A Contextual Approach to Dynamic Search and Topic Change

A Thesis
submitted to the Faculty of
Georgetown College
of Georgetown University
in partial fulfillment of the requirements for the
degree of
B.S.
in Computer Science

By

Angela Yang

Washington, DC
May 6, 2017

A Contextual Approach to Dynamic Search and Topic Change

Angela Yang

Thesis Advisor: Grace Hui Yang

Abstract

In this thesis, we propose a unique approach to dynamic search that uses a LinUCB contextual bandit algorithm to automatically search topics and change queries when finished. The algorithm models actions that a user might utilize when querying a search engine, including the action of ending one topic search and starting another by completely changing the query, and uses the context surrounding the current state of the search to choose one action each iteration with which to interact with the search engine. By using a contextual approach to reinforcement learning to select an action at each iteration, this process fully automates the search from beginning to end, choosing both how to search the topic and how long to search for, leading to a more effective retrieval with varied numbers of queries for each topic. This experiment is conducted using the data from TREC 2016 Dynamic Domain Track. Our results show that using the context features of the current search iteration to select the next action in the search allows the system to retrieve enough relevant documents for a complex search task and also that allowing the algorithm to decide how long to search each topic does not negatively affect the search process's ability to satisfy a user's information need.

Index words: Information retrieval, Dynamic search, Machine learning, Contextual bandit, Multi-armed bandit

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

In our everyday lives, we use two types of search tasks to find different types of information: single query searches and complex topic searches. A single query search is usually a search for a question or query with a clear information goal. This can be searches for facts or specific information that was not previously known, such as “Georgetown University address” or “how long do birds live” This makes it easy to categorize documents as relevant or non-relevant. Complex topic searches, on the other hand, are more complicated in that they consist of a broad information goal. These searches can be something like “California droughts” or “Japanese food.” The information need in these searches is more broad and typically have more than one informational aspect that must be satisfied. This goal can usually only be reached through multiple iterations, with iterations to mean queries, through a variety of topics within that topic domain. Furthermore, a document that is considered relevant for one sub-topic within that information domain may not be considered relevant for another sub-topic. Search engines are currently optimized to work with single query searches, but the format in which they handle searches may not be adequate for more complex searches like broad topic domain searches. By focusing on how to deal with search over multiple iterations, taking into account relevancy feedback at each

Table 1.1: TREC 2016 dynamic search on the Polar domain [32]

	Topic		Subtopic
1	Ice sheet level rise	1.1	Greenland
		1.2	NSIDC
		1.3	Sea-level rise
2	Polar oceans freshwater sensitivity	2.1	Surface freshwater forcing
		2.2	Arctic Freshwater Initiative
3	Sea-level rise and coastal erosion	3.1	Shoreline erosion
		3.2	NOAA
		3.3	Sea-level rise
		3.4	Coastal hazards

iteration and the length of time that the user has been searching, we can better satisfy the user's needs for these complex searches.

In a complex dynamic search, there is a shift in the representation of the information space. The user, when looking at a document, provides feedback on the document's relevance towards whichever topic within the entire topic domain they are currently looking at. The feedback they give is dependent on the current topic that they are searching. We aim to give the search system the responsibility to use this information to formulate a new query and retrieve more documents that will fit the user's need for that same topic, not the user. Rather than navigating through separate information spaces in each query, the search system is instead navigating within a section of the information space repeatedly to fulfill the user's complex information need.

Table 1.1 shows an example dynamic search on the Polar domain from TREC Dynamic Domain 2016 [32], or data related to the polar sciences. In this example, we see that the overall search's information need can be encapsulated in the entire Polar domain's information space. Each topic is necessary to the user's search and satisfies a portion of the user's need, from topic 1 of "Ice sheet level rise" to topic 3 of "Sea-level rise and coastal erosion." Within each topic too, there are multiple facets, or subtopics, that the user would like to explore and considers relevant. The goal of the dynamic search is to adequately search all of these topics and subtopics through multiple iterations, learning from the user's feedback what is and is not relevant. In the following sections, we will define the requirements of dynamic search more completely.

1.2 Dynamic Search

Dynamic search is defined as search that uses multiple iterations to fulfill the search, with the search system dynamically modeling the search process through relevance feedback that is given by the user. Dynamic domain is, then, a dynamic search over an entire information domain. The multiple iterations of search are used to traverse the different topics within that domain.

1.2.1 A Concrete Example

Consider a search user who is planning a vacation to New York City. This user's search can be considered a quest for information rather than a simple query and response. A single search on "New York City" will likely not yield enough results to satisfy her information need, as the search topic is broad and extensive, containing many topics and subtopics. Instead, the user will choose to split this broad topic domain into many topics within that cover the wide information need she has from her vacation plans plans. One topic might be "flights to New York City" while another might be

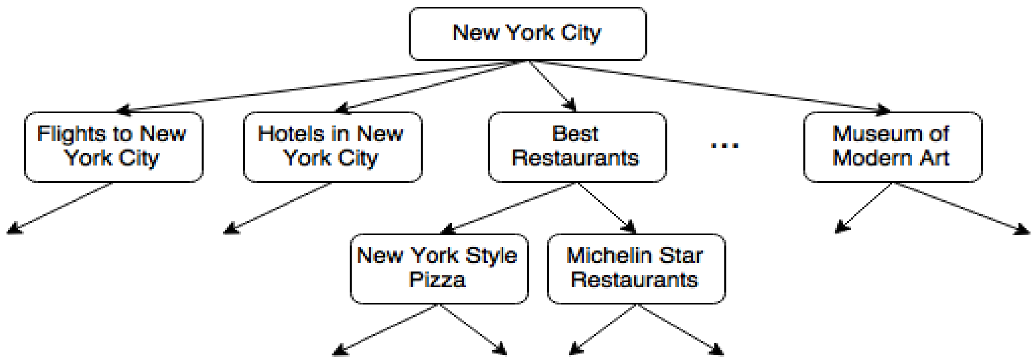

Figure 1.1: The hierarchy of topics and subtopics that make up the user's information need.

"New York Museum of Modern Art." These topics are all relevant to the user's topic domain and make up the complex information the user has.

In this way, a dynamic search problem is created. The user's complex information need requires a longer search process in order to retrieve enough relevant documents and thus satisfy her need. Her broad topic, "New York City," is split logically into multiple topics underneath the main topic domain, as seen in figure 1.1. These different topics, though related to the same domain, can be considered separate search queries with different relevant documents related to each one.

A single iteration of search, as an average search engine would expect, does not provide enough relevant results to cover all the topics within the broad topic domain the user intends to search for. Furthermore, any single query that can be formulated for all of the topics will be too vague to retrieve documents that are specific enough to be relevant. Instead, an iterative approach with each topic being searched separately would provide better results. The user can expect that by splitting the entire domain

Table 1.2: Start of an example search

User motivation	Query
User starts search with topic "hotels in New York City"	hotels in New York City
User doesn't find relevant results and decides to be more specific	budget hostels
User finds some relevant results and chooses to specify more	budget hostels in Brooklyn
User switches to topic "Best restaurants in New York City"	best restaurants in New York City
User finds many diverse results and makes the query more specific	New York style pizza
User moves to another subtopic on restaurants	Michelin star restaurants
User starts a new topic	Central Park

into topics and searching those topics over multiple iterations will lead to more relevant results. However, just as with a simple single-iteration search, the user would not want to spend too long on each topic and as such the ideal search system would provide relevant and widespread results so as to cover more ground in each iteration and shorten her search as much as possible.

The user's search may look like the example shown in table 1.2. The user will begin by looking at one topic within the topic domain, such as "hotels in New York City." The results that are returned might be related to the nice hotels in the New York City area, such as the Marriott or the Hilton, which may lead her to narrow her search to "budget hostels," specifying her more specific information need for a cheaper living arrangement. Then, she may continue to narrow her search with the query "budget hostels in Brooklyn." Once this topic's information need has been satisfied, the user will then decide to start another topic search within her search topic domain. This process is continued, then, until all topics have been searched to completion and the user's topic domain information need is completely satisfied.

1.2.2 The Future of Complex Search

In a complex or dynamic search task as the one described in the previous example, current search processes and search engines place most of the responsibility on the user. The burden is placed on the user's shoulders to navigate through the search and formulate queries to continue the search in a useful way. Modern search engines do not use any specialized method or system for dealing with complex searches, or searches with multiple topics and iterations.

The dynamic search research task focuses on progressing the complex search process to a more intuitive method that shifts the responsibility from the user to the system. The research is focused on improving the search system to take on the task of moving the search along and helping the user to navigate the search and even decide how long to search, rather than forcing the user to make these decisions themselves. In this way, the search task becomes a more interactive process between the search engine and the user and allows for a more satisfying search process, where the user is able to find the information they want both quicker and more completely.

