$RS = P\left(Q \rightarrow Q'\right).$$

- F_2: $AS(Q)_{[t_i,t_{i+1}]}$ for the query Q is a probability to abandon (to give no clicks) $SERP_Q$ within a particular time period $[t_i, t_{i+1}]$.

- F_3: $VS(Q)_{[t_i,t_{i+1}]}$ for the query Q is a frequency of this Q within a particular time period $[t_i, t_{i+1}]$.

Algorithm 1 Algorithm for Detection the Type of Drift in user SATisfaction ($DTDSAT$). We leave out variables, i.e., $drift$ stands for $drift_{Q,Q',w}$, for readability.

Require: the train period $\Delta t = [t_i, t_{i+1}]$;
the test window $w = \{w_1, w_2\}$;
the error thresholds: $e_{RS}, e_{AS}, e_{CS}, e_{VS}$;
Ensure: $drift \leftarrow true, false$;
$drift\_positive \leftarrow true, false$;
$serp\_fail \leftarrow true, false$
for $\{Q, Q'\}_{k=1}^{N}$ **do**
  $\Delta_{RS} = \mu(RS_{\Delta t+w}(Q,Q')) - \mu(RS_{\Delta t}(Q,Q'))$
  if $|\Delta_{RS}| \geq e_{RS}$ **then**
    $drift \leftarrow true$
    if $\Delta_{RS} > 0$ **then**
      $drift\_positive \leftarrow true$
      $\Delta_{AS} = \mu(AS_{\Delta t+w}(Q)) - \mu(AS_{\Delta t}(Q))$
      $\Delta_{CS} = \mu(CS_{\Delta t+w}(Q)) - \mu(CS_{\Delta t}(Q))$
      $\Delta_{VS} = \mu(VS_{\Delta t+w}(Q)) - \mu(VS_{\Delta t}(Q))$
      if $(|\Delta_{AS}| \geq e_{AS}$ **or** $|\Delta_{CS}| \geq e_{CS})$ **then**
        $serp\_fail \leftarrow true$
        $url_Q \leftarrow getMissingTopURL()$
        $driftType \leftarrow getDriftType(|w|, \Delta_{VS})$
        $fixSerp(url_Q, driftType)$
      else
        $serp\_fail \leftarrow false$
      end if
    else
      $drift\_positive \leftarrow false$
    end if
  else
    $drift \leftarrow false$
  end if
end for
return $drift, drift\_positive, serp\_fail$

- F_4: $CS(Q)_{[t_i,t_{i+1}]}$ for the query Q is an average position clicked on $SERP_Q$ within a particular time period $[t_i, t_{i+1}]$.

We call F_1 as a primary drift metric of drift and $\{F_j\}_{j=2}^{4}$ as a list of secondary drift metrics. Each of them can be estimated straightforwardly based on observed frequencies in the period: for RS, we calculate the probability of reformulation per day, and for the period use the (observed) average over days (μ).

The proposed algorithm $DTDSAT$ to discover types of changes in user satisfaction is presented in in Algorithm 1. It is a straightforward application of the adaptive windowing algorithm from concept drift detection [6], which calculates a theoretically motivated threshold e for observing a significant drift based on a confidence value δ. We will first explain how $DTDSAT$ works.

DTDSAT Input. We assume that we can detect the *sudden drift* within a short period of time (w_1) such as from three days up to two weeks. In contrast, the *incremental drift* is detected on a larger time slot (w_2) from more than two weeks and up to one month. As the train window (Δt), we use fixed period of time for both considered types of drifts. We calculate an error thresholds: e_{RS}, e_{AS}, e_{CS}, e_{VS} using a standard method described in [6].

DTDSAT Output. The algorithm $DTDSAT$ returns an alarm as an output if the drift happens. Additionally, it produces an extra information about a detected drift: a sign: **(1)** the *'positive sign'* means we need to fix $SERP$; **(2)** the *'negative sign'* means users are no longer reformulating Q, so no action should be taken. A

collected sequence of positive and negative drifts for Q can be used to build a dynamics of changes for Q. This dynamics may help to understand if Q has a gradual drift (e.g., Figure 3) or a reoccurring one (e.g., Figure 4) over some longer period of time (e.g., 6 months or 1 year).

