Contextual Search and Exploration

Julia Kiseleva[1(✉)], Jaap Kamps[2], and Charles L.A. Clarke[3]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
julianakiseleva@gmail.com
[2] University of Amsterdam, Amsterdam, The Netherlands
[3] University of Waterloo, Waterloo, Canada

Abstract. Personalized (mobile) devices are radically changing information access tools, with rich context allowing for far more powerful, personalized search. Rather than retrieving a "document" on the topic of a "query," the rich contextual information allows for tailored search and recommendation, and solve user's complex tasks by taking into account complex constraints, exploring options, and combining individual answers into a coherent whole. This paper reports on a RuSSIR 2015 course covering the challenges of contextual search and recommendation, with a concrete focus on the venue recommendation task as run as part of TREC 2012–2015. It consisted of both lectures and hands-on "hackathon" sessions with data derived from the TREC task.

1 Introduction

The ubiquitous availability of information on the web and personalized (mobile) devices has a revolutionary impact on modern information access, challenging both research and industrial practice. Searchers with a complex information need typically slice-and-dice their problem into several queries and subqueries, and laboriously combine the answers post hoc to solve their tasks. Rich context allows for far more powerful, personalized search, without the need for users to write long complex queries. Consider planning a social event at the last day of RuSSIR, in the unknown city of Saint Petersburg, factoring in distances, timing, and preferences on budget, cuisine, and entertainment. Rich context and profiles in combination with a curated set of web data allow us to solve complex tasks with just a simple query: `entertain me`. Rather than retrieving a "document" on the topic of a "query," the rich contextual information allows for tailored search and recommendation, and solve their complex task by taking into account complex constraints, exploring options, and combining individual answers into a coherent whole.

This RuSSIR 2015 course covered the challenges of contextual search and recommendation, with a concrete focus on the venue recommendation task as run as part of TREC 2012–2015 Contextual Suggestion Track. It consisted of both lectures and hands-on "hackathon" sessions with data derived from the TREC task. Our goal was enabling students to understand the challenges and opportunities of contextualized search over entities, and learn effective approaches for

P. Braslavski et al. (Eds.): RuSSIR 2015, CCIS 573, pp. 3–23, 2016.
DOI: 10.1007/978-3-319-41718-9_1

the concrete application to venue recommendation domain, as well as obtain hands-on experience with developing and evaluating personalized search and recommendation approaches.

The rest of this paper is structured as follows. After this introduction, Sect. 2 gives an overview of approaches to contextual search and exploration, focusing on venue recommendation. Next, Sect. 3 details how to set up an experiment to evaluate contextual suggestion based on the TREC track. Section 4 provides detailed approaches of to using contextual information in modeling search and interaction behavior. Then, Sect. 5 discusses the setup and results of the hackathon. Finally, Sect. 6 concludes the paper with some discussion on the outcome of the lectures and hackathon.

2 Approaches to Contextual Search and Exploration

In the first session, Jaap provided an overview of the tutorial and hackathon, and introduced various approaches to contextual search and exploration. It is motivated by complex search tasks now requiring several independent searches and put the onus on the user to manage the overall task progress, and combine individual results into a coherent whole.

As explained before, the official goal of the course was to enable students to understand the challenges and opportunities of contextualized search over entities, and learn effective approaches for the concrete application to venue recommendation domain. The unofficial goal, however, was to have the students plan our time in St. Petersburg. The lecturers wanted to visit an amazing city but are clueless about what to do, and invented a course so the students attending RuSSIR will be planning our holiday in St. Petersburg! A special edition of the TREC Contextual Suggestion Track's batch task was run as a hackathon, with profiles of Charlie, Julia and Jaap, and 102 candidate venues in St. Petersburg (Palaces, Museums, Restaurants, Bars, Clubs, etc.) to visit. We asked the students to build a system that gives us the best venues to visit after the lectures.

2.1 Slogan #1: Standard IR Fails for Venue Recommendation!

The course focused on contextual search and exploration, with a planning problem as leading example. The overall goal is to address complex information needs on mobile devices, using rich contextual information and user profiles, and taking into account complex constraints, exploring options, and combining individual answers into a coherent whole. The specific focus is on venue or point of interest (POI) recommendation for travelers, e.g., Canadian and Dutch people in St. Petersburg in August. What are we going to do this evening? How to plan what to do in an unknown city? What to see? Where to eat? Where to drink? The most popular things? Or those that I like best? Is there actually a ballet performance tonight? How do I get from venue 1 to venue 2 to ...?

The venue recommendation problem gets as input: (1) a *start signal* such as an App click or generic query; (2) a *context* such as a location or city; and

(3) a *profile* of the user, containing explicit profile information such as age and gender, and implicit profile information such as likes/dislikes in other cities that can be derived from earlier interactions on a phone.

There are many travel sites online, including Tripadvisor, Foursquare, Yelp, Google Places, Yandex Cities, etc. Most of these sites offer venue search with some level of support for the context (typically the data is organized by location equated by city or country/region), and essentially no support for the profile (typically very limited personalization/customization).

Standard search is not getting us very far: there is no query or statement of request in the traditional sense, and just using a city name or venue type as query leads to very poor results, unless the context and profile are taken into account. This leads to our first slogan: *Standard IR Fails for Venue Recommendation!*.

2.2 Slogan #2: Location Is Context

Venue recommendation isn't the same as geographical or location based search. Geo search exists for a few decades within IR. It is typically using a selection of typical search engine queries, focusing on those queries where part of the query is, or has, a location. For example, think of a query like "`restaurant in beijing china`." Each of the queries tends to have an exact answer, which is the same for anyone issuing the query, e.g., the query "`taj mahal`" linking to http://www.tajmahal.gov.in/.