1.2.3 TREC Dynamic Domain Track

In dynamic search, the focus of the task is not on each iteration separately, but on the overall search as it moves from iteration to iteration. While each topic within the information domain can be considered a search task within itself, the information retrieval task is only finished when all the topics have been exhausted and completed. Thus, this task of completing a complex search within a broad topic can be broken

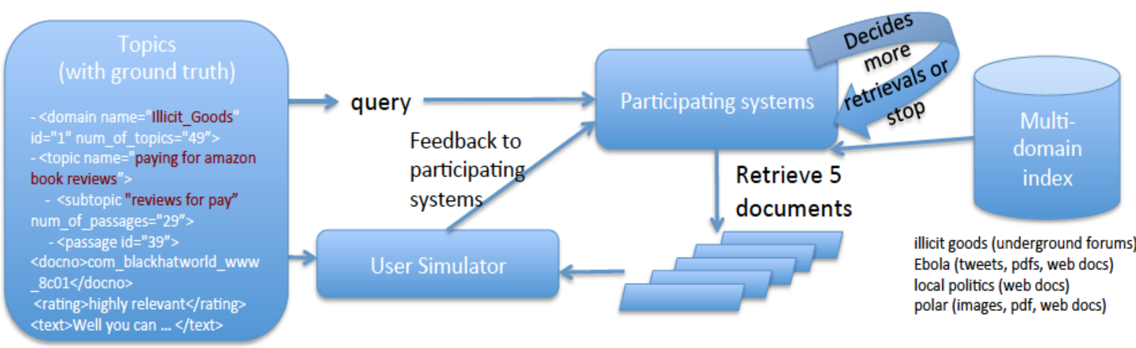

Figure 1.2: An illustration of the TREC dynamic domain search process.

down into two decisions: the decision of how to search the current topic at each iteration, and the decision of when to complete the search of a topic and move on to the next one. While trying to retrieve relevant information for the current topic, the system must select between multiple methods of continuing the search and modifying the query to find different relevant documents from the last iteration. However, when the user has decided that they have either exhausted all relevant documents or that they should give up on finding the rest, due to the sheer difficulty of finding relevant results, they can then decide to finish this search and move on to the next topic.

The TREC (Text REtrieval Conference) Dynamic Domain track [32] is modeled after this type of complex search, with a broad set of information and multiple iterations to find relevant results. In view of a search topic with a wide array of information that can be found or considered relevant to the user, the search is split into topics that are more specific and easier to focus on. Within each topic, the system can edit the query to continue to retrieve unique results, before the query is completely changed in order to move from one topic to the next.

As visually represented in figure 1.2, in the TREC Dynamic Domain track, a specific domain is provided with the goal of retrieving information about that specific topic dynamically, taking feedback from a simulated user, called the "jig" for every document retrieved. The systems are seeded with a beginning query, the topic that is to be searched, and the search system is tasked with retrieving five documents to send to the simulated user for feedback. Using the feedback, the system will then decide whether to continue the search by retrieving 5 more documents or ending the search on that topic. A successful run is considered one in which the entire topic space is explored and the user is provided with relevant documents both frequently and quickly. The aim of this track, then, is to effectively model dynamic search within a complex topic domain through algorithmic iterations of query and relevance judgments[1]. The feedback given by the simulated user is represented by the annotation of relevant passages within a document marked relevant, as well as a 1 to 5 relevance scoring of both the relevant passages and the relevant document as a whole. This feedback should be used by the user to re-formulate the query in some way and return even more relevant documents to the user.

The Dynamic Domain track was created to specifically deal with a complex search over a broad topic rather than a specific query, as well as the interaction between the user and the search system through the many iterations of a complex search [32].

1.3 Our Approach

This thesis focuses on the two subproblems posed within dynamic search: how to traverse the information space for one topic and when to complete the search on a topic. Most algorithms proposed to deal with dynamic search focus on the movements of the search process within a topic, the first subproblem, and merely manually specify

[1] http://infosense.cs.georgetown.edu/publication/mov2.mov

a stopping condition for each topic. In this thesis, we propose using a contextual bandit approach to traverse each topic as well as end one topic search and start another, thus automating the entire search under the algorithm's decision-making process.

1.3.1 Using a Contextual Bandit

Dynamic search consists of a decision at each iteration—whether to continue searching the same topic or end that search and start another, or in broader terms, how to search for documents this iteration—and a robust system should dynamically choose an action that will achieve a satisfactory result for the user both at the current iteration and for the search as a whole. By viewing these decisions as actions to be selected in each round, we apply a contextual bandit algorithm to the dynamic search task.

Contextual bandits are a subset of multi-armed bandits, which are a form of reinforcement learning algorithm that selects an action at each iteration or round, with the aim of a bandit algorithm being to choose an action that will maximize the reward received [15]. Contextual bandits, then, are multi-armed bandits with the addition of context [38]. On a conceptual level, the contextual bandit is used to dynamically decide the best approach for exhaustively searching the entire information space, both within topics and between topics, before the user is frustrated from too many iterations. The contextual bandit approach, then, amalgamates the two subproblems of dynamic search into one solution. Both the action of how to search a topic and the action of switching topics are given to the bandit to decide on when to run that particular action.

1.3.2 Traversing the Information Space and Changing the Topic

In dealing with how to complete the dynamic search task using a contextual bandit, we suggest some actions for the contextual bandit that will move the search around the entire topic domain's information space. The actions that the contextual bandit can choose from are the different directions that the search can be taken at any point in time. A novel approach that we take in this thesis is giving the decision of when to stop, to completely transplant the search from one area of the information space to another, to the algorithm. Rather than specifying a stopping condition manually or setting a static number of iterations that the system should run each time, we propose to leave the choice of when to complete the topic search to the contextual bandit.

Directions that the bandit can take the search correspond to editing the query within a topic. There are many ways to decide how to traverse the topic space and formulate new queries at each iteration. However, in their most basic form, there are only three ways to edit a query to search through a topic space. Terms can be added to the query, terms can be removed from the query, and the weight of the query's terms can be adjusted. These different possible actions, adding terms, removing terms, and changing the terms' weights, move the query, and by extension, the search in different directions within the topic.

These different actions that the contextual bandit can select from result in a search system that dynamically searches an entire information domain both simply and automatically from start to finish.

1.4 Contributions of this Thesis

This thesis presents the use of a contextual bandit algorithm, which in the past have been primarily used for recommender systems, to work as a dynamic search system. By using a contextual approach, the environmental context of the search can be utilized to determine the current state of the search process and point to the best action at any certain point in time. We define the context of the search through various factors of the search at that point in time, and provide this to the contextual bandit algorithm to make decisions using it. This approach also views the search as a constant attempt to maximize the relevancy returned and satisfaction of the user, the reward of each iteration.

In specifics of the algorithm itself, this thesis presents novel contributions in terms of the methods used to search each topic as well as its automation of the topic completion within the dynamic search. This automated approach simplifies the search process as seen by both the developer and the search user through the use of four actions which exhaustively explore the entire information space of the search topic domain.

1.5 Thesis Statement

In this thesis, we will use a reinforcement learning algorithm, a contextual bandit, to model the search system in a dynamic search process. As a contextual bandit uses the context given in each round to select an action, we utilize the current search context to determine how the search continues at each step of the search. We also automate the decision of the stopping condition for each topic search by providing it as an action for the bandit to choose from, rather than hard-coding it into the system.

Chapter 2

Related Works

2.1 Dynamic Search

Search that is done in sessions using multiple queries has had many names in Information Retrieval (IR) literature, including interactive search [22], session search [19], exploratory search [30], and dynamic search [32]. The differences in these tasks are subtle. From 1997 to 2002, the TREC Interactive Track [22] ran for session-based search with a human user in the loop. From 2010 to 2014, the TREC Session Track [11] evaluated the effectiveness of the last iteration of retrieval in a session with the previous iterations' search history provided. Most recently, the TREC Dynamic Domain (DD) Track [32] revitalized interest in the field when it was first introduced [31], by providing a simulated user, called the 'jig,' to interact with and provide feedback to the search system in real time. In this thesis, we use the term 'dynamic search' for this type of search process as it includes many dynamically changing aspects, from the topic to the user's real time interactions.

In recent years, research on dynamic search has relied on methods such k-means clustering [8], Markov Decision Process [32] or passage language models [1] to retrieve more documents to provide the user. Albahem, Spina, Cavedon and Scholer, in their submission to the TREC Dynamic Domain track [1] use passage representation to rank documents and expand queries. This method of utilizing the passage-based representation given in the relevance feedback focuses on the use of the passages returned by the user for relevant documents to produce the query for the next iterations. Each run is capped at 10 iterations, with very little discussion on the stopping condition or decision used to select 10 iterations as the number of iterations for each topic. Moraes, Santos and Ziviani [20], along with their proposed search system, discuss stopping strategies in dynamic search. They propose three strategies: stopping after a fixed count of 10 iterations, stopping after 10 off-topic documents returned, or stopping after 10 off-topic documents returned in a contiguous window. They found that their best run utilized their second stopping strategy, stopping after a total of 10 off-topic documents are returned. These strategies still involve user discretion, a condition or variable set before the run and is manually optimized. Most other algorithms still utilize a hard-coded stopping condition such as 2 iterations or 10 iterations for their system [5, 8]. While these algorithms focus on the movements of the search process within a topic and manually specify a stopping condition for each topic, in this thesis we introduce the idea that the movement to end search for one topic and move on to the next topic search can also be decided by the algorithm by utilizing the features of the context of the search.