The Algorithm 1 includes the following methods:

- a method $getMissingTopURL()$ returns a missing url_Q as defined in Definition 1. It gets url_Q based on a statistics about the most frequently clicked $URLs$ after issuing drifted reformulation Q to Q'.
- a method $getDriftType(\Delta_{AS}, \Delta_{CS}, \Delta_{VS})$ takes into account a statistics about the secondary metrics of drifts in order to estimate a type of drift. There are cases when our system detects a drift on RS but none of the secondary metrics has changed. We called this situation a *'positive reformulation'*. In this case we are not dealing with failed $SERP$.
- a method *fixSerp(*url_Q*,driftType)* that will produce a list of $URLs$ that users mostly click on after running query revisions. The top $URLs$ from the list a candidates to be included on the $SERP$ served for the original Q, and avoid future user frustration and the need to revise their queries.

Summarizing, in this section, we studied different types of possible query intent drift inspired by the literature on concept drift [15]: sudden, incremental, gradual and reoccurring. We identified relevant parameters, such as the window of change, volume or popularity of queries, and relevant behavioral indicators, such as the probability of reformulation, abandonment rates, and click through rates. For the two main categories of intent drift, we define an unsupervised approach to detect failed SERPs caused by drift. We also showed how the detected changes can be used to improve a ranking of search results.

5. EXPERIMENTS AND RESULTS

In this section we will study RQ3: *How effective is our approach on a realistic sample of traffic of a major internet search engine?*

5.1 Experimental Data

Our experimental data comprises of massive raw and unfiltered search logs of a major commercial search engine that were collected during the whole 2014 year. Our audience consists of about *25 millions users per day*. Our traffic consists of approximately *150 millions of queries per day*. In our experimentation, we are dealing with a multilingual traffic that has at least *five* dominant languages.

5.2 Evaluation Methodology

We now describe our methodology to evaluate the quality of drifts detection algorithm.

Our algorithm is unsupervised, and detects drift in query intent based on a concept drift technique using a simple, theoretically motivated threshold, needing only a single linear pass through the data. To the best of our knowledge, there is no alternative approach to detect failed SERPs that could function as a baseline. We run our algorithm of Section 4 in a simplified form on the logs. First, we choose three fixed windows of 3, 7, and 14 days rather than calculate the optimal window based on the threshold. Second, we use the change in the probability of a revision (Δ_{RS}) as the criterion to select data. Third, we use a single threshold (e_{RS}) based on $\delta = 0.1$, with values in the range of $[0.2, 0.5]$, as we did not observe major differences between the settings.

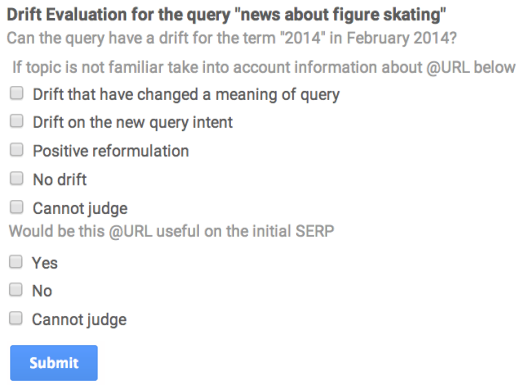

Figure 5: A fragment of an evaluation task for the annotators. We suggest the most clicked *url* after a query revision.

As an approximation, we define the drift type by the size of the test window, so a three days windows size is related to a sudden drift type. However, it is important to note that a fresh intent classifier (based on mostly a query popularity) is already working within the search engine. Let us call it FIC. Therefore, most popular changes in query intents might picked up by FIC and $SERP$ is already fixed for *'head queries'*. As was shown in [26] this can be done within very short period of time.

For evaluation, we selected *randomly* about 150 examples from different batches. Therefore, in total, we selected 450 examples of drifts for test set which we use to report the final results. Each detected drift in our test set was evaluated by three judges and we report overall scores. As a evaluation metrics we use accuracy rates. In the current settings we are more interested to obtain rather precise results.