Benchmarks on Geo search include the Geo IR tracks at CLEF[1] and the Geo-Time task at NTCIR.[2] Approaches to Geo IR typically use special resources or knowledge bases, with explicit locations like cities and countries, or POIs with GPS coordinates. The task is mostly about identifying the location part of the query, and mapping it to these resources, and search engines provide APIs for this.

Venue recommendation is different from Geo search. In venue recommendation, the query is a normal generic query without a location, e.g., "`restaurant`," "`bar`," or "`museum`." But the result should take the location into account: location is the *context* of the request, and venues too far will never be relevant. So a different context means an entirely different result set. This leads to our second slogan: *Location is Context*.

2.3 Slogan #3: Need to Blend Search and Recommendation

Venue recommendation isn't the same as collaborative recommendation. Work on recommendation is dominated by collaborative filtering. Here the input is a large set of ratings by many people, and the profile of a person x. The output is a ranked list of items y unrated by x, that x will rate high, based on people similar to x giving high ratings to y. There are many approaches to recommendation, the standard collaborative filtering approach treats each person as a vector of

[1] GeoCLEF 2005–2008, see: http://www.clef-initiative.eu/track/geoclef.
[2] NTCIR 8–9, 2010–2011, see: http://metadata.berkeley.edu/NTCIR-GeoTime/.

ratings: like/dislike/unknown, and looks at similar persons by cosine similarity over these vectors. The person most similar to person x is x herself, so we pick the next most similar person. Clustering and machine learning approaches are used to learn patterns in the training data, and to make predictions on unseen data.

Venue recommendation is different from collaborative recommendation. Collaborative recommendation assumes rich profiles of many users, but suffers from cold start problems: new users without history, and very sparse profiles. Most e-commerce providers see their users a few times a year, and have a continuous cold start problem. Hence we need to factor in search or content based recommendation.

We need aspects from both search and recommendation. Venue recommendation is not just serendipitous recommendation, such as a random book you like, but focused on a specific information need. But there is also no explicit query to match, such as when querying to look up the location of a particular known venue, but it is initiated by a generic query ("`st. petersburg`") or App click. This leads to our third slogan: *Need to Blend Search and Recommendation.*

2.4 Slogan #4: Search Is Getting Personal

There is no one size fits all approach to venue recommendation. Contextual search and recommendation requires a radical departure from the query-response paradigm of prototypical search, which takes as input a short query, and outputs a ranked list of results. This approach is still dominating research and industrial practice, with current systems excelling at short narrow scoped queries, heavily optimized against log data.

In terms of user satisfaction and user experience, this is likely a local optimum, where we cannot break out without changing something more fundamental. This implies that we need to step away from this "ten blue links" approach, and think about ways to support the user's whole search task. Currently, the emerging intelligent personal assistants come closest to this: Google Now, Microsoft Cortana, Apple's Siri, Facebook's M, etc. are starting a new search paradigm where context and profile information is key.

Your phone knows you, may know more about you than you know yourself: your work moved online into the clouds, and your personal life moved as well—everything you ever did is there... This data is personal, but also highly curated with clear entities and structure, allowing for powerful graph search with highly expressive queries.

We need to go beyond the query-response paradigm. This is not about personalization in terms of slightly changing the ranking by swapping some results, but an extreme form of personalization where different users get fundamentally different results: the profile is determines your result set—*you* are the query. This leads to our fourth slogan: *Search is Getting Personal.*

2.5 Wrap Up

We discussed venue recommendation as a personalized and contextualized task with complex constraints. Location is only part of the problem: it is not the same as geographical search. Profiles matter but are sparse: it not the same as collaborative recommendation. It is a form of extreme personalization: it cannot be handled by a one-size-fits-all approach. Sessions are highly interactive complex search going beyond the traditional query-response paradigm.

In the next section, we discuss a simplified form of venue recommendation for which a benchmark evaluation is being developed at TREC.

3 The TREC Contextual Suggestion Track

In the second 90-min session, Charlie provided an overview of the TREC Contextual Suggestion Track[3], which creates open data collections for evaluating contextal suggestion and point-of-interest recommendation. Since 2012 [9,10,14], the Contextual Suggestion Track has operated as part of the TREC[4] series of evaluation experiments, sponsored by the U.S. National Institute of Standards and Technology. The track imagines a traveler in a unknown city seeking sites to see and things to do that reflect his or her own personal interests, as inferred from their interests in their home city. For example, a group of information retrieval researchers visiting Saint Petersburg in August, such as the authors of this tutorial, should be directed to museums, restaurants, and bars that reflect their individual tastes. According to the Second Strategic Workshop on Information Retrieval in Lorne [4]:"*Future information retrieval systems must anticipate user needs and respond with information appropriate to the current context without the user having to enter an explicit query...*" The TREC Contextual Suggestion Track establishes an evaluation framework allowing researchers to investigate this problem, at least within the limited domain of point-of-interest recommendation.

The tutorial session began with an overview of task as it operated from 2012 to 2014 [9,10,14]. As input to the task, participating research groups were given a set of profiles, a set of example suggestions, and a set of contexts. Each profile corresponded to a single user, indicating that users preference with respect to each example suggestion, while each context represented a target city that the user might visit. For each profile/context pairing, participating researchers were required to return a ranked list of 50 proposed suggestions. Each suggestion was expected to be appropriate to the profile (based on the user's preferences) and the context (according to the target city). Profiles correspond to the stated preferences of real individuals, primarily recruited through crowdsourcing. These crowdsourced workers first judged example attractions in seed locations, representing their home cities, and later returned to judge suggestions proposed by the participating research groups for various target cities.

[3] See: http://sites.google.com/site/treccontext/.
[4] See: http://trec.nist.gov/.

Most of this overview was drawn from track reports, which can be consulted for detailed information [9,10,14]. In the remainder of the session we discussed a number of issues related to the structure of the track, as detailed below, as well as the lessons learned from it. The tutorial ended with a discussion of ongoing and future work.