2.2 Reinforcement Learning

In this thesis, we propose using reinforcement learning to model the dynamics seen in a complex search process. Reinforcement learning is a class of machine learning algorithms which aim to optimize long term rewards instead of short term gains [9]. Reinforcement Learning has been successfully applied to robotics [9, 12], resource-bounded information extraction [10], natural language grounding [4, 28], and artificial intelligence in general. A reinforcement learning model often contains elements such as states, actions, rewards, beliefs, transitions, and policies, which will vary for different applications. For example, research done by Kotov and Zhai [14] defines states as queries while Luo, Zhang and Yang [19] uses user's decision making status as states, and Zhao [37] and Yue and Joachims [34] make actions in their algorithm be documents in their work, while Luo, Zhang and Yang [19] calls different ad-hoc search algorithms actions.

By emphasizing and capturing the dynamics in a process and reacting according to the environment, reinforcement learning algorithms match well with the characteristics of dynamic search, especially one with a long term search goal, as well as with a trial and error setting. It is therefore not surprising that these approaches have shown promising results in information retrieval problems.

2.2.1 Multi-Armed Bandit

Multi-armed bandit approaches are a popular reinforcement learning algorithm used in solving the dilemma of exploration and exploitation in RL. In the field of reinforcement learning and multi-armed bandits in particular, exploration is defined as the search for the action that will provide the best reward, while exploitation refers to the repeated use of that action with the best reward in order to reap the benefits from that action [15]. Historically, they are often used for recommender systems and ad selection systems, though they have been expanded to solve more modern problems such as large-scale electric vehicle charging [33] and LTE and WiFi coexistence [24]. The exploration vs. exploitation dilemma deals with the trade-off between exploring the different actions to find the one that maximizes the reward, and exploiting the action with the highest estimated reward. The classic exploration vs exploitation strategies, presented by Kuleshove and Precup [15], include ϵ-greedy, Boltzmann exploration, pursuit, and reinforcement comparison. In these models, an arm is considered an action. In the ϵ-greedy algorithm, the bandit uses a parameter ϵ to control how exploratory a process is: the bigger the ϵ is, the higher chance that a random arm instead of the arm with the maximum reward would be chosen in the next run, and the more exploratory the strategy is. Boltzmann exploration and reinforcement comparison both use the Softmax function to model the probability of choosing an arm in the next run.

Modern Bandit models include contextual bandit and dueling bandit. The K-armed dueling bandit algorithm [34, 35, 40] minimizes regret using noisy comparisons. This is useful for cases where it is difficult to obtain an absolute observation of the payoff, but simple to obtain relative judgment of the payoff. Feedback can be obtained then, from pairwise comparisons instead of absolute measures. A direct application of this work is a search system where, among K built-in retrieval functions for the search engine, the one that will retrieve the best results should be chosen. Reward is defined, then, with pairwise feedback based on the user clicks in the interleaved rankings. In other words, the payoff is the user's satisfaction, which is easier to describe using pairwise comparisons than absolute measures.

Multi-armed bandit research recently has focused on expanding solutions to more complex versions of the problem. Komiyama et. al [13] proposed an extension of the Thompson sampling algorithm for use with the multiple-play multi-armed bandit problem, a case in which several arms can be selected, in order to achieve an optimal regret bound. Carpentier and Valko [7] utilize an algorithm that minimizes simple regret for an infinitely many-armed bandit problem, or a version of the multi-armed bandit problem with too many arms to try each one. There has also been research done on the properties and suitable algorithms for a multi-armed bandit problem in which the rewards are measured qualitatively rather than quantitatively [26].

2.2.2 The Contextual Bandit

Contextual bandits [38] are defined as a multi-armed bandit problem with side information, or context, which can be applied to information retrieval as well. Shivaswamy and Joachims [23] look at multi-armed bandit with historical data, which provides some information before the algorithm begins. To make use of the historic data, they extend three different versions of the UCB (Upper Confidence Bound) [2] policy, UCB1, UCB3 and UCBV, to incorporate a logarithmic amount of historic data. For these bandit algorithms that use contextual information, that context can be anything from historical data to user information which provides more contextual information as to which action would produce the most rewarding result. In general, contextual bandits have been used for recommender systems [3, 27, 36, 39]

There has also been research done for both warm-start problems, when there is logged historical data given [25], and cold-start problems, when new users enter the system and previous users can be used as context information [6, 21]. Other contextual bandits use the contextual features of the current action selection to influence the bandit selection policy. Langford and Zhang [16] proposed the epoch-greedy algorithm for the contextual multi-armed bandit problem. The epoch-greedy algorithm runs exploration and exploitation phases in epochs, where in each epoch exploration followed by exploitation is performed, and at each round an arm is chosen using the current context x. In 2010, Lihong, Langford and Schapire [17] introduced LinUCB, or linear UCB, a contextual bandit that associated feature vectors with different actions in order to make informed action selections based on the current features present for that round. LinUCB assumes a linear relationship between the context features and the expected reward of an arm.

The goal each round for a contextual bandit algorithm is the same as the original bandit problem, to maximize the reward received from selecting an action each round. However, with the addition of a historical context, contextual bandits utilize this context alongside the reward received in previous iterations to select an action. To model the dynamic search process, we utilize a contextual bandit algorithm.

Chapter 3

The Dynamic Search System

A dynamic search engine interacts with the simulated user to find relevant documents. The framework proposed in this thesis uses a LinUCB contextual bandit algorithm built on top of the search engine to interact with the simulated user in order to search within topics and move from topic search to topic search.

At each iteration t, the bandit is provided with contextual information surrounding where the search currently is and what direction previous searches were moving. This feature vector is used by the bandit, along with the reward information about each action, to select one of four actions to play for that iteration. The action selected is performed, and the search engine is queried once more with the new query, and the results passed to the simulated user for relevance feedback. The relevance feedback is then used to provide a reward for the action selected. This process is repeated until the search is finished. The intuition behind this is the idea that the information about where the search currently is, the context features, as well as the relevancy of the previous search, dictate the direction that the search should continue to move in, the action that the bandit should select.

3.1 LinUCB

While the contextual bandit algorithms like LinUCB were originally designed for personalized news recommendation, it can be adapted to other problems as well. This thesis will adapt the LinUCB algorithm to dynamic search.

LinUCB, as created by Lihong, Langford and Schapire [17], is used to select news articles to show a user based on the contextual information gathered from the user and articles. It takes a context feature vector at each iteration and uses these features to estimate the reward for each action that iteration.

The expected reward of an action a is presumed to be linear in its feature vector $\mathrm{x}_{t,a}$, using a coefficient vector that is unknown but estimated, denoted $\hat{\theta}_a$. In the LinUCB algorithm, the action that is chosen in each iteration is the action that maximizes the score given from the estimated coefficient vector multiplied by the feature vector, as shown in eq. 3.1. $\sqrt{\mathbf{x}_{t,a}^{\top}\mathbf{A}_a{}^{-1}\mathbf{x}_{t,a}}$ is the standard deviation of the expected reward, where $\mathbf{A}_a{}^{-1}$ is the inverse of a matrix made by adding a design matrix corresponding to the previous contexts to an identity matrix the size of the context.

$$a_t \stackrel{\text{def}}{=} \operatorname*{argmax}_{a \in A_t}(\mathbf{x}_{t,a}^{\top}\hat{\theta}_a + \alpha\sqrt{\mathbf{x}_{t,a}^{\top}\mathbf{A}_a{}^{-1}\mathbf{x}_{t,a}}) \tag{3.1}$$

In their paper, Lihong, Langford and Schapire [17] demonstrated that LinUCB showed improvements on personalized news recommendations when compared to more simplistic methods, such as ϵ-greedy methods. Specifically, the use of context features to select the action, or the article recommendation, does show evidence of benefiting the system.

In our adaptation of LinUCB, we emphasize the use of the context surrounding the search iteration, such as what the user found relevant and where previous actions have moved to in the document space, to dictate what the appropriate next action should be. Based on this idea, our approach uses these context features to exhaustively search a topic and then move on to the next topic. By using a bandit algorithm, we focus on maximizing the selection of relevant documents through a combination of

Table 3.1: Actions and their associated search directions

Action	Search Direction
Add	Move into a more detailed subtopic "Kaci Hickox"$\rightarrow$"Kaci Hickox actions"
Remove	Broaden search from subtopic "dynamic modeling"$\rightarrow$"modeling"
Weight	Search for more documents within the same subtopic "1.0 US 1.0 Military 1.0 crisis 1.0 response" $\rightarrow$"0.8 US 0.85 Military 1.0 crisis 0.9 response"
Change	Move from one topic space into another "alleged alternative Ebola treatments" $\rightarrow$"urbanization/urbanisation"

exploring the different directions, or actions, the search can take and then exploiting that direction, or action, of the search.

