Cost-effective Identification of On-topic Search Queries using Multi-Armed Bandits

David E. Losada
david.losada@usc.es
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela
Spain

Matthias Herrmann
matthias.herrmann@stud.uni-regensburg.de
Chair for Information Science, University of Regensburg
Germany

David Elsweiler
David@Elsweiler.co.uk
Chair for Information Science, University of Regensburg
Germany

ABSTRACT

Identifying the topic of a search query is a challenging problem, for which a solution would be valuable in diverse situations. In this work, we formulate the problem as a ranking task where various rankers order queries in terms of likelihood of being related to a specific topic of interest. In doing so, an explore-exploit trade-off is established whereby exploiting effective rankers may result in more on-topic queries being discovered, but exploring weaker rankers might also offer value for the overall judgement process. We show empirically that multi-armed bandit algorithms can utilise signals from divergent query rankers, resulting in improved performance in extracting on-topic queries. In particular we find Bayesian non-stationary approaches to offer high utility. We explain why the results offer promise for several use-cases both within the field of information retrieval and for data-driven science, generally.

ACM Reference Format:
David E. Losada, Matthias Herrmann, and David Elsweiler. 2021. Cost-effective Identification of On-topic Search Queries using Multi-Armed Bandits. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3412841.3441944

1 INTRODUCTION AND MOTIVATION

The problem of algorithmically determining the type or topic of a search query is important for many reasons. Several of the key services provided by Web search engines, including ranking and advertising, rely on understanding the user's intention. Increasingly, the adverts shown [30] or the means of support provided by search engines, such as the provision of in-line results [8] or answering questions directly [5], are determined by making a prediction about the user's information need.

A further motivation for identifying queries related to a given topic of interest is the development of test collections for research purposes. Researchers in diverse fields require to identify materials from the web relating to specific topics or themes and often achieve this by sampling user search queries [9, 27, 33]. We are particularly interested in the identification of web pages which relate to food and weight-loss. As such, we take this problem as an exemplar in this paper, but one could imagine the techniques presented being used in diverse comparable contexts.

There are several aspects that make the estimation of the topic of a query a challenging research problem. First, queries are typically short and contain little information. Second, the queries in existing query samples or logs are in most cases extremely topically divergent and, as such, isolating specific topics can be akin to a finding a needle in a haystack. Third, queries are often ambiguous and noisy (e.g., contain misspelled words). Researchers, moreover, must contend with a lack of resources when estimating the topic of search queries. Few datasets are available for training and testing and, as with any human annotation problem, collecting labels is expensive. We argue that reinforcement learning and multi-armed bandits in particular can reduce the cost required to create resources and extract samples of queries associated with certain kinds of information needs.

Given a large sample of queries and a set of ranking methods that *nominate* candidate queries (i.e. queries that are potentially on-topic) from the query set, the process of judging queries from the candidate rankings can be naturally cast as a reinforcement learning problem. Initially, we know nothing about the relative quality of the rankers but, as judgements become available, we can dynamically adapt the process. Guided by multi-armed bandit algorithms, we can increasingly focus on the most effective rankers and, as a result, we can extract a sample of on-topic queries in a cost effective way. We demonstrate this empirically using a case study where the challenge is to identify queries associated to food and nutrition from within a large sample of web queries.

We structure the remainder of our paper by first reviewing appropriate literature in Section 2. We continue, in Section 3, to explain the methodology applied in detail, including the bandit allocation strategies tested and rankers used to provide a variety of signals, as well as how ground-truth judgements are created. Finally, we report and discuss our results in Sections 4 and 5, respectively.

2 RELATED WORK

First, we summarise literature on query classification and related problems in information retrieval (IR). Second, we review approaches for addressing the cost of human judgements in test collection generation, including pooling in IR, which is related to the problem we address. Third, we examine how multi-armed bandits have been

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/10.1145/3412841.3441944

applied to diverse problems. These reviews combine to motivate our work and justify the decisions taken with respect to the design of the presented experiments.

2.1 Query Classification

Pioneering researchers in IR recognised the potential for improving search technologies with text classification techniques [20, 37]. Query Classification –one specific text classification task in IR– is a challenging topic that has attracted researcher attention for many years. Some efforts have classified queries in terms of type, such as in terms of Broder's taxonomy of web searches [17]. Others have classified queries with respect to geographic locality as a means to establish local or global intent [15]. Others still have focused on predicting the topic of queries [4]. Indeed, some evaluation campaigns have focused on such query categorisation. For example, in 2005, the KDD Cup –the annual Data Mining and Knowledge Discovery competition– organised a challenge aiming to categorise 800,000 queries into 67 predefined categories [21]. This campaign recruited human editors to label a portion of the available queries, producing a ground truth composed of only 800 queries. This illustrates the difficulties in creating large samples of categorised queries. We argue that smartly prioritising the queries to be judged is an effective way to create larger samples for query classification problems.