3.1 Issue #1: Assessor Quality

The first of these issues concerns the quality of assessment provided by crowdsourced workers, who are not real travelers. Can we assume that these workers will provide judgments that accurately and consistently reflect their own opinions? We discussed ways in which worker/assessor quality can be measured, given that the degree to which an assessor likes or dislikes a point-of-interest attraction is purely a subjective question. We cannot simply look at agreement between assessors to determine assessment quality, as we would do for a traditional TREC retrieval task.

Each assessor has the implicit goal of ordering the systems according to their true ranking Thus, we measure assessor consistency by comparing the system ranking implied by the judgments of a single assessor with the average system ranking implied by the judgments of all assessors. In the tutorial, we examined the results of studying assessor consistency over TREC 2013 results [11].

The goal of the study was the identification of careful and consistent assessors in the early stages of the experiment, allowing us to minimize assessment costs and improve assessment quality. Unfortunately, while consistency can be high for some assessors, and appears reasonable for most assessors, we were unable to find a method of reliably detecting assessors. Moreover, assessors themselves do not remain consistent from context to context. However, despite this lack of consistency on the part of individual assessors, the group as a whole were able to identify significant differences between systems. Moreover, other research [12] into selecting the number of assessors to employ, supports the numbers of assessors selected for the TREC tasks.

3.2 Issue #2: Limitations of Evaluation Measures

The TREC Contextual Suggestions Tracks use precision@5 and mean reciprocal rank (MRR) as their primary evaluation measures. Unfortunately, precision@5 implicitly assumes that a user will always look at exactly the first five results, no more and no less, while MRR implicitly assumes that the user stops at the first useful result. Can we create an evaluation measure that better matches user behaviour?

A user's reaction to a suggestions could be negative ("dislike"), as well as positive ("like") or neutral, and too many disliked suggestions may cause the user to abandon the results. On the other hand, by reviewing caption descriptions, the user may be able to quickly skip suggestions that are not of interest, reaching much deeper into the list than the first five. Building on the *time-biased gain* (TBG) framework of Smucker and Clarke [32], which recognizes time as a critical

element in user modeling for evaluation, we developed an evaluation measure that directly accommodates these factors [13].

The tutorial presented this version of TBG, which is tailored to the Contextual Suggestion task, along with some motivation and results. This version of TBG accounts for the impact of descriptions and disliked suggestions, both of which are ignored by the official track measures. The measure models a user working their way through a ranked list of suggestions, pausing to investigate the webpages associated with descriptions they like. Gain—or benefit to the user—is recognized after the user views a page they like. Disliked suggestions may cause the user to stop browsing. The model has four parameters, reflecting the probabilities of taking certain actions and the time needed to take these actions. These parameters may be set through studies of actual user behaviour, as captured in query logs and other sources.

3.3 Issue #3: Reusability and Repeatability

One goal of the TREC Contextual Suggestion Track is the creation of reusable test collections for future experiments. Output from the TREC Tracks during 2012–2014 included judgments from hundreds of assessors for hundreds of suggestions across hundreds of cities. Can these suggestions and judgments be re-used to evaluate future system?

We are still working on this issue [17]. Unfortunately, the reusability of collections developed for TREC 2012–2014 has proven to be limited. One problem in these years is that each participating group developed their own sets of candidate attractions for each venue, as well as their own descriptions for these attractions. For TREC 2015 (see below) suggestions must be made from a closed set of attractions, which may improve reusability.

3.4 TREC 2015 and Beyond

The track continued for TREC 2015, but with a very different character. This year, we took a "living labs" approach. Participants provided a continuously running online engine, and our server connected crowdsourced users with suggestions provided by these engines. In addition, suggestions were limited to a pre-defined set, with the goal of improving reusability.

If the track continues into the future, we hope to transition to a continuously running online evaluation service, managing a federation of online recommendation engines. Ideally, the service could be used for evaluation experiments outside the bounds of TREC, perhaps with real travelers slowly replacing crowdsourced workers. We are looking for volunteers to help make these ideas work!

In the next section, we discuss detailed approaches of to using contextual information in modeling search and interaction behavior.

4 Using Contextual Information to Understand Searching and Browsing Behavior

In the fourth session,[5] Julia detailed approaches to use contextual information to model search and browsing behavior. Modern search still relies on the query-response paradigm, which is characterized by a sharp contrast between the richness of data in the index, and the relative poverty of information in the query, usually expressed in a few keywords to capture a complex need. This is particularly true in online search services, where the same query may be observed from many users, with considerable variations in their search intents. Contextual information is the obvious route to try to restore the balance, and behavioral data related to user's searching and browsing activities provides new opportunities to model contextual aspects of user needs.

The importance of contextual information in search applications has been recognised by researchers and practitioners in many disciplines, including recommendation systems, information retrieval, ubiquitous and mobile computing, and marketing. Context-aware systems [20,21] adapt to users operations and thus aim at improving the usability and effectiveness by taking context into account. In this work we consider two types of behavior: (1) 'searching'—when users are issuing queries and we are trying to improve search results (SERP) taking context of sessions into account; and (2) 'browsing'—when users are surfing a website and we are predicting their movements utilizing context.

The main research problem we are investigating is the value of context in searching and browsing user behaviour on web: *how to discover, model and utilize contextual information in order to understand and improve users' searching and browsing behaviour on web*? We start by giving an overview of context as used in the literature (in Sect. 4.1). We continue by developing a general analytic framework that views context aware search from the system's perspective (in Sect. 4.2). This analytic part defines a general framework for modeling context, and introduces the notions of optimal contextual models and useful contextual models. Next, we look at the impact of specific contextual aspects, starting with geographic location as static contextual aspect (in Sect. 4.3), and similar behavioral trails of search and browse actions as dynamic contextual aspects (in Sect. 4.4). Finally, we look at behavioral dynamics—changes in aggregated user behavioral features over time—such as the frequency of query revisions and SAT/DSAT clicks to detect changes in user satisfaction and drifts in query intent (in Sect. 4.5).