3.2 Bandit Actions

The bandit has four actions: add, remove, weight and change. These actions modify the query in order to move the search in different directions. The first three actions, "add," "remove," and "weight," traverse the topic through a widening or narrowing of the current subtopic, while the action "change" is akin to ending the traversal of one topic space and moving to another.

3.2.1 Add

The action "add" provides more detail to the search by adding another term to the query. To select the most relevant word, we use a form of tf-idf to select the word w from the set of words in the retrieved relevant documents D. 3tc The score s_w of any given word is the maximum tf-idf score received from that word in any document in the set of relevant documents D, as seen in Eq. 3.2. The word with the highest score is considered to be the most important or relevant word found in the retrieved document, and is therefore selected to be added to the query.

In the set of words from the relevant documents, stop words are stripped out to prevent a stop word devoid of useful meaning from being added to the query.

3.2.2 Remove

The action "remove" also applies this tf-idf to select which word to remove. Notably, while the "add" action uses the set of relevant documents, "remove" uses the set of off-topic documents in the list to generate each tf-idf score. This generates a set of "important" non-relevant words retrieved in the last iteration. We then select the word in the current query that has the highest score to be removed. By removing a word from the query, the search is widened to look at more subtopics within the topic.

3.2.3 Weight

The action "weight" modifies the weights to the terms in the query, allowing the focus to shift between the terms in the query. This can be considered a continued search within the same subtopic in order to find more relevant documents on the same or similar information. The terms in the query which are found in the top 20 most frequent words in the relevant documents or in the top 20 most frequent words in the irrelevant documents have their weight adjusted accordingly. The word is assigned a score from 1 to 20 based on its rank in the list (0 being the score for the most frequent and 19 for the 20^{th} most frequent), and the term's weight is increased (if it was on the relevant list) or reduced (if it was on the irrelevant list) in proportion to its frequency ranking according to equation seen in Eq. 3.3, with a maximum of 0.2 being added or subtracted from the weight at any one time.

$$w_t = w_{t-1} \pm .2(\frac{(20 - r)}{20}) \tag{3.2}$$

3.2.4 Change

The action change strips the topic name of any stop words and then changes the current query to the topic name. This effectively transplants the search from one area within the document space to another to begin search on another topic. By considering the end of one topic search and the beginning of another one as a single action, the bandit is able to predict the most ideal end to a topic search based on the reward estimation of transplanting to a new topic space compared to the other directions represented by the other actions.

These four actions represent varying movements the algorithm can take when searching for relevant information (see table 3.1). The action "add" can be considered a narrowing of the information space, traversing into a more specific subtopic, as new terms are added to the query to indicate more detail about what is being searched. In the other direction, "remove", through the subtraction of terms in the query, is the broadening of the search from a subtopic to a wider range of subtopics, allowing more exploration through the topic space. The action "weight" is a continued exploitation of the current subtopic being searched. By adjusting the weights to the current words in the query, the system can find different documents of the same subtopic and thus exploit the current search direction. Finally, the action of changing the topic is the movement from one area within the information space to another. In conjunction with each other, the bandit can use the context of the current iteration to move the search in a novel direction in order to retrieve more relevant results and keep the user engaged and interested in the search.

3.3 Bandit Reward and Context Features

In bandit algorithms, the reward selected is crucial to the bandit application as this is the value that the algorithm will work to maximize over each iteration. For the LinUCB algorithm specifically, the context features chosen to represent the context of the iteration will greatly affect how the algorithm calculates the next action to play. The following sections describe the context and reward that the system proposed in this thesis uses to perform the dynamic search task.

3.3.1 Context

This thesis proposes the use of the search context in order to locate the search within the current information space and inform the system with the direction and action it should next take. The context surrounding a search places the current iteration into the document space and gives relevant information to how long the user has been searching and how far into the search they currently are.

The context given to the LinUCB model consists of six features. These features span different aspects of the previous iteration(s) in order to provide the bandit with more information about the current state and direction of the search. The features given are the number of documents seen so far in this topic, the number of relevant documents retrieved in the last iteration (this is different from the reward given at the end of the iteration, which is the sum of relevance scores given to all noted relevant passages returned by the jig), the total number of relevant documents retrieved during the search of the current topic, and the number of times the actions weight, add, and

Table 3.2: The features of the search

#	context feature	explanation
1	# of documents seen	how far search is
2	# of relevant documents last iteration	current effectiveness of search
3	# of relevant documents this topic	depth of search in current topic
4	# of times action weight called this topic	path of movement
5	# of times action add called this topic	path of movement
6	# of times action remove called this topic	path of movement

remove, respectively, were chosen for this topic. The action change is not included because by nature of this action, this feature is always the same for each topic. The features chosen to inform the bandit of the current context describe the search in terms of its depth, current effectiveness and the path of movement that has led it to its current direction.

Using the reward and the context provided, the LinUCB bandit selects the action to play at each round that will maximize reward.

3.3.2 Reward

After each action, a reward is given to denote the effectiveness of that action. The reward r_t, provided to the bandit for a selected action for iteration t, is

$$r_t = \sum_{p \in D} RS_p \tag{3.3}$$

where RS is the relevance score for each relevant passage and p is a relevant passage from the set of documents D retrieved that iteration. The relevancy score for a relevant passage is a score from 1 to 5 attached to the passage, with 5 meaning extremely relevant and 1 being only somewhat relevant. As there is no limit to the number of relevant passages a document can have, there is no limit to the relevancy score a document, and by extension, an iteration, can have. Note that the reward is never negative but it can be 0 if there were no relevant passages, which would show that the action was not effective at all in retrieving relevant documents for that iteration.

3.4 Search and Retrieval Framework

To implement the LinUCB bandit, we used the open source python package Striatum bandit[1] to implement the bandit portion of the algorithm. We used the Indri 5.8 search engine[2] for the retrieval portion of the algorithm. After an action is selected and the query modified accordingly, the query is submitted to the search engine and the documents retrieved passed to the simulated user, the "jig" (provided by TREC) for feedback.

Since a relevant document is no longer relevant the second time it is seen, we prevent duplicate documents being retrieved for any one topic, by assigning a document prior value of $-1\mathrm{e}{-24}$ to a document that has been retrieved once for a specific topic, to reduce the chance that it is retrieved again. This prior value was selected experimentally as the magnitude needed to remove all previous documents from future retrievals. These values are topic specific, so they are removed for the next topic, allowing the same document to be retrieved for a separate topic.

[1]https://github.com/ntucllab/striatum
[2]https://www.lemurproject.org/indri/

Chapter 4

Evaluation of the Dynamic Search System

The measure of a dynamic search system's effectiveness is the satisfaction of a user's information need over multiple iterations for each topic within a topic domain. Dynamic search metrics measure the effectiveness of a search system as time goes on, not just the overall effectiveness by the end.

4.1 Cube Test Metrics

To measure the effectiveness of the LinUCB algorithm, we use the Cube Test (CT) and Average Cube Test (ACT) [18], the official TREC metrics for Dynamic Domain. The Cube Test and Average Cube Test evaluate the speed that the search is completed. The tests looks at how quickly the task's "cube" can be filled with relevant information over a diverse set of documents. The "cube" in this case represents the actual search task and the entire information need that the user has. As relevant information is found, the cube is "filled up" and more and more of the user's information need is satisfied. This metric takes into account both the amount of relevant documents that are found as well as the rate at which they are found. Compared to the Cube Test, the Average Cube Test looks at the time it takes to complete a search task over the averages of the cube test values.

At each iteration, the search system runs a (new) query and retrieves a list of five documents, these documents are then scored in terms of their information gain to the search task at hand, taking into account their diversity compared to the previously retrieved documents and how long the search is going on. The higher the score is for the cube test metric, the better it is, as a higher score signifies that the documents led to a large information gain for the current search topic. It is also important to note that the cube test metric is an average speed function that measures the change in the information gain, or the volume of the cube over a given time period [18]. This means that the scores should decrease at each iteration, as it becomes harder and harder to increase gain as the cube becomes filled with information.

We compare the cube test metrics of our system with the evaluation of the TREC-DD 2016 baseline results as well as the TREC median results. The median results are calculated among the submitted results listed in the TREC-DD 2016 Overview paper [32]. These comparisons serve to place the contextual bandit system results in context with the current state-of-the-art dynamic search systems.We also provide comparisons with a submitted run that utilizes a relevant passage language model to rank its documents. This algorithm represents a vastly different approach than the one presented in this thesis, as it focuses on the passages and documents retrieved rather than the context of the search and the actions at each iteration. This comparison provides a more direct example of the effectiveness of the contextual bandit algorithm as compared to other systems.

In order to depict the system's run more completely, the different CT results are compared over 10 iterations. This allows the systems to be compared in terms of their effectiveness over time as well as their objective overall effectiveness.