In order to evaluate the obtained results we set up the following evaluation task. Every judge is supplied with the definition of sudden and incremental types of drift. We gave to judges the following explanation for the drift labels:

1. Drift is detected: *'real' drift* in users intent i.e. new target intent replaces the old target intent: e.g. Q = *'referendum in the Crimea'* has a drift on its revision Q' = *'referendum in the Crimea 16 march'* in March 2014 (due to some events happening in the world); similarly Q = *'Sochi'* has a drift on its revision Q' =*'Sochi **2014**'* in February 2014 (due to Olympics games that took place in the city Sochi in February 2014);
2. Drift is detected: *'new drift'* in users intent i.e. an another/new target intent added to a multifaceted query: e.g. Q = *'Happy New Year wishes'* has a drift on its revision Q' = *'Happy New Year wishes **2015**'* in December 2014 (due to the fact that people are trying to find the next year); similarly Q = *'RoboCop'* has a drift on its revision Q' = *'RoboCop **2014**'* in February 2014 (due to a release of the new movie);
3. Positive reformulation is detected: it signals about a shift in users intent but we suppose that $SERP$ is not broken in this case: e.g. Q = *'schedule of matches of the World Cup 2014'* has a detected positive reformulation Q' = *'schedule of matches of the World Cup 2014 **on tv**'* in June 2014; similarly Q = *'voice 22.11.2014'* has a detected positive re-

Table 2: Accuracy of drift detection (including positive reformulations)

	Window	Drift_Accuracy	URL_Accuracy
1	3 days	0.58	0.87
2	7 days	0.66	0.97
3	14 days	0.91	0.91
4	Combined	0.72	0.92

formulation $Q' =$ *'voice 22.11.2014* ***watch online****'* ('Voice' a a popular TV show);

4. It is not a drift;

5. There is not enough information to judge.

In order to evaluate a quality of our procedure to fix failed $SERPs$ we asked judges to check the suggested $@url$ and answer the question *'Would be this* $@url$ *useful on the initial* $SERP$ *for the query* $@Q$ *at the particular moment in time* $@\Delta t$*?'*. Judges were supplied with the following set of labels:

1. *Yes*, it would be useful to insert the $@url$ to the $SERP$ for the $@Q$ during the time period $@\Delta t$;

2. *No*, it does not make sense to insert the $@url$ to the $SERP$ for the $@Q$ during the time period $@\Delta t$;

3. There is not enough information to judge.

The interface for our labeling procedure is presented in Figure 5. As it turned out, judges were struggling to distinguish between the first two categories of Figure 5, hence we decide to collapse these two categories into a single case of drift due to a failed $SERP$, consistent with the algorithm in Section 4.

5.3 Experimental Results

We detect 100s of thousands of revisions over a whole year, and over 200,000 unique $\langle Q, Q' \rangle$ pairs. This is considerable number, but of course still a small fraction of the overall traffic. The set of revision terms is rather varied, with a revision term occurring in 3-4 unique pairs of $\langle Q, Q' \rangle$. Familiar patterns like 'year' revisions (i.e., '2014' or '2015') account just a around 2-3 percentage of the revisions, and around 17-18 percent contains any number. This suggests we capture a wide variety of revisions, beyond those that could be detected based on rules and templates. For the non-numerical revisions, we also see queries and revisions in many languages, show-casing the general applicability of the unsupervised approach.

Table 2 shows the results of the drift detection approach for three test windows with 3 days, 7 days, and 14 days, where we look at all cases of drift (failed $SERP$ and positive reformulations) versus the 'no drift' judgment, and on the judgment on the utility to include the URL clicked after revisions on the $SERP$ of the original query.

We observe a 72% accuracy for the drift detection and a 92% accuracy for the usefulness of the URL to be included on the original page's $SERP$. While the detection doesn't work flawless, these accuracies are a clear indicator of the value of the approach to detect failed $SERPs$. To put this performance into perspective, these number are based on the simplified algorithm based on the probability of query revisions and the theoretical threshold, rather than optimized tuning. The high levels of accuracy for the picked up $URLs$ confirm that these are of interest to be included on the $SERP$ of the original query.