4.1 What Is (Not) Context?

In this section, we give a short overview of "context" in the literature. First, we give a broad overview of context as used in various field. Second, we detail

[5] The third session introduced the hackathon and the tools and data available for it, and will be discussed together with the outcome of the hackathon in the next section.

Table 1. The evolution of context definition

Context	Year
Location	1992
Taxonomy of explicit context	1999
Predictive features versus contextual	2002
Hidden context: clustering, mixture models	2004
Contextual bandits	2007
History of previous interaction	2008
Independence of predicted class	2011
Two level prediction model	2012
Focus on context discovery	2012–

the use of context in search systems. Third, we discuss the use of context in recommender systems.

Many interpretations of the notion of context have emerged in various fields of research like psychology, philosophy, and computer science [6]. In literature, a context was presented as additional (situational) information: a user's location [1], helping to identify people near the user and objects around [19], current date, season, and weather [7]. Later, the user's emotional status was added to the context-aware application, Dey et al. [15] broadened the definition to "any information that can characterise and is relevant to the interaction between a user and an application." These works typically assume that context is explicit and given by a domain expert, whereas our focus is on implicit contextual information.

In machine learning, context was considered as *contextual features* in supervised concept learning [35]. The contextual features are useful for classification only when they are considered in combination with other features. For example, in medical diagnosis problems, the patent's gender, age, and weight are often available. These features are contextual, since they (typically) do not influence the diagnosis when they are considered in isolation. Later it was discovered that a context may not necessarily be present in form of a single variable in the feature space. It can be hidden in the data. Turney [34] formulated the problem of recovering implicit context information and proposed two techniques: input data clustering and time sequence. According to Prahalad [29], a context has temporal (when to deliver), spatial (where), and technological (how) dimensions. In terms of interactive systems, Palmisano et al. [28] has shown that it was useful to consider the history of user interaction (changes in these entities). In Zliobaite [39] a context was defined as an artifact in the data that does not directly predict the class label, e.g. accent in speech recognition. Zliobaite et al. [40] proposed context-aware systems as two level prediction models for food sales. The timeline of the main milestones related to the research of context in predictive modeling is presented in Table 1.

In information retrieval, context of a search query often provides a search engine with meaningful hints for answering the current query better and can be utilised for ranking. Given a query, a search engine returns the matched documents in a ranked list to meet the user's information need. Understanding users' search intent expressed through their search queries is crucial to Web search. A web query classification has been widely studied for this purpose. Cao et al. [8] incorporates context information into the problem of query classification by using conditional random fields models (context is used to expand a feature space). This approach uses neighbouring queries and their corresponding clicked Web pages in search sessions as a context. Context-aware search adapts search results to individual search needs using contexts. While personalised search considers individual users long and/or short histories, context-aware search focuses on short histories of all users. Xiang et al. [37] adopts a learning-to-rank approach and integrates the ranking principles into a state-of-the-art ranking model by encoding the context as a feature of the model. The experimental results clearly show that this context-aware ranking approach improves the ranking of a commercial search engine.

In recommender systems, Adomavicius and Tuzhilin [2] showed that the *situation* in which a choice is made is important information. E.g., using a temporal context in a travel recommender system would provide a vacation recommendation in the winter that can be very different from the one in the summer. Similarly, in the case of personalised content delivery on a Web site, it is important to determine what content needs to be recommended to a customer. The purchase intent of a customer is considered as contextual information in an e-commerce application because different purchasing intents may lead to different types of behavior [2]. The purchase intent usually is considered as a hidden context which has to be derived. Then it can be used to select 'right' model. The context-aware recommenders utilize the information about a situation to make predictions. Palmisano et al. [28] defined a hierarchy of a context in the recommendation system they used the obtained contextual features to expand feature space. The other effective method for a context-aware rating prediction is Multiverse Recommendation based on the Tucker tensor factorization model [33]. Stern et al. [33] presented probabilistic model for generating personalised recommendations of items to users of a web service. Their system makes use of explicit context information in the form of a user (e.g. age and gender) and meta data of an item (e.g. author and manufacturer) in combination with collaborative filtering information from previous user behaviour in order to predict the value of an item for a user. The contextual information is integrated into the prediction process using a feature set expansion manner to produce the better recommendations. Rendle et al. [30] proposed a novel approach applying Factorization Machines to model contextual information and to provide context-aware rating predictions using context explicitly specified by a user to the set of predictive features.

4.2 General Definition of Useful Context

We will now discuss how to define a general analytical framework for context-aware systems. By defining a general framework, we can clarify concepts, and define the abstract problem underlying the use of context in concrete applications.

First, we define a general view of what is contextual information. Then we introduce how contextual information might be utilized. Let $\Theta = C_1 \times \cdots \times C_i \times \cdots \times C_N$ be the space of all possible contextual features associated with every data instance, where each C_i is a context. Denote $\theta_s \in \Theta$ as the contextual feature vector associated with each data instance s. Let $M : \Theta \times D \mapsto V$ be a predictive contextual model that maps each test instance $s \in D$ associated with the contextual information θ_s to a decision space V. Let $F(s, M(\theta_s, s)) : D \times V \mapsto \mathbb{R}$ be a function evaluates how good a model is. For example, in the case of the next action prediction, it foretells a next users' activity. The space of users' activities is the following set: $\{Search = a, Click\,on\,Ad\,Banner = b, Click\,on\,Recommendation = c\}$. In this case, our decision space V is the same as our data instance space D. An example of the evaluation function might be the number true predictions made by M over the test instance s. For instance, assume that the model M predicts the following set of activities $s = ababc$ as $M(\theta_s, s) = \underline{ab}ed\underline{c}$ then it makes three true predictions corresponding to the underlined activities, i.e. $F(s, M(\theta_s, s)) = 3$.