There are other evaluations metrics that can be used to discuss dynamic search systems, as dynamic search is still an information retrieval task. These metrics are not included in this discussion as the Cube Test and Average Cube Test are considered the official metrics used by TREC-DD and are the one metric that is consistently used to evaluate all system runs for the Dynamic Domain task.

4.2 Experiments

4.2.1 Dataset

The contextual bandit search system proposed in this thesis is evaluated on its effectiveness in searching broad topic domains through dynamic search. Using a complex search topic or information need, the system's task is to search through the domain iteratively and retrieve documents that are relevant to the topic.

Experiments were run using the Ebola dataset provided in TREC Dynamic Domain 2016. This topic domain is based on the Ebola breakout in Africa from 2014-2015. The dataset consists of over 600,000 documents and 27 topics, which range from "urbanization/urbanisation" to"US Military crisis response." Documents consist of tweets about the Ebola outbreak, web pages from sites that were from affected countries, as well as PDFs retrieved from organizations like The World Health Organization, Financial Tracking Service and The World Bank. We use the ground truth and topics provided by TREC to retrieve documents and generate relevance feedback from the simulated user.

The data was created using human annotation as the basis for the relevance scores that are provided by the simulated user for each topic. The topics are formatted like a broad query and contain multiple subtopics that deal with a more specific idea touched on by the topic. These subtopics are not available for the system to see and must be discovered during the search. It is up to the system to exhaustively search the subtopics within each topic in order to satisfy the user's full information need. An

```
        <domain name="Ebola" id="2" num_of_topics="27">
                <topic name="US Military Crisis Response" id="DD16-1"
num_of_subtopics="3">
                        <description>Who are key leaders (field grade officers and senior
NCO's) in charge of U.S. military units combating the ebola epidemic in Africa, what are
the protocols for personnel safety, and what is their mission? </description>
                        <narrative>Units from all four services were deployed to West
Africa, some 3000 personnel in all, in support of U.S. and international aid workers. The
military was not directly involved in treating ebola patients. Their mission was to
provide support services such as facilities construction, logistics, telecommunications,
road building, etc. so that the medical aid workers could function more effectively.
Additionally, the research addressed safety protocols for the U.S. forces, including
mandatory quarantines upon return to their home bases. Non relevant information
encompasses non-U.S. military units and civilian aid workers and U.S. military units that
did not deploy to Africa.</narrative>
                        <subtopic name="West African mission" id="DD16-1.1"
num_of_passages="1862">
                                <passage id="25268">

<docno>ebola-634445eda14aa2756fbd3eff24b0ccf10f24543c4da9bdd54cbb354c46ba5c66</docno>
                                        <rating>3</rating>
                                        <text><![CDATA[In this week's AFRICOM Update
engineers continue to build Ebola Treatment Units in Liberia while a special facility for
infected healthcare workers nears completion.]]></text>
                                        <type>MANUAL</type>
                                </passage>
                                <passage id="25273">
```


Figure 4.1: A section of the ground truth document used to provide the basis for the relevance feedback and topics within the Ebola domain.

example section of the ground truth document with this information provided within it is seen in figure 4.1.

The contextual bandit system was run on the Ebola dataset from start to finish, with the topic name being seeded to the system as a query to start the search of each topic. The system automatically edits or changes the query as decided by the bandit algorithm.

4.2.2 Topic Traversal

First, we decided to look at the contextual bandit algorithm's run solely through its changing CT and ACT score, separate from other submitted runs and algorithms. Using an automatic topic completion method means that we must show that the method the algorithm chooses to use to change topics still exhaustively traverse the topics and provides enough iterations to retrieve relevant results for the user.

The overall CT and ACT scores of the contextual bandit's run over 15 iterations gives a fuller picture of how the algorithm searches the topics and whether the topics are being searched effectively and thoroughly. By first analyzing the results separate from the other submitted runs, we are able to better understand the algorithm's effectiveness objectively.

4.2.3 Non-Automatic vs. Automatic Topic Completion

The two main focuses of this thesis are the use of context in the search system, through the contextual bandit algorithm, and the automation of the the topic completion portion of dynamic search, through the addition of a "change" action. In order to better understand the benefits that arise from both ideas proposed, we implement a contextual bandit search system identical in every way except with the "change" action eliminated. Instead, we use a hard-coded 10-iteration stopping condition in the search system, and compare its performance with the system proposed in this thesis. This comparison allows us to narrow in on the benefits that arise from using context in the search process separate from the influence of automating the topic search completion.

4.2.4 Baseline Comparisons

The effectiveness of the contextual bandit search system proposed in this thesis can be represented by the scores seen in the first and tenth iteration of the system. These two scores show both the initial, or overall effectiveness of the search system as well as the effectiveness of the system over time. Furthermore, since the contextual bandit search system automatically decides when to complete one topic search, the contextual bandit has a variable number of iterations for each topic, so the tenth iteration allows us to compare the contextual bandit system search efficiency. even with its variable search length, through this common end iteration as used by other algorithms.We compare our system run to the the other systems, the baseline algorithms, through the first and tenth iterations.

TREC Baseline

We chose to compare the system to the TREC Baseline run as submitted by Georgetown University to TREC-DD 2016. This run is considered a minimum scoring run that was created through the use of the most basic retrieval. The TREC Baseline sits lower than all of the other submitted runs of the previous runs as it does not use a particular algorithm to improve on past iterations.

Comparing the contextual bandit's run to the baseline run provides useful information as to whether the contextual bandit search system improves upon the initial system and does in fact benefit the dynamic search process.

Table 4.1: Algorithms compared to the contextual bandit system

Baseline Algorithm	Description
TREC Baseline	A basic retrieval run to show the algorithm's effectiveness over a basic search system
TREC Median	The median CT and ACT score of all TREC submitted runs for each iteration
TREC Best	The best CT and ACT score of all TREC submitted runs for each iteration
rmit-lm-psg-max	A submitted run that uses a theoretically different model to retrieve relevant documents

TREC Median

We also compare our system to a set of TREC median scores, taken from the submitted runs in TREC-DD 2016. The median scores do not come from a single submitted run but is calculated by taking the median of the scores of all of the runs listed in the TREC Dynamic Domain 2016 Overview paper [32]. This set of median scores is representative of the average dynamic search process and is an amalgamation of 3 different algorithms over the first and tenth iteration scores seen in this experiment.

By looking at how the contextual bandit search system fares against the median scores of 2016 TREC Dynamic Domain, we can ascertain the effectiveness of our algorithm compared to official algorithms researched and submitted by teams around the world. Furthermore, the median gives a fair representation of the systems available currently, from the less effective to the most effective ones.

TREC Best

We also look at the set of best TREC scores from each iteration. These scores are combined from all the different submitted runs in TREC-DD 2016, calculated by finding the maximum score at each iteration listed in the TREC-DD Overview paper [32]. However, the scores from one submitted run are not included in the TREC Best scores as the submitted run was a manual run utilized to find an upper bound from a set of documents.

Using the TREC Best scores allows us to see how our algorithm compares with the best state-of-the-art systems found in the current research.

rmit-lm-psg-max

To provide a more substantial comparison between our system and another one within the TREC Dynamic Domains submitted runs, we selected a system from the group of algorithms listed in the overview paper [32].

The selected system from the set of submitted runs, rmit-lm-psg-max, is a run that is relatively median in its evaluation scores, providing a direct comparison to an average system being used for dynamic search. The submitted run uses a very different approach than the one proposed in this thesis, which also provides a useful counterpart for the contextual bandit algorithm to show how it fares compared to a language-based dynamic search system, a system that uses a completely different method to retrieving relevant documents for the user. rmit-lm-psg-max uses a passage language model to score documents, with the maximum score of the document's passages used as the document's overall score.

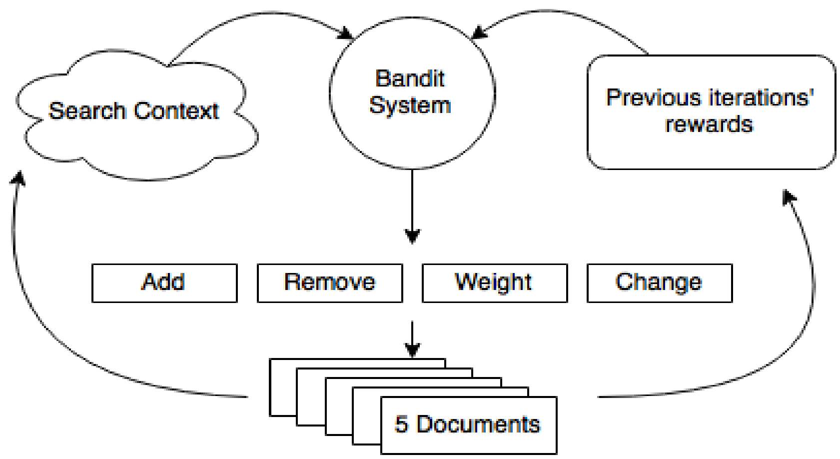

Figure 4.2: The creation of the bandit search policy

This algorithm was a submitted run to the TREC 2016 Dynamic Domain track, thus providing a concrete comparison to search systems that are currently being researched and designed in the dynamic search field.