Table 3: Accuracy of failed SERP and positive reformulation detection

	Window	Drift (Failure)	Pos_Reformulation
1	3 days	0.17	0.41
2	7 days	0.41	0.20
3	14 days	0.80	0.11
4	Combined	0.47	0.25

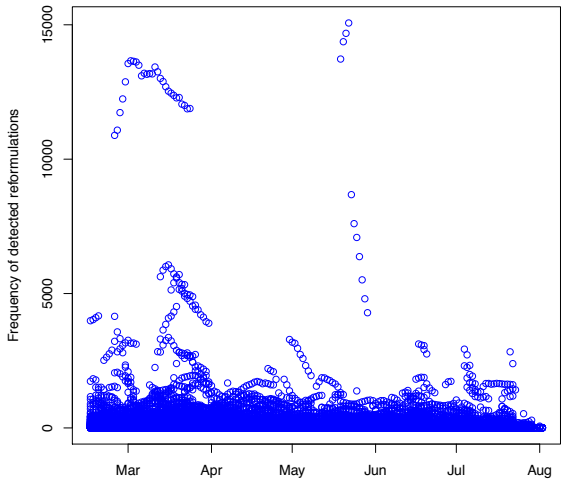

Figure 6: Frequency of detected reformulations over time.

Looking at the breakdown over the duration of the test windows, we see a considerable increase in accuracy for the longer test periods, reaching up to 91% accuracy for the 14 day window. This leads to two observations. First, the approach seems to benefit from more observations to make more reliable judgments, and a revision pattern observed over two weeks is obviously a clearer signal. This also suggests the value of our approach for the detection of incremental change. Second, we expected to obtain high accuracy for popular or head queries in the 3 day window, but observed mostly queries with 1-3 revisions per day. A plausible explanation is that these are picked up and corrected by the $SERP$ already, as recency ranking tools are employed in the search engine that can respond within hours. A possible resolution is to use smaller and adaptive windows, as is done in the concept drift literature, or defining window size in terms of an absolute minimum number of revisions leading to very short window sizes for popular queries.

In Table 3 we break down the two cases of drift: due to a genuine shift in intent toward a new direction, hence indicating a failed SERP, or due to the exploration of another aspect or facet of the original request beyond what's presented on the initial SERP. We observe 47% accuracy for the failed SERP detection relative to all detected $\langle Q, SERP \rangle$ pairs, hence the remaining 25% are positive reformulations. In particular in these 47% of the cases it would be important to consider including the URL clicked after revision into the $SERP$ of the original query. As observed above, the detection accuracy goes significantly up over the larger duration of the detection windows. Over 14 days, no less than 80% of the detected cases indicate a failed $SERP$ due to query intent drift. As observed above, our approach is very effective to detect cases of incremental drift, but less effective to pick up sudden drift over the shortest period.

In this section, we limited ourselves by the varying time windows of detection due to patterns of sudden and incremental drifts. Figure 6 shows the frequencies of detected query reformulations over time. What we observe is that we detect the same drifts on consecutive days, but also that the revisions may disappear after a period of time. Anecdotal evidence suggests that this can be both due to another drift in query intent, for example for revisions specific to events or months of the year, or due to updates of the SERP served for the original query. This supports to importance of detecting both positive and negative drift patterns, and also look at gradual and reoccurring drifts.

Summarizing, in this section, we ran a simplified version of our algorithm on a massive transaction log, and detected over 200,000 pairs of $\langle Q, SERP \rangle$ suspected of failing due to drifting query intents. We observed a reasonable accuracy of drift detection (72%) and a high accuracy of candidate URLs to be included on the $SERP$ of the original query. For incremental change over the longer detection period of 14 days, we detected failed $SERPs$ due to query intent drift with an 80% accuracy. Under the specific conditions of the recency optimized search engine, the performance for detecting sudden change over shorter periods was less effective.

6. CONCLUSIONS

This paper investigated how the dynamic nature of web content and user intents have consequences for the SERP to be displayed for a particular query. In particular, there remain cases of failure where the organic search results on the search engine result page ($SERP$) are outdated, and no relevant result is displayed. This can be caused by temporal query intent drift, where the desired pages for a query are changing over time, and the historical transaction logs privilege the outdated results. Our main research question was: By analyzing behavioral dynamics at the SERP level, can we detect an important class of detrimental cases (such as search failure) based on changes in observable behavior caused by low user satisfaction?