Let $T \subseteq D$ be a set of test instances and denote $Pr(s)$ as the probability that $s \in T$. The expectation of an evaluation function $F(s, M(\theta_s, s))$ over our test set is defined as $E[T, M] = \sum_{s \in T} Pr(s) * F(s, M(\theta_s, s))$. The value of the expectation $E[T, M]$ can be considered as an objective function that needs to be maximized. We assume that $\exists M^*$ which is a (sub-)optimal model, i.e. $M^* = \arg\max_M E[T, M]$. Essentially, the optimal model uncovers the optimal weights of each contextual feature (either static as location, or dynamic as search trail characteristics) in order to predict the outcome (such as the next action, or a result click, or a query revision).

Let C be a context with n categories: $C = \{c_1, \ldots, c_j, \ldots, c_n\}$ associated with each data instance $s \in D$. A context may have different categories, e.g. the geographical context can be divided into four categories such as continents: Europe, Africa, American, or Asia. For simplifying our discussion, we consider the context that have only two categories, as the discussion for the general case which includes than two categories is very similar. Assume that we have a context C with two categories c_1 and c_2 dividing the test set into two disjoint subsets T_1 and T_2 such that $T = T_1 \cup T_2$. Denote M_1 and M_2 as two predictive models built for the category c_1 and c_2 respectively. Let $P(c_1)$ and $P(c_2)$ are probabilities that a test instance belonging to the category c_1 and c_2 respectively.

Theorem 1. (Contextual Principle). *Let M^* be an (sub-)optimal model for T then it is a combination of M_1^* and M_2^*. Where M_1^* is an optimal model for T_1 and M_2^* is an optimal model for T_2.*

Theorem 1 (the formal proof is provided in [26]) shows that the problem of finding the best model for every test instance can be solved by considering the sub-problems of finding optimal models for test instances in each individual contextual category. This is a technical result of a desirable property that allows us to work on customization to user types or profiles, or personas, rather than personalization to specific individuals.

Nevertheless, in practice finding an optimal model for each contextual category is usually as hard as finding an optimal model for the whole data. Indeed, it is usually the case that the type of model is chosen in advance, e.g. Markov models. Model's parameters are estimated from training data D. Under this circumstance, contextual predictive analytics seeks for a context such that it divides the training data into two subsets D_1 and D_2 and the predictive models trained on D_1 and D_2 improve the predictive performance in comparison to the model trained on the whole training data. To this end, we define useful contexts as follows:

Definition 1. (Useful Context). *Given a model M built based upon the whole training data D and M_1, M_2 are two models built based upon D_1 and D_2 corresponding to each contextual category of a context C respectively. The context C is useful if an only if: $E[T_1, M_1] \geq E[T_1, M]$ and $E[T_2, M_2] \geq E[T_2, M]$*

Essentially, this definition captures the usual operational situation in which no global optimum is sought, but there is a current system (captured by model M) that we seek to improve by taking into account context C.

4.3 Location as Context

Next, we will discuss what is the impact of geographical location as a contextual information. The geographical location of users is one of the prototypical aspects as a contextual information. In the literature, it was shown that the *user's location* is useful contextual information in many applications [5,31,36]. A context based on geographical location can have different levels of granularity like continent, country, city and so on.

In our experiments with StudyPortals [26], we consider a task of users' next action prediction. In order to accomplish this task we build contextual Markov models. We concentrate on a continent level of geographical location due to limitations from the data size side. We use users IP addresses as contextual features, then $\theta_s = IP$ is contextual vector associated with each user session s. We define six contextual categories: $C_{geo} = \{C_1 = Europe, C_2 = Africa, C_3 = North\ America, C_4 = South\ America, C_5 = Asia, C_6 = Oceania\}$. We have shown in [26] that for the case of StudyPortals the geographical location is no useful context.

Geographical location on a city level is considered as a context in TREC Contextual Suggestion Track [9,10,14]. The main goal of this task is to learn user's preferences out of provided examples of users' profiles where users rate different attractions. Afterwards, we need to return a ranked list of up to fifty ranked

suggestions for each pair of user profile and context. The list of suggestions is ranked based on the user's preferences in the particular geographical location. As a source for contextual suggestions we used data from four social networks namely Facebook, Foursquare, Yelp, and Google Places, which are combined into one dataset. In order to achieve this goal, Kiseleva et al. [23] formulated the problem setup as a learning to rank problem where we directly optimize the required evaluation metrics, e.g., precision at rank 5 ($P@5$). We showed that our approach can be used in a preselection phase of contexts in the contextual suggestion task, but also that location is not equally useful for all web applications.

4.4 User Behavioral Aspects as Context

In addition to the relatively static location context, we will now look at dynamic context and discuss how to discover users behavioral aspects as contextual information. In case of StudyPortals[6] [25–27], users historical behavior is given as a log of web sessions corresponding to historical browsing activities of a user. In our case the users' actions are categorized by the type of the users' actions: searches, clicks on ads or homepage visits. Users' activities and their possible orderings within user web sessions is summarized as a user navigation graph. We want to understand if there are any groups of nodes in the navigation graph and then use this knowledge to characterize the users' behaviour in order to improve effectiveness of next users' action prediction. In order to achieve our goal we propose to use several machine learning techniques: First, we discover two types of user's behaviour on a site by grouping the user navigation graph: (1) expert users, who is experienced with website interface or searches extensively to find required information, and (2) novice user, who needs more time to learn about a website or is not interested much in content. Second, we discover changes in user intents while browsing a website. In order to achieve it we apply hierarchical clustering techniques with different optimisation functions: (1) directly maximizing the accuracy of next action prediction [25], and (2) directly minimizing the compression length [27] of decomposed web sessions. We described how the discovered contexts can utilized for the benefits of particular applications and use cases.