4.2.5 Bandit Search Policy

We also investigate the details of the search system proposed by this thesis through the policy that the bandit algorithm uses to iterate through the complex search. We examine the actions that the bandit selects at each iteration based on the reward from the previous iteration(s) as well as the context given to the system for the current iteration. This gives us an understanding of how the system moves within the topic space from topic to topic and the search system's effectiveness in a detailed manner. Our bandit search policy can be visualized as a process in figure 4.2, where the previous rewards received and the current search context are used to select an

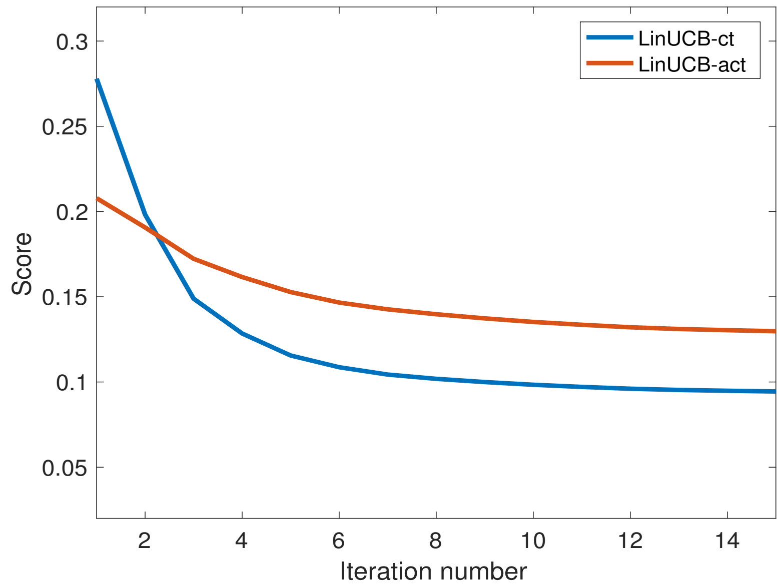

Figure 4.3: The CT and ACT scores of the system over 15 iterations.

action used to re-formulate the query and retrieve 5 documents. The feedback from the documents is then used to create a reward and update the context. Using this format, we will explore how the bandit uses this process to further the search at each iteration of the search.

The policy chosen by the bandit describes the shape and direction that search takes and the decisions that the algorithm uses in order to traverse the information space. The policy decides how to move the search in a way that will retrieve more relevant documents for the user.

4.3 Results

4.3.1 Topic Traversal

The dynamic search system proposed in this thesis was tested on the TREC Dynamic Domain 2016 Ebola dataset, which provides 27 topics for the system to run through within the topic domain of "Ebola." In figure 4.3, the CT and ACT results are seen graphed over time. This provides a visual representation of the system's efffectiveness over time. The system shows an initial peak score of 0.278 for its CT score and 0.208 for its ACT scores. These gradually decrease over time and begin to level out after around 6 iterations.

The contextual bandit search system is also observed to have an average number of 7 iterations per topic, though each individual topic could range from a mere couple iterations to 26 iterations before the system ends the topic search. These results indicate that while the system did have variance among the search lengths between topics, the system did in fact have some way for telling that the topic had adequately searched the topic space. The system is able to traverse the topic and retrieve relevant documents for multiple iterations but the system does not allow the search to continue on for too many iterations and frustrate the user with the length of time.

As other systems use fixed iterations or a manually specified stopping condition through each topic, the results are focused on maximizing the results returned from the Cube Test. Thus, our results for the algorithm's topic traversal separate from the other submitted runs provides valuable information about the algorithm itself. As figure 4.3 shows the scores of the contextual bandit algorithm returned over 15 iterations, it allows us to better understand how the system performs over time and see the effectiveness of allowing the algorithm to decide when to stop one topic search for another by itself. The results shown in figure 4.3 indicate that the algorithm is

Table 4.2: A comparison of the scores of the non-automated topic completion system and the automated topic completion system

Algorithm	Iteration 1	Iteration 5	Iteration 10
Non-Automatic CT	0.277	0.090	0.049
Automatic CT	0.277	0.116	0.098
Non-Automatic ACT	0.208	0.141	0.101
Automatic ACT	0.208	0.153	0.135

capable of detecting when the topic has been exhausted and retrieving more results would no longer be effective.

4.3.2 Non-Automatic vs. Automatic Topic Completion

In order to better understand how the two contribution in this thesis, we compare our system with an identical system that doesn't use automated topic completion but a plain 10-iteration stopping condition. As can be seen in table 4.2, the two systems start out with identical CT and ACT scores that both gradually reduce through the iterations. The non-automatic topic change system, however, shows a faster decline in CT and ACT scores, with the scores at iteration 5 slightly lower than the automatic system but drastically lower by iteration 10.

This shows the benefits the the automated topic search completion functionality has on the search, as higher scores for the automatic system could show that the system is able to tell when the search has been exhausted and further searching will not lead to much information gain. In terms of the use of context in the search process, the slightly reduced scores in iteration 5 show that the search context used in the search process are effective as a dynamic search algorithm but benefit from the automated topic completion as continued iterations lead to a continued decline in information gain in each iteration.

4.3.3 Search Effectiveness

Table 4.3 shows the CT and ACT results for the LinUCB algorithm compared to another submitted run as well as some baseline values from TREC 2016 submitted runs. The contextual bandit search system shows an initial CT and ACT score similar to the TREC baseline run, but by the 10^{th} iteration, the contextual bandit search system has improved on the baseline's CT and ACT scores by 0.063 and 0.049, respectively. Similarly, we observe a similar CT and ACT score as the TREC median of all submitted runs at the 1^{s} iteration, with results actually slightly higher initially. By the 10^{th} iteration, our system shows a slight increase above the median of 0.022 for the CT score and 0.016 for the ACT score.

It is also important to note that because our algorithm handles topic search completion automatically, rather than being manually set before a run begins, it is considered a success that our system is able to run on par with the other submitted runs to the TREC Dynamic Domain 2016 track. Since this contextual bandit system removes an entire subset of the dynamic search problem that must be set or optimized by the researcher otherwise, running at similar results to the median of other submitted search systems is satisfactory.

Compared to TREC Baseline

The results seen in table 4.3 show that while the contextual algorithm's scores begin at very similar values to the baseline scores, by the 10^{th} iteration, they are significantly better. This indicates that the contextual bandit search system does in fact provide a boost in effectiveness to the dynamic search system. The significant improvement on

Table 4.3: Search effectiveness from Iteration 1 to Iteration 10 on TREC DD 2016 Ebola Dataset (* indicates stat. significant improvement over baseline, † indicates stat. significant improvement over rmit-lm-psg-max, ($p<0.05$, t-test, one-sided))

Algorithm	CT@1	ACT@1	CT@10	ACT@10
LinUCB	0.278	0.208	0.098*†	0.135*†
rmit-lm-psg-max	0.246	0.268	0.049	0.089
TREC Baseline	0.294	0.206	0.035	0.086
TREC Median	0.260	0.187	0.076	0.119
TREC Best	0.319	0.236	0.176	0.150

the baseline score shows clear evidence that the contextual bandit model of selecting actions at each iteration benefits the search system and retrieves more relevant results both more often and more frequently. The contextual bandit algorithm, then, can be adapted to be used as a dynamic search system as well as a personal news recommendation system and ad recommender.

Compared to rmit-lm-psg-max

Similarly, our bandit algorithm is able to achieve results significantly better than the passage language model approach rmit-lm-psg-max by the 10^{th} iteration. This submitted run, using a very different approach than the algorithm proposed in this thesis, also starts off with relatively comparative results in the 1^{st} iteration but, by the 10^{th} iteration, our search system shows significantly better CT and ACT scores. These results indicate that the contextual bandit provides beneficial research in the current dynamic search field and provides results that can be compared to actual search systems currently being proposed.

Compared to TREC Median

Our algorithm achieves similar results compared to the median results among all submitted TREC runs. The results seen by the contextual bandit search system are comparable to the median scores seen from the 1st iteration to the 10th iteration. Statistically, we are not significantly better or worse than TREC median scores. Our system, then, is on par with the current median of the state-of-the-art search systems currently being researched. This shows that the contextual bandit system proposed in this thesis achieves results similar to the average search system that can be found, as given by the runs submitted to TREC. Furthermore, as our system proposes a layer of automation beyond other search systems, this result shows that the addition of completing the topic search as a decision left to the system does not negatively affect the system when compared to the average dynamic search system found in the dynamic search field.