The KDD Cup 2005 also revealed barriers to understanding the meaning and intention of search queries. As many queries are vague or ambiguous and most are very short, there is a need to gather extra information to augment the queries (e.g., by running the queries against web search engines and using the retrieved results to expand the queries). For example, the winning team in the KDD Cup 2005 [34, 35] employed an ensemble of search engines to produce intermediate representations of the queries which were then mapped to the target categories. The KDD Cup 2005 task was a multi-label task involving 67 target categories, while we focus here on a two-class categorisation task where the challenge is to extract queries related to a given topic of interest. However, the lessons learned in this campaign are useful to us when defining appropriate query representation approaches.

2.2 Test Collection Generation

One of the biggest challenges in the generation of test collections in any field is the cost of human judgements. In adhoc search these judgements involve estimating the relevance of documents for queries (topics). This is especially problematic in modern test collections, which try to simulate naturalistic search scenarios such as the web, and as such contain hundreds of millions of documents [7]. One solution from Information Retrieval has been to employ "pooling", which was advocated as a means of efficiently locating a sample of relevant documents within a large test collection. For each query, the output of diverse searches is merged to form a pool of documents, which is then assumed to contain all or nearly all relevant documents [38]. In pooled test collections, relevance assessments are only done for the documents that are in the pool. With a sufficient number of rankers and a reasonable *pool depth* (number of top documents extracted from each ranker), the manual judgements can be made at an affordable cost and the resulting test collection is solid and reusable [41].

Given that pooling is fundamental to modern IR evaluation, the concept has attracted considerable research attention in the decades since its initial use e.g. [6, 10, 11, 18, 40]. A number of studies have concentrated on efficient ways to scan pools, with the objective of extracting a sufficient number of relevant documents as quickly as possible. MoveToFront [11] and Moffat et al.'s methods [28] are classical prioritisation algorithms in this area. Losada and colleagues [24] have recently shown that effective document prioritisation together with smart stopping can reduce up to 95% of the assessment effort and still produce a reliable test collection.

Recent work, as described in the next section, has attempted to use a family of algorithms related to the multi-armed bandit problem to improve on such methods.

2.3 Applications for Multi-armed Bandits

Multi-armed bandits are one manifestation of the exploration versus exploitation problem, which is the search for a balance between exploring one's environment to find profitable actions while taking the empirically established best known action as often as possible [3]. Multi-armed bandit approaches have become fundamental in reinforcement learning [39]. Such approaches have recently been applied for various purposes in IR. Hofmann et al. [16] proposed bandit-based models to handle user interactions with a search engine as a means to improve online learning to rank. Here, the web search engine has to *exploit* existing knowledge regarding how to provide a good ranking, but it must also to *explore* by testing new variations of the current ranking algorithm. Taking a similar approach, Yue and Joachims [42] presented an online learning framework based on duelling bandits to compare retrieval algorithms. The approach is based on feedback gathered from users (ordinal judgements) and learns by observing interactions with interleaved results. Sloan and Wang [36] argued that document relevance changes over time and proposed a dynamic method that learns from clickthrough data and tries to optimise user satisfaction by employing multi-armed bandits and the Portfolio Theory, which handles diversity.

Radlinski et al. [31] used a multi-armed bandit approach that uses click-through data with the aim of balancing relevance and diversity in rankings. The authors' approach analysed user clickthrough behaviour and aimed to minimise user abandonment in interactive environments. Besides bandit-based approaches, other authors have proposed other methods to handle the exploration/exploitation tradeoff. This includes Karimzadehgan and Zhai [19], who balanced presenting search results with the highest immediate utility to a user against presenting search results with the best potential for collecting useful feedback information. The aim being to optimise the utility of relevance feedback over a session of interaction. In a spirit similar to ours, Losada and colleagues [22, 23] applied bandit algorithms to the pooling problem where decisions must be made as to which documents are judged by expensive human assessors. As multiple retrieval systems contribute to the pool, an exploration/exploitation trade-off arises: exploiting effective systems could find more relevant documents, but exploring weaker systems might also be valuable for the overall judgement process.