4.5 Changes in User Behavior over Time

In the final part we will look at changes in context over time, and discuss how to define and to detect changes in user satisfaction with retrieved search results. We look at indicators of a drop in user satisfaction due to SERPs trained on historical data becoming dis-aligned with a drift in query intent over time [22,24].

When users struggle to find an answer for query Q they run a follow-up query Q' that is an expansion of Q. Query reformulation is the act of submitting a next query Q' to modify a previous $SERP$ for a query Q in the hope of retrieving better results [18]. Such a query reformulation is a strong indication

[6] See: http://www.studyportals.eu/.

of user dissatisfaction [3]. We call this the *reformulation signal*. Our hypothesis is that a decrease in user satisfaction with $\langle Q, SERP \rangle$ correlates nicely with the reformulation signal. In other words, the probability of reformulating Q will grow dramatically.

We propose an unsupervised approach, called *Drift Detection in user SATisfaction (DDSAT)*, for detecting drifts in user satisfaction for pairs $\langle Q, SERP \rangle$ by applying a concept drift technique [38] leveraging reformulation signal. Concept drift primarily refers to an online supervised learning scenario when the relation between the input data and the target variable changes over time [16]. Furthermore, the reformulation signal is considered to be less noisy and if reformulations are fresh and done only by users' initiative then we can say that a reformulation signal is not biased by information coming from the search engine.

We conduct a large-scale evaluation using search log data from Microsoft Bing[7] [22] and Yandex[8] [24] where we extend our framework by taking into account more signs of user frustration (lack of search satisfaction) such as: a rate of search abandonment, a dramatical change in query volume, a lowering in average click positions. Our experiments show that the algorithm *DDSAT* works with a high accuracy. Moreover, our framework outputs the list of drift terms and the list of $URLs$, which can be used for the future re-ranking of $SERP$. The algorithm of the drift detection in user satisfaction can be incorporated in many search-related applications where freshness is required, e.g. in recency ranking, query auto-completion.

In addition, we conducted conceptual analysis to clarify the meaning of core concepts and their relations and dependencies. And as a conceptual model, we worked with an idealized model that abstracts away from other factors outside the scope of our interest.

In the next section, we discuss the setup and results of the hackathon.

5 The Hackathon

The hackathon consisted of a miniature version of the TREC Contextual Suggestion Track. The stated goal of the hackathon was to provide recommendations to the organizers of this tutorial regarding sites to see and things to do in Saint Petersburg during their visit. The hackathon was initiated the evening of Tuesday, August 25, with pizza and beer provided by the organizers as inspiration. Teams reported out two days later, with presentations on the evening of Thursday, August 27. The hackathon involved ten teams, with a total of 30 participants.

No time was allocated for working on the task during the days in between—only evenings were available, limiting the amount of work that could be done. Nonetheless, as described below, a number of teams put considerable effort into the task, coming up with many highly creative and interesting solutions. After the report-out, the organizers awarded prizes for Best System, Best Presentation,

[7] See: http://www.bing.com/.
[8] See: http://yandex.ru/.

Table 2. Context mapping from IDs to Cities and States

Id	City	State
151	New York City	NY
152	Chicago	IL
⋮		
421	Walla Walla	WA
422	Lewiston	ID
423	Saint Petersburg	Russia

and Most Original Approach. Slides from some of the student presentations are available online.[9]

5.1 Data Resources Available

The hackathon used a variant of the TREC 2015 Contextual Suggestion Track's Batch Task,[10] tailored to the lecturers visiting St. Petersburg. The data is available from http://plg.uwaterloo.ca/~claclark/russir2015/.

Data. First, there is the core data material: the requests (input) and sample responses (the results your system should generate). In short: you get a new context (city) with candidate venues to rank, plus detail about the person asking, including what she/he likes in other cities. The contexts are a simple csv file with `id`, `city`, and `state` fields, as shown in Table 2. The contexts are based on the TREC contextual suggestion track (US cities) extended with St. Petersburg. The lecturers apologized for the format which makes the inappropriate suggestion that Russia is a US state. The requests consists of the profile and context, as well as the candidates to rank as shown in Table 3. It details the request (901), the context (location), and details about the person requesting (person) and the trip, as well as the candidates to rank. The profile contains a large set of preferences n another context or city. The responses consists of the group and run details, as well as a ranked list of suggestions (derived from the candidates) as shown in Table 4.

Evaluation. Second, there is an evaluation package for the US cities, to evaluate or train systems. This consists of the judgments by the person for each of the candidates (ratings and tags/endorsements), a script to transform the response file into the TREC format, and the TREC script to calculate standard IR measures.

[9] http://plg.uwaterloo.ca/~claclark/russir2015/Students.
[10] See: https://sites.google.com/site/treccontext/.

Table 3. Request: Profile and candidates to rank

JSON request


```
{ "body" : {
      "group" : "Friends",
      "duration" : "Longer",
      "season" : "Autumn"
      "trip_type" : "Holiday",
      "person" : {
         "preferences" : [
            {"documentId" : "TRECCS-00247656-160",
               "tags" : [
                  "Bar-hopping",
                  "Clubbing"
               ],
               "rating" : "4"
            },
            {"documentId" : "TRECCS-00211603-161",
               "tags" : [
                  "Fast Food",
                  "Restaurants"
               ],
               "rating" : "0"
            },
         ...
         ],
         "id" : 1234568,
         "age" : "47",
         "gender" : "male"},
      "location" : {
         "id" : 423,
         "lat" : 59.95, "lng" : 30.3,
         "name" : "Saint Petersburg"},
   },
   "id" : 901,
   "candidates" : [
      "TRECCS-00000001-423",
  ...
      "TRECCS-00000102-423"]}
```


Additional Data. Third, there is additional data that can be used. There is the crawled page of the venues (all URLs of venues in the Batch task, as well as those in St. Petersburg). There is also the data used at TREC about a much larger set of pages, as detailed on the TREC pages. And there are categories and ratings of the US venues, obtained from a commercial service.