Compared to TREC Best

Form iteration 1 to iteration 10, our algorithm shows slightly lower scores compared to the aggregated best TREC scores from TREC-DD 2016, with an exception of the CT score at iteration 10 being considerably lower. These scores show the potential of our system to improve, as our scores are comparable to the median scores but are lower than the best scores. As our system uses relatively basic algorithms for its actions, the number of relevant documents retrieved and the overall search effectiveness are lower than what they could be. As such, our system is currently limited by the algorithms implemented for the bandit's actions. Through the improvement of our current actions' algorithms, optimized to be more efficient in retrieving relevant documents, our scores have the potential to be raised higher. Furthermore, the potential for our system to improve with the bandit actions would not only show an overall increase in the scores from each iteration, it would also show a slower decrease in CT and ACT scores over the iterations as the use of more efficient actions would allow the bandit to better exploit the action that provides the highest reward. Overall, though our scores are lower than the TREC best scores over the iterations, this comparison shows the potential in our system to improve and increase our scores to the level of the TREC best scores.

4.3.4 Bandit Search Policy

By allowing the bandit to decide when to end the search of each topic, the number of iterations given each topic varied greatly. The number of iterations for a topic typically range from 6 to 12. The policy created by the bandit focused on retrieving as many relevant documents each iteration as possible. Thus, when the bandit does not retrieve relevant documents through one action, it quickly changes to move in a different direction in the information space in order to find other relevant documents. We can see in an example from a relatively difficult topic (one which consistently shows smaller results among all the submitted runs to TREC) in Table 4.4, when changing the weights does not yield results, the bandit moves instead to specify more detail through adding terms, which does yield relevant results. It then continues to add more detail in order to retrieve even more relevant documents. The bandit repeats this process, exploiting actions that retrieve many relevant results as well as exploring which action will retrieve the most relevant documents until it decides to complete that topic search and begin the process over again.

In figure 4.4, the distribution of the different actions are broken down. The action “change” is not included in this pie chart as it is only called once per topic search and as such will be called 27 times in the entire dynamic domain search on the Ebola

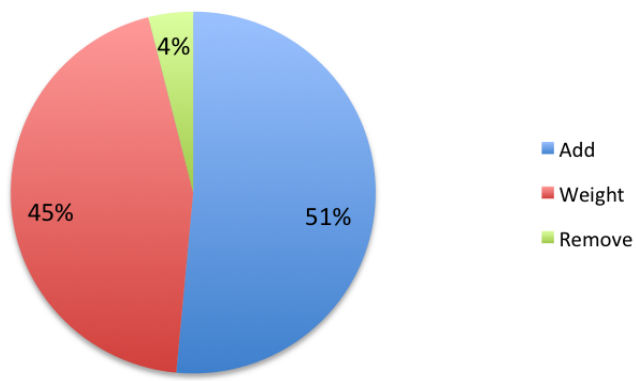

Figure 4.4: Bandit action distribution chart

topic. The actions "add" and "weight" are both called at relatively equal proportions, while "remove" is called less frequently than the other two actions. This could signify that the "remove" action as a whole is less successful in retrieving relevant documents and as such was not exploited as much as the other two actions. However, all 3 actions are called, showing that they are all explored to find the best action in the search process.

The policy seen in table 4.4 depicts the system run through topic 3 of the Ebola dataset, a particularly difficult topic titled "Healthcare impacts of Ebola." This example shows how the bandit search policy deals with a topic that starts off without any relevant documents. Using the information gathered from the non-relevant documents and the context of the search within the topic space, the bandit is able to move the search in a direction that retrieves relevant results for the user. The policy demon-

Table 4.4: Example policy created by contextual bandit search system

Iteration	Action	Motivation	Result
1	**Weight** "1.0 healthcare 1.0 impacts" → ".8 healthcare 1.0 impacts"	bandit exploits the current direction and subtopic	no relevant documents
2	**Add** "1.0 healthcare 1.0 impacts" → ".8 healthcare 1.0 impacts 1.0 reveals"	bandit moves to more specific subtopic	1 relevant document
3	**Add** ".8 healthcare 1.0 impacts 1.0 reveals" → ".8 healthcare 1.0 impacts 1.0 reveals' 1.0 declares"	bandit continues to add more detail	1 relevant document

strated in this example provides a concrete example of the bandit's effectiveness in dealing with this complex search.

4.4 Discussion

As observed in the previous sections, the dynamic search system we propose is effective in searching a broad topic over multiple iterations. The scores seen over 15 iterations indicate the system's effectiveness in retrieving relevant documents quickly and efficiently and then identifying when the search has been exhausted and move on. Since our proposed system takes on another problem within dynamic search the results are considered acceptable in two regards: the scores show that the system is both able to retrieve relevant documents at a quick rate and able to provide enough iterations to the system to search the topic to this extent.

We then attempted to isolated and view the benefits that the two contributions our system proposed—the use of search context in the search system and the automation of the topic search completion aspect of the search task. In doing so, we found that automating the topic search completion did in fact provide a benefit to our system in detecting when it would be a good time to stop a search such that the topic had been exhausted and that further queries would lead to minimal information gain. As for the context portion of our system, we found that the use of the search context to direct the search direction at each iteration was useful and effective but benefitted from the topic completion automation to stop the search before the information gain at each iteration of the search decreased too much.

We also compared our algorithm with other submitted runs to TREC Dynamic Domain 2016, using their scores as a baseline for our system. The compared results showed that the contextual bandit system significantly improves on the baseline submitted to TREC and does significantly better than the contrasting algorithm rmit-lm-psg-max which uses a very different approach than we do. We also saw that our search system had comparable results to the median scores seen in all of the submitted runs' scores, indicating that our system does a satisfactory job at traversing a complex information space in a dynamic search and is not negatively affected by automating the stopping condition (changing topics). Finally, in comparing with the best scores in the TREC submitted runs, we found that though our system did worse, the best scores show the potential for our system to improve as the contextual bandit search system's action algorithms are improved.

These results show that using the context of the current search iteration to select the next action in the search provides a good method for navigating the information space and finding relevant documents for the user. The context features of the current search enables the algorithm to better decide the direction of the search through each iteration and react to the feedback given by the user at each iteration. These results also indicate that not only is LinUCB an effective algorithm to be applied to dynamic search, but that the movements of the different bandit actions throughout the topic space is successful in traversing the topic space and exhausting relevant information. Automatic topic change is thus feasible through the use of a contextual bandit algorithm. Overall, the use of contextual bandit to direct a dynamic search process is evidenced to be effective and comparable to dynamic search systems currently being proposed, and also that the contextual bandit search system has areas of improvement that can lead to an even more effective dynamic search system.

Chapter 5

Conclusion

5.1 Research Summary

In this thesis, we proposed a novel contextual bandit solution to the dynamic search task over a broad topic domain. We present the use of a contextual approach to the task by using the context of the current search environment to decide the next action in the search. We also propose the novel idea of allowing the system to decide the stopping condition completely on its own and provide the system with a choice of "change topic" as an action along with other actions editing the query within the topic itself.

The actions that we propose for the contextual bandit search system, add, remove, weight and change, move the search within the information space in order to find relevant documents quickly and exhaustively. While most systems focus on moving around each topic space, our algorithm considers both the movement within the topic space and the movement from one topic space to another as actions that can be performed during the search. Our system, then, moves the search around the information space of not just the topic that it is currently searching, but the entire domain information space as a whole.

In Chapter 4, we discuss the use of the Cube Test metrics to evaluate our results compared to other submitted runs to the TREC Dynamic Domain 2016 track. The Cube Test and its counterpart, the Average Cube Test, evaluate a search system run on its effectiveness at finding diverse and relevant results at a quick rate over time. Based on our system's interaction with the TREC-DD Ebola topic domain, we found that taking into account the context features of the current state of the search is useful for improving the search's effectiveness over time. The results observed in our contextual bandit system are significantly better than an example submitted run from the 2016 TREC conference and the TREC baseline. The results of our system were similar to the median of all TREC runs, though not statistically better.

Our results also serve to show that our contextual bandit adaptation allowed automatic topic change through dynamic search without negatively affecting the effectiveness of the search system's traversal through the information space. In having similar results to the other submitted runs in the track, we demonstrate that using a contextual bandit to decide this aspect of the search as well can lessen the load of the developer or researcher without a difference in search effectiveness.

5.2 Significance of Thesis

This thesis looks at an approach to the dynamic search task that allows the system to tackle both of the subproblems found within dynamic search–how to traverse the information space for one topic as well as when to end one topic search and start another. Using the context of the search at hand, we propose a contextual bandit approach that relies on this context to move the search around the information space through actions such as adding or removing terms from the query, changing the weights of terms in the query, or changing the topic completely.

Our approach presents a novel improvement on current dynamic search systems which deal with the stopping condition (topic change) by either setting a fixed number of iterations to be run or by setting a stopping condition for the system to check for and stop at. We give the decision of when to start the next topic search within the complex search process to the search system. Furthermore, we present the use of a contextual bandit algorithm, generally used for recommender systems, for the dynamic search task by representing the context as a set of features describing the current state of the search. We then show results that indicate that the use of this proposed system is effective and comparable to current state-of-the-art dynamic search systems.