We presented an overview of prior work on topic and concept drift, behavioral dynamics, and user satisfaction on the web, with a special focus on the $SERP$ level. We conducted a conceptual analysis of success and failure at the SERP level in order to answer our first research question: How to include the SERP into the conceptual model of behavioral dynamics on the web? How to identify (un)successful SERPs in terms of drastic changes in observable user behavior? Specifically, we introduced the concept of a successful and failed SERP and analyzed their behavioral consequences identifying indicators of success and failure. By analyzing success and failure in light of changing query intents over time, we identified an important case of SERP failure due to query intent drift. This suggested an approach to detect a failed SERP due to query intent drift by significant changes in behavioral indicators of failure.

We continued our analysis of different types of drifts in query intent over time, answering our second research question: Can we distinguish different types of SERP failure due to query intent drift (e.g., sudden, incremental), and when and how should we update the SERP to reflect these changes? Inspired by the literature on concept drift [15], we studied different changes in query intent: sudden, incremental, gradual and reoccurring, and identified relevant parameters, such as the window of change, volume or popularity of queries, and relevant behavioral indicators, such as the probability of reformulation, abandonment rates, and click through rates. For the two main categories of intent drift, we define an unsupervised approach to detect failed SERPs caused by drift, requiring only a single pass through a transaction log.

Finally, we ran experiments on massive raw search logs, answering our third research question: How effective is our approach on a realistic sample of traffic of a major internet search engine? We ran a simplified version of our algorithm and detected over 200,000 pairs of $\langle Q, SERP \rangle$ suspected of failing due to drifting query intents, observing a reasonable accuracy of drift detection (72%) and a high accuracy of candidate URLs to be included on the SERP of the original query. For incremental change over the longer detection period of 14 days, we detected failed SERPs due to query intent drift with an 80% accuracy, but under the specific conditions of the recency optimized search engine the performance for detecting sudden change over shorter periods was less effective.

As future work, we are further developing the conceptual model, and are running further offline experiments exploring further window sizes, and further features of user dissatisfaction. We are also planning to do online evaluation of how the discovered drifts are useful for fixing SERPs. In addition to the unsupervised methods of this paper, we are also experimenting with tuning the optimal parameters and theshold based on behavioral features and initial results suggest futher improvements.

Real data is messy and has many intricate dependencies, such as continually changing ranking, personalization, customization and localization, and specific tools to update the ranker fast on other signals (i.e., recency ranking). This makes data-driven research a difficult enterprise, and we strongly feel that this should be coupled with theoretical and conceptual analysis. We made a first attempt at this in the current paper, where we conduct conceptual analysis to clarify the meaning of core concepts and their relations and dependencies. And as a conceptual model, work with an idealized model that abstracts away from other factors outside the scope of our interest. For example, we observed in the experimental data relatively few or popular queries as those are tackled within hours by recency ranking methods. We view the experimental part more as initial validation experiments, mostly used to inform the conceptual model as well as identify the most useful features in the context of real world traffic. For this reason we did not "optimize" for the data using supervised methods, but collected a single set of data for three time windows and analyzed this to assess the value of the variables in the conceptual model, and to further develop our model. We strongly belief that conceptual and experimental research should go hand in hand, and without denying the value of "things that work in practice" we should put equal value on experiments that contribute to our conceptual or theoretical understanding.

Acknowledgments

This research has been partly supported by STW and it is the part of the Context Aware Predictive Analytics (CAPA) project.

References

[1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: a game for modeling different types of web search success using interaction data. In *SIGIR*, 2011.

[2] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26, 2006.

[3] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between ir effectiveness measures and user satisfaction. In *SIGIR*, pages 773–774, 2007.

[4] J. Allan. Incremental relevance feedback for information filtering. In *SIGIR*, pages 270–278, 1996.

[5] A. Arampatzis and A. van Hameran. The score-distributional threshold optimization for adaptive binary classification tasks. In *SIGIR*, pages 285–293, 2001.