Table 4. Response: group and run details plus suggestions.

JSON response

```
{
   "groupid" : "demo",
   "runid" : "demoA",
   "id" : 901,
   "body" : {
      "suggestions" : [
            "TRECCS-00000099-423",
            "TRECCS-00000006-423",
            ...
            "TRECCS-00000079-423"      ]
   }
}
```

5.2 Student Presentations

To provide a sense of the breadth and variety of approach, we provide a short overview of the efforts from several groups, who were kind enough to provide slides and other material after the hackathon.

- Team **MAD IT** (Maria Zagulova, Andrey Poletaev, Dmitry Zhelonkin, Ivan Grechikhin, and Tania Nikulina) clustered people on the basis of demographics (age, gender, etc.) and personalized each cluster using tag activity. For one organizer, suggestions included Le Tour de Vin wine bar, Saint Petersburg 300 Year Park, and well as various cafes. For the other two organizers, suggestions included the Grand Market Russia, the Faberge Museum, and the Mikhailovsky Theatre, as well as a tattoo parlor. These suggestions were well received by the organizers, with the exception perhaps of the tattoo parlor. The team has made source code, as well as more information about their work available for interested readers[11].
- Team **No Name** (Michael Nokel) took a colaborative filtering approach, by throwing out all data other than the attraction ratings and applying singular value decomposition (SVD). Gradient boosting regression was then applied to a combination of user features and SVD features, showing improvements over SVD along on the training data. Unfortunately, no recommendations for the organizers were made.
- Team **Rambler & Co** (Maria Zagulova, Andrey Poletaev, Dmitry Zhelonkin, Ivan Grechikhin, and Tania Nikulina) used vowpalwabbit rank approximations to predict user ratings. Features included gender, age, season, etc., as well as LDA topics of Tripadvisor and Foursquare titles trained on translated titles. For candidate attractions in Saint Petersburg, the team manually assigned tags. Suggested attractions incuded the El Copitas Cocktail Bar, the Wine Bar Bratya Tonet, and the Co-op Garage Bar.

[11] bitbucket.org/poletaev/russir-2015/src.

– Team **sleep_deprived** (Sagun Pai and Sheikh Muhammad Sarwar) applied collaborative filtering methods, approaching the cold start problem through an tag expansion approach, for example, automatically expanding the tag "food" to include "seafood". An expanded vector of tags was created for each user, and these expanded vectors were used to select recommendations within Saint Petersburg. Suggestions included some of the same attractions recommended by other groups, (e.g., the Mikhailovsky Theater), as well as various bars (8th Line Pub), restaurants (Wave Burgers), and palaces (Catherine Palace).

5.3 The Outcome

After student presentations were complete, prizes were awarded as follows:

– the *Best System Award* went to MAD IT (Maria Zagulova, Andrey Poletaev, Dmitry Zhelonkin, Ivan Grechikhin, and Tania Nikulina);
– the *Best Presentation Award* went to SalsaRoulette (Navid Rekabsaz, Larisa Adamyan, Ioanna Miliou, and Aldo Lipani); and
– the *Most Original Approach Award* went to sleep_deprived (Sagun Pai, Sheikh Muhammad Sarwar).

Congratulations! And thank you to all groups for making the experience so enjoyable. The organizers visited several of the recommended places over the weekend, and can confirm that the suggestions were indeed highly relevant.

In the next section, we concludes the paper with some discussion on the outcome of the lectures and hackathon.

6 Conclusion

This paper described the course on contextual search and exploration, given as part of the ninth Russian Summer School in Information Retrieval (RuSSIR 2015).[12] Our goal was to enable students to understand the challenges and opportunities of contextualized search over entities, and learn effective approaches for the concrete application to venue recommendation domain, as well as obtain hands-on experience with developing and evaluating personalized search and recommendation approaches.

The course consisted of both lectures and hands-on "hackathon" sessions with data derived from the TREC task. First, we gave an overview of approaches to contextual search and exploration, both in terms of the general problem of complex task support, and specifically focusing on a venue recommendation task. Second, we detailed how to set up an experiment to evaluate contextual suggestion based on the TREC track. Third, we detailed a general approach to using contextual information in modeling search and interaction behavior. Fourth, we discussed the setup and results of the hackathon, in which the students were asked to make recommendations to the lecturers on what to do in St. Petersburg after the course.

[12] See: http://romip.ru/russir2015/.

Acknowledgments. We are grateful to RuSSIR to cover the travel expenses of the first author. We are thankful to the 30 students that actively participated in the hackathon—we were deeply impressed by the amount of work and creative ideas that were tried within 48 hours!