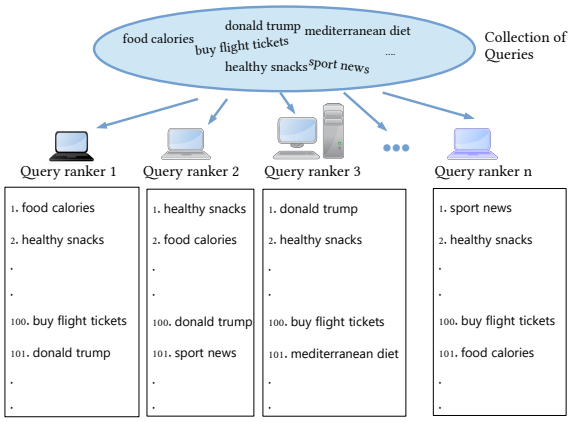

Figure 1: Query Rankers.

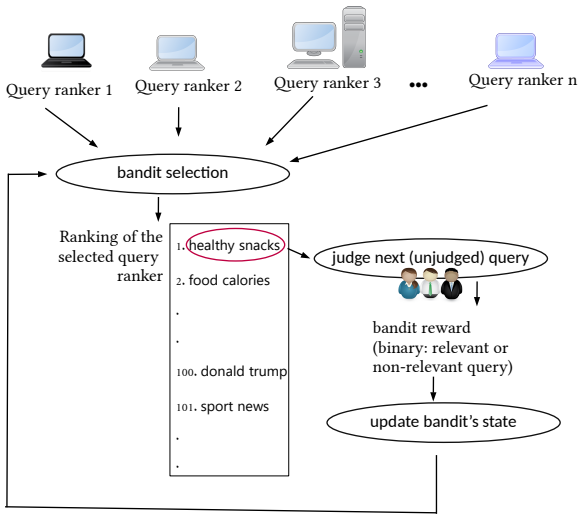

Figure 2: Bandit algorithm for iteratively selecting query rankers.

Losada et al. showed that simple multi-armed bandit models provide superior performance to all previous adjudication strategies. In this work, we test if a similar approach can be applied to query classification which, as we view it, is analogous to the pooling problem in IR.

3 METHOD

We start from a large collection of queries and we utilise a number of differing query rankers, where queries are ranked in order of how likely they are predicted to relate to a topic of interest (i.e. diet and nutrition). We evaluate competing reinforcement learning approaches to establish which approach can best combine signal from the provided rankings to generate evidence for topical relatedness of queries. Reinforcement learning is a dynamic process and, as such, we evaluate the approaches both in terms of how able they are to use evidence provided via different rankers and the amount of evidence required (i.e., how deep in the respective rankings should be explored).

Given a set of rankers ranking queries in decreasing order of estimated likelihood that they are related to diet and nutrition (see Fig. 1), we are interested in finding as many relevant queries as possible for the same amount of assessor effort. Initially, we have no knowledge about the relative quality of the rankers. As we extract queries from the provided rankings, we gain evidence on the quality of the rankers and the judging process can be oriented towards the most effective sources. At any given point, we can opt to further explore rankers that currently look suboptimal because these inferior rankers may at some point become good sources of relevant queries. When we refer to "Playing a machine" we mean selecting a ranker and examining the next query supplied by the ranker. Every ranker supplies queries in order of ranks (i.e., the top query precedes the second and so on). The query is judged and the outcome of the play is the binary relevance of the query (i.e. food related or not). Queries that have already been judged (i.e., have already supplied by another ranker) are simply skipped. Figure 2 illustrates this bandit selection process.

In our bandit experiments, the judgements are obtained from an oracle, which as shown in Section 3.3 can achieve extremely high precision on a comparable, but small dataset with expensive human judgements. This oracle is comparable to an expert human judge, not only because of its high performance, but because it is associated with financial cost. Annotating a full query sample would be beyond the means of most researchers. This is why bandits are required to prioritise the queries sent to the oracle for judgements. This is a comparable situation to pooling in the main adhoc search task.

In the following subsections, we complete the description of the methodology applied first, by detailing the various bandit allocation strategies tested, next by outlining the rankers used and lastly by explaining how the oracle judgements were derived.

3.1 Bandit Allocation Strategies

In the context of multi-armed bandits, an allocation strategy or policy is an algorithm that chooses which machine should be played next based on past plays and obtained rewards. Each policy captures distinct ideas on how to handle the tradeoff between exploration and exploitation. Regret is the expected loss due to the fact that the allocation strategy does not always play the best machine. The following paragraphs explain the main features of well-known allocation strategies, which we test empirically in this work.

3.1.1 Random. This is a naive allocation strategy that chooses the next machine to play at random. This serves as a baseline for comparison.