[6] A. Bifet and R. Gavaldá. Learning from time-changing data with adaptive windowing. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, 2007.

[7] A. Chuklin and P. Serdyukov. How query extensions reflect search result abandonments. In *Proceeding of SIGIR*, pages 1087–1088, 2012.

[8] A. Chuklin and P. Serdyukov. Good abandonments in factoid queries. In *Proceeding of WWW (Companion Volume)*, pages 483–484, 2012.

[9] N. Craswell, O. Zoeter, M. J. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM*, pages 87–94, 2008.

[10] N. Dai, M. Shokouhi, and B. D. Davison. Learning to rank for freshness and relevance. In *Proceeding of SIGIR*, pages 95–104, 2011.

[11] A. Diriye, R. White, G. Buscher, and S. T. Dumais. Leaving so soon?: understanding and predicting web search abandonment rationales. In *Proceeding of CIKM*, pages 1025–1034, 2012.

[12] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, and C. L. F. Diaz. Towards recency ranking in web search. In *WSDM*, pages 11–20, 2010.

[13] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceeding of WWW*, pages 331–340, 2010.

[14] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *SIGIR*, pages 34–41, 2010.

[15] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

[16] Q. Guo, R. W. White, Y. Zhang, B. Anderson, and S. T. Dumais. Why searchers switch: understanding and predicting engine switching rationales. In *SIGIR*, pages 335–344, 2011.

[17] A. Hassan and R. W. White. Personalized models of search satisfaction. In *CIKM*, pages 2009–2018, 2013.

[18] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *WSDM*, pages 221–230, 2010.

[19] S. Ieong, N. Mishra, E. Sadikov, and L. Zhang. Domain bias in web search. In *Proceeding of WSDM*, pages 55–64, 2012.

[20] J. Jiang, A. H. Awadallah, X. Shi, and R. W. White. Understanding and predicting graded search satisfaction. In *Proceeding of WSDM*, 2015.

[21] T. Joachims. Optimizing search engines using clickthrough data. In *Proceeding of KDD*, pages 133–142, 2002.

[22] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, pages 154–161, 2005.

[23] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of WSDM*, pages 193–202, New York, NY, USA, 2014. ACM.

[24] J. Kiseleva, E. Crestan, R. Brigo, and R. Dittel. Modelling and detecting changes in user satisfaction. In *Proceeding of CIKM*, pages 1449–1458, 2014.

[25] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *WSDM*, pages 167–176, 2011.

[26] D. Lefortier, P. Serdyukov, and M. de Rijke. Online exploration for detecting shifts in fresh intent. In *Proceeding of CIKM*, pages 589–598, 2014.

[27] K. Radinsky, S. Davidovich, and S. Markovitch. Predicting the news of tomorrow using patterns in web search queries. In *WI*, 2008.

[28] K. Radinsky, K. Svore, S. T. Dumais, J. Teevan, A. Bocharov, and E. Horvitz. Modeling and predicting behavioral dynamics on the web. In *WWW*, pages 599–608, 2012.

[29] K. Radinsky, K. M. Svore, S. T. Dumais, M. Shokouhi, J. Teevan, A. Bocharov, and E. Horvitz. Behavioral dynamics on the web: Learning, modeling, and prediction. *ACM Transactions on Information Systems (TOIS)*, 31(3):16, 2013.

[30] J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In *AAAI*, 1986.

[31] M. Shokouhi. Detecting seasonal queries by time-series analysis. In *SIGIR*, pages 1171–1172, 2011.

[32] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *SIGIR*, pages 601–610, 2012.

[33] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. Predicting query performance by query-drift estimation. *TOIS*, 30:11:1–11:35, 2012.

[34] Y. Song, X. Shi, R. White, and A. H. Awadallah. Context-aware web search abandonment prediction. In *Proceeding of SIGIR*, pages 93–102, 2014.

[35] R. W. White and S. T. Dumais. Characterizing and predicting search engine switching behavior. In *CIKM*, pages 87–96, 2009.

[36] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning (ML)*, 23(1):69–101, 1996.

[37] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *WWW*, pages 1011–1018, 2010.