References

1. Abowd, G.D., Dey, A.K.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: CARS (2010)
3. Ageev, M., Guo, Q., Lagun, D., Agichtein, E.: Find it if you can: a game for modeling different types of web search success using interaction data. In: SIGIR (2011)
4. Allan, J., Croft, B., Moffat, A., Sanderson, M.: Frontiers, challenges, and opportunities for information retrieval: report from SWIRL 2012 the second strategic workshop on information retrieval in Lorne. SIGIR Forum **46**(1), 2–32 (2012)
5. Alves, A.O., Pereira, F.C.: Making sense of location context. In: Proceedings of the 1st International Workshop on Context Discovery and Data Mining (ContextDD 2012), vol. 4, p. 7. ACM, New York (2012). http://dx.doi.org/10.1145/2346604.2346609
6. Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. In: SIGMOD (2007)
7. Brown, P., Bovey, J., Chen, X.: Context-aware applications: from the laboratory to the marketplace. IEEE Pers. Commun. **4**, 58–64 (1997)
8. Cao, H., Hu, D.H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., Yang, Q.: Context-aware query classification. In: SIGIR (2009)
9. Dean-Hall, A., Clarke, C.L., Kamps, J., Thomas, P., Voorhees, E.: Overview of the TREC 2014 contextual suggestion track. In: 23rd Text REtrieval Conference, Gaithersburg, Maryland (2015)
10. Dean-Hall, A., Clarke, C.L., Simone, N., Kamps, J., Thomas, P., Voorhees, E.: Overview of the TREC 2013 contextual suggestion track. In: 22nd Text REtrieval Conference, Gaithersburg, Maryland (2014)
11. Dean-Hall, A., Clarke, C.L.A.: Assessing contextual suggestion. In: 6th International Workshop on Evaluating Information Access, Tokyo, December 2014
12. Dean-Hall, A., Clarke, C.L.A.: The power of contextual suggestion. In: 37th European Conference on Information Retrieval, pp. 352–357, Vienna, March 2015
13. Dean-Hall, A., Clarke, C.L.A., Kamps, J., Thomas, P.: Evaluating contextual suggestion. In: 5th International Workshop on Evaluating Information Access, Tokyo, June 2013
14. Dean-Hall, A., Clarke, C.L.A., Kamps, J., Thomas, P., Voorhees, E.: Overview of the TREC 2012 contextual suggestion track. In: 21st Text REtrieval Conference, Gaithersburg, Maryland (2013)
15. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of contextaware applications. Hum. Comput. Interact. **2**, 97–166 (2001)
16. Gama, J., Žliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**, 4:1–4:37, Article 44 (2014). http://dx.doi.org/10.1145/2523813

17. Hashemi, S.H., Clarke, C.L., Dean-Hall, A., Kamps, J., Kiseleva, J.: On the reusability of open test collections. In: 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, pp. 827–830, August 2015
18. Hassan, A., Shi, X., Craswell, N., Ramsey, B.: Beyond clicks: query reformulation as a predictor of search satisfaction. In: CIKM, pp. 2019–2028 (2013)
19. Hull, R., Neaves, P., Bedford-Roberts, J.: Toward situated computing. In: ISWC, pp. 146–153 (1997)
20. Kiseleva, J.: Context mining and integration into predictive web analytics. In: WWW (Companion Volume), pp. 383–388 (2013)
21. Kiseleva, J.: Using contextual information to understand searching and browsing behavior. In: Submission of SIGIR (Doctoral Consorcium) (2015)
22. Kiseleva, J., Crestan, E., Brigo, R., Dittel, R.: Modelling and detecting changes in user satisfaction. In: Proceeding of CIKM, pp. 1449–1458 (2014)
23. Kiseleva, J., García, A.M., Luo, Y., Kamps, J., Pechenizkiy, M., Bra, P.D.: Applying learning to rank techniques to contextual suggestions. In: Proceeding of Text REtrieval Conference (TREC) (2014)
24. Kiseleva, J., Kamps, J., Nikulin, V., Makarov, N.: Behavioral dynamics from the SERP's perspective: what are failed SERPS and how to fix them? In: CIKM, pp. 1561–1570 (2015)
25. Kiseleva, J., Lam, H.T., Pechenizkiy, M., Calders, T.: Discovering temporal hidden contexts in web sessions for user trail prediction. In: Proceedings of WWW (Companion Volume), pp. 1067–1074. ACM (2013)
26. Kiseleva, J., Lam, H.T., Pechenizkiy, M., Calders, T.: Predicting current user intent with contextual markov models. In: ICDM Workshops (2013)
27. Lam, H.T., Kiseleva, J., Pechenizkiy, M., Calders, T.: Decomposing a sequence into independent subsequences using compression algorithms. In: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytic, pp. 67–75 (2014)
28. Palmisano, C., Tuzhilin, A., Gorgoglione, M.: Using context to improve predictive modeling of customers in personalization applications. IEEE Trans. Knowl. Data Eng. (TKDE) **20**(11), 1535–1549 (2008)
29. Prahalad, C.: Beyond CRM: predicts customer context is the next big thing. In: AMA MwWorld (2004)
30. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, pp. 635–644 (2011)
31. Schmidt, A., Beigl, M., Gellersen, H.-W.: There is more to context than location. Comput. Graph. **23**(6), 893–901 (1999)
32. Smucker, M.D., Clarke, C.L.: Time-based calibration of effectiveness measures. In: 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, Oregon, pp. 95–104 (2012)
33. Stern, D., Herbrich, R., Graepel, T.: Matchbox: large scale online bayesian recommendations. In: WWW, pp. 111–120 (2009)
34. Turney, P.: Exploiting context when learning to classify. CoRR (2002)
35. Turney, P.: The management of context-sensitive features: a review of strategies. CoRR (2002)
36. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Trans. Inf. Syst. (TOIS) **10**(1), 91–202 (1992)

37. Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., Li, H.: Context-aware ranking in web search. In: SIGIR (2010)
38. Zliobaite, I.: Learning under concept drift: an overview. CoRR abs/1010.4784 (2010)
39. Žliobaitė, I.: Identifying hidden contexts in classification. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 277–288. Springer, Heidelberg (2011)
40. Zliobaite, I., Bakker, J., Pechenizkiy, M.: Beating the baseline prediction in food sales: how intelligent an intelligent predictor is? Expert Syst. Appl. (ESWA) **39**(1), 806–815 (2012)