3.1.2 ϵ_n-greedy. Employing a greedy strategy would mean always playing the bandit with the highest average reward[1]. In doing so, this strategy maximises immediate rewards by allocating no time to exploring seemingly inferior actions. A greedy method performs worse in the long-term as it often gets stuck performing suboptimal actions, which showed promise early on. A simple variation is to apply a greedy strategy most of the time while occasionally selecting an action at random. A simple algorithm that implements this idea is ϵ-greedy [39]. At each step, ϵ-greedy plays the machine with the highest mean reward with probability $1-\epsilon$, and a randomly chosen machine with probability ϵ. ϵ-greedy eventually performs better than a purely greedy algorithm because it continues to explore, improving the chances of identifying optimal actions.

Rather than setting a constant probability of exploration, it is usually good to make that ϵ decreases as our estimates become more accurate. To meet this aim, ϵ_n-greedy lets ϵ go to zero with a certain rate:

$$\epsilon_n \quad = \quad min(1, \frac{c \cdot K}{d^2 \cdot n}), n = 1, 2, \ldots \tag{1}$$

where n is the round number, $c > 0$ is a parameter, K is the number of machines, and d is usually set to the difference (in mean reward) between the best choice and the second best[2].

3.1.3 Upper Confidence Bound (UCB). UCB policies associate a so-called upper confidence index to each machine. The machine to be played at round n is the machine with the largest empirical mean of obtained rewards. While it would be desirable to simply sample from this seemingly superior machine, we need to ensure that the other machines have been sufficiently sampled such that we can be reasonably confident that they are indeed inferior. One way of achieving this is to compare the upper confidence bound for the mean of an apparently inferior approach to the mean of the leader. The index of UCB1 policy computes the current mean reward and adds a term related to the size of the one-sided confidence interval for the mean reward. UCB1-Tuned [3] is a variant of UCB1 that accounts for the variance in performance of each machine and has been shown to be empirically superior to UCB1:

Algorithm 1: UCB1-Tuned

Play each machine once;
Loop
  Play machine j that maximises the following estimate:
  $\mu_j + \sqrt{\frac{\ln n}{n_j} \cdot min(1/4, \sigma_j^2 + \sqrt{\frac{2 \ln n}{n_j}})}$

where μ_j and σ_j^2 are the mean and variance of the rewards obtained from machine j so far, n_j is the number of times machine j has been played, and n is the overall number of plays.

A feature of the algorithm is that the quantity added to the sample average is steadily reduced as the machine is played, and uncertainty about its reward probability is reduced. As a result, by choosing the machine with the highest optimistic estimate, UCB1-Tuned smoothly shifts from exploration to exploitation.

[1]Initially, all averages are set to 0.5.
[2]In document pooling experiments [22] it was shown that effectiveness was insensitive to d and moderately sensitive to c (all $c \in (0, 0.1]$ yielded similar results). We therefore set d to 0.1 and c to 0.01.

3.1.4 Bayesian Bandits. The methods described so far all take a frequentist approach, where mean rewards are considered as unknown deterministic quantities and the goal of the algorithm is to achieve the best parameter-dependent effectiveness. Bayesian approaches, in contrast, apply quantitative weighting of evidence supporting alternative hypotheses.

Each machine is characterised by a parameter which reflects a prior distribution. This parameter represents the probability of winning (in our case the probability of supplying a relevant query). The Bayesian process begins by assuming complete ignorance of these probabilities and, therefore, applying a uniform prior, $\mathcal{U}(0, 1)$, for each machine. We select our next machine from these distributions and observe the result of playing the machine. With binary rewards, the result is Bernoulli or, equivalently, Binomial with a single trial. This binary reward is used to revise our belief about the probability of the specific machine. The initial priors are $Beta(1, 1)$ –Beta handles the uniform distribution as a particular case– and Beta is the conjugate prior distribution for Binomial. Given a prior distribution $Beta(\alpha, \beta)$ and a binary reward R, the posterior distribution is also Beta: $Beta(\alpha + R, \beta + 1 - R)$. Bayesian inference, therefore, provides a natural framework allowing us to formally handle uncertainty about the probabilities of winning.

Bayesian Learning Automaton (BLA) [14] (Algorithm 2) follows this approach and randomly samples from the posterior distributions to choose the next machine to play. BLA is parameter-free and typically performs significantly better than both UCB and ϵ_n-greedy [14]. A further Bayesian solution we implement selects the next machine by taking the maximum expectation of the posterior distributions. This exploitation method will be referred to as *MM* (MaxMean)[3]: $next\_machine \leftarrow \arg\max_m \alpha_m/(\alpha_m + \beta_m)$.