5.3 Future Directions

Dynamic search is an information retrieval task that is both practical and relatively novel in current times, leading to many research avenues that can still be continued. The process of automation and optimization in this task can still be improved on and research more completely.

In the system proposed in this thesis, we use a weighting scheme based on the ranking of the words of the query in the relevant documents. The add and remove actions of this system use a tf-idf scheme to select a term to add or remove from the query. While these implementations are satisfactory, future works can focus on alternative implementations to these and discuss and compare them.

The task of having the system set the stopping condition to end one topic search and start the next is a novel idea proposed in this thesis. The challenge of selecting a stopping condition has been tackled in many ways in the past, but mostly through manual means. Thus, this area of dynamic search is largely unresearched and has many avenues for further research.

Bibliography

[1] Ameer Albahem, Lawrence Cavedon, Damiano Spina, and Falk Scholer. Rmit @ trec 2016 dynamic domain track: Exploiting passage representation for retrieval and relevance feedback. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016. URL `http://trec.nist.gov/pubs/trec25/papers/RMIT-DD.pdf`.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[3] Djallel Bouneffouf. R-UCB: a contextual bandit algorithm for risk-aware recommender systems. *CoRR*, abs/1408.2195, 2014. URL `http://arxiv.org/abs/1408.2195`.

[4] S. R. K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 82–90, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL `http://dl.acm.org/citation.cfm?id=1687878.1687892`.

[5] Emanuele Di Buccio and Massimo Melucci. Evaluation of a feedback algorithm inspired by quantum detection for dynamic search tasks. In *Proceedings of*

The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016, 2016. URL http://trec.nist.gov/pubs/trec25/papers/UPD_IA-DD.pdf.

[6] Stéphane Caron and Smriti Bhagat. Mixing bandits: A recipe for improved cold-start recommendations in a social network. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, SNAKDD '13, pages 11:1–11:9, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2330-7. doi: 10.1145/2501025.2501029. URL http://doi.acm.org.proxy.library.georgetown.edu/10.1145/2501025.2501029.

[7] Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pages 1133–1141, 2015. URL http://jmlr.org/proceedings/papers/v37/carpentier15.html.

[8] Robin Joganah, Richard Khoury, and Luc Lamontagne. Laval university at TREC dynamic domain 2016: Subtopic extraction focused on named entities. In Voorhees and Ellis [29]. URL http://trec.nist.gov/pubs/trec25/papers/LavalLakehead-DD.pdf.

[9] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[10] Pallika Kanani, Andrew McCallum, and Shaohan Hu. Resource-bounded information extraction: Acquiring missing feature values on demand. In *Advances in Knowledge Discovery and Data Mining*, volume 6118, pages 415–427. Springer

Berlin / Heidelberg, 2010. doi: http://dx.//doi.org/10.1007/978-3-642-13657-3_45.

[11] Evangelos Kanoulas, Ben Carterette, Mark M. Hall, Paul D. Clough, and Mark Sanderson. Overview of the TREC 2012 session track. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, 2012.

[12] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11): 1238–1274, 2013. doi: 10.1177/0278364913495721. URL http://dx.doi.org/10.1177/0278364913495721.

[13] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In Francis R. Bach and David M. Blei, editors, *ICML*, volume 37 of *JMLR Proceedings*, pages 1152–1161. JMLR.org, 2015. URL http://dblp.uni-trier.de/db/conf/icml/icml2015.html#KomiyamaHN15.

[14] Alexander Kotov and ChengXiang Zhai. Towards natural question guided search. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 541–550, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772746. URL http://doi.acm.org/10.1145/1772690.1772746.

[15] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *CoRR*, abs/1402.6028:1–48, 2014. URL http://arxiv.org/abs/1402.6028.

[16] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 817–824, 2007.

[17] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772758. URL `http://doi.acm.org/10.1145/1772690.1772758`.

[18] Jiyun Luo, Christopher Wing, Hui Yang, and Marti Hearst. The water filling model and the cube test: Multi-dimensional evaluation for professional search. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 709–714, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2523648. URL `http://doi.acm.org/10.1145/2505515.2523648`.

[19] Jiyun Luo, Sicong Zhang, and Hui Yang. Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 587–596, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609629. URL `http://doi.acm.org/10.1145/2600428.2609629`.

[20] Felipe Moraes, Rodrygo L. T. Santos, and Nivio Ziviani. UFMG at the TREC 2016 dynamic domain track. In Voorhees and Ellis [29]. URL `http://trec.`

nist.gov/pubs/trec25/papers/ufmg-DD.pdf.

[21] Hai Thanh Nguyen, Jérémie Mary, and Philippe Preux. Cold-start problems in recommendation systems via contextual-bandit algorithms. *CoRR*, abs/1405.7544, 2014. URL http://arxiv.org/abs/1405.7544.

[22] Paul Over. TREC-7 interactive track report. In *Proceedings of The Seventh Text REtrieval Conference, TREC 1998, Gaithersburg, Maryland, USA, November 9-11, 1998*, pages 33–39, 1998.

[23] Pannagadatta K. Shivaswamy and Thorsten Joachims. Multi-armed bandit problems with history. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 1046–1054. JMLR.org, 2012. URL http://jmlr.csail.mit.edu/proceedings/papers/v22/shivaswamy12.html.

[24] M. G. S. Sriyananda, Imtiaz Parvez, Ismail Güvenç, Mehdi Bennis, and Arif I. Sarwat. Multi-armed bandit for LTE-U and wifi coexistence in unlicensed bands. In *IEEE Wireless Communications and Networking Conference, WCNC 2016, Doha, Qatar, April 3-6, 2016*, pages 1–6. IEEE, 2016. ISBN 978-1-4673-9814-5. doi: 10.1109/WCNC.2016.7564899. URL http://dx.doi.org/10.1109/WCNC.2016.7564899.

[25] Alexander L. Strehl, John Langford, Lihong Li, and Sham Kakade. Learning from logged implicit exploration data. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *NIPS*, pages 2217–2225. Curran Associates, Inc., 2010. URL http://dblp.uni-trier.de/db/conf/nips/nips2010.html#StrehlLLK10.

[26] Balázs Szörényi, Róbert Busa-Fekete, Paul Weng, and Eyke Hüllermeier. Qualitative multi-armed bandits: A quantile-based approach. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1660–1668, 2015. URL `http://jmlr.org/proceedings/papers/v37/szorenyi15.html`.

[27] Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, and Tao Li. Personalized recommendation via parameter-free contextual bandits. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 323–332, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767707. URL `http://doi.acm.org.proxy.library.georgetown.edu/10.1145/2766462.2767707`.

[28] Adam Vogel and Dan Jurafsky. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[29] Ellen M. Voorhees and Angela Ellis, editors. *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, volume Special Publication 500-321, 2016. National Institute of Standards and Technology (NIST). URL `http://trec.nist.gov/pubs/trec25/trec2016.html`.

[30] Ryen W. White, Gheorghe Muresan, and Gary Marchionini. Report on acm sigir 2006 workshop on evaluating exploratory search systems. *SIGIR Forum*.

[31] Grace Hui Yang, John Frank, and Ian Soboroff. Overview of the trec 2015 dynamic domain track. In *TREC'15*, 2015.

[32] Hui Yang and Ian Soboroff. TREC 2016 dynamic domain overview. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016. URL `http://trec.nist.gov/pubs/trec25/papers/Overview-DD.pdf`.

[33] Zhe Yu, Yunjian Xu, and Lang Tong. Large scale charging of electric vehicles: A multi-armed bandit approach. In *53rd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2015, Allerton Park & Retreat Center, Monticello, IL, USA, September 29 - October 2, 2015*, pages 389–395. IEEE, 2015. ISBN 978-1-5090-1824-6. doi: 10.1109/ALLERTON.2015.7447030. URL `http://dx.doi.org/10.1109/ALLERTON.2015.7447030`.

[34] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1201–1208, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553527. URL `http://doi.acm.org/10.1145/1553374.1553527`.

[35] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. In *COLT*, 2009.

[36] Yisong Yue, Sue Ann Hong, and Carlos Guestrin. Hierarchical exploration for accelerating contextual bandits. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL `http://icml.cc/2012/papers/933.pdf`.

[37] Xiaoxue Zhao and Jun Wang. A theoretical analysis of two-stage recommendation for cold-start collaborative filtering. In *Proceedings of the 2015*

International Conference on The Theory of Information Retrieval, ICTIR '15, pages 71–80, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3833-2. doi: 10.1145/2808194.2809459. URL `http://doi.acm.org.proxy.library.georgetown.edu/10.1145/2808194.2809459`.

[38] Li Zhou. A survey on contextual multi-armed bandits. *CoRR*, abs/1508.03326, 2015. URL `http://arxiv.org/abs/1508.03326`.

[39] Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. *CoRR*, abs/1604.06743, 2016. URL `http://arxiv.org/abs/1604.06743`.

[40] Masrour Zoghi, Shimon Whiteson, Rémi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. *CoRR*, abs/1312.3393, 2013. URL `http://arxiv.org/abs/1312.3393`.