Algorithm 2: Bayesian Learning Automaton

foreach $m \in machines$ **do**
  $\alpha_m \leftarrow 1, \beta_m \leftarrow 1$;
Loop
  foreach $m \in machines$ **do**
    Draw a sample x_m from $Beta(\alpha_m, \beta_m)$;
  $next\_machine \leftarrow \arg\max_m x_m$;
  Play $next\_machine$ and get $R_{next\_machine}$;
  $\alpha_{next\_machine} \leftarrow \alpha_{next\_machine} + R_{next\_machine}$;
  $\beta_{next\_machine} \leftarrow \beta_{next\_machine} + 1 - R_{next\_machine}$;

Rather than simply updating the distribution of the played machine (i.e. the ranker that supplied the last query judged), we opt to update the Beta distribution of all rankers that retrieved the same query. In doing so we allow evidence about relevance to affect other rankers[4].

3.1.5 Non-stationary variants. In the Bayesian models described above, the unknown probability of bandit success does not change, and all rewards –recent or old– are treated equally. Non-stationary solutions, on the other hand, account for the possibility that these

[3]The expectation of a distribution $Beta(\alpha, \beta)$ is $\alpha/(\alpha + \beta)$.
[4]With *MM*, this update sometimes leads to several machines having the maximum mean. Ties are resolved by choosing the played machine.

distributions may change and, as a consequence, allow to weight recent rewards more heavily than long-past ones [39]. This makes sense in our case as the quality of the rankers will reduce as we move down in the rankings –we would not expect even a good ranker to constantly supply relevant queries to the bottom of the ranking. As queries are examined the probabilities of relevance of the rankers will change and so will the relative performance of different rankers. A ranker that was initially strong might be weak at lower rank positions when compared to the competing rankers. Stationary bandit approaches run the risk of concentrating too much on rankers with old wins, leading to suboptimal solutions. One popular means of tracking non-stationary problems is to incorporate a parameter that ensures accumulated rewards are computed as a weighted average of the past rewards and the last reward. This idea can be easily incorporated into the Bayesian models BLA and MM. At any given point, the parameters of the posterior distribution of a given ranker r are:

$$\alpha_r = 1 + jrel_r \quad (2)$$

$$\beta_r = 1 + jret_r - jrel_r \quad (3)$$

where $jrel_r$ is the number of judged queries that are relevant and were retrieved by r, and $jret_r$ is the number of judged queries that were retrieved by r. Updating $jrel_r$ and $jret_r$ can be governed by a rate parameter that motivates the method to learn changing environments [13]. Given the binary relevance of the last query judged, rel_q, the parameters of the rankers retrieving this query are updated as:

$$jrel_r \leftarrow rate \cdot jrel_r + rel_q \quad (4)$$

$$jret_r \leftarrow rate \cdot jret_r + 1 \quad (5)$$

If $rate = 1$ this is the standard approach, where all outcomes count the same. If $rate > 1$ the method applies more weight to early relevant queries. Conversely, if $rate < 1$ the method applies more weight to the last relevant query found. Here, we test the most stringent variant, $rate = 0$ (with such a setting, only the last judgement counts). The non-stationary approach with this setting was shown to be highly effective in nominating documents to be judged for a standard adhoc search task [22]. Updating the parameters of the posterior distributions in this way leads to new Bayesian methods, which we refer to as non-stationary Bayesian solutions ($rate = 0$) (BLA-NS and MM-NS). BLA-NS, once the distributions are updated, selects the next ranker by sampling from the posterior distribution. MM-NS simply selects the posterior distribution with the largest mean. Observe that setting $rate$ to 0 preserves the formality of the model: rewards are still Bernoulli and priors/posteriors are still Beta. Setting $rate = 0$ can be seen as a re-initialisation of the machine's counts immediately before to playing the machine.

3.2 Rankers Tested

We implemented 15 query rankers to provide the basis for our bandit experiments. These rankers reflect a broad spectrum of retrieval mechanisms with divergent quality. This gives the bandit algorithms the opportunity to identify and focus on the most effective approaches. The rankers consist of variants of two main types.

The first class of methods follow standard IR techniques and search for relevant queries using a state-of-the-art retrieval model, BM25 [32]. The BM25 ranker was implemented using the Whoosh Python library with the default configuration[5]. Search was done with three different queries: a base query (*BQ*), which contains only the words "food" and "diet", and two expanded queries. These two expanded queries aim to mitigate the effect of poor overlapping between the base query and the target queries. We expanded the base query with i) the N most similar terms to "food" and ii) the N most similar terms to "diet". The two expanded queries, BQ+15 and BQ+30, correspond to setting N to 15 and 30, respectively. Words similar to the base words were obtained using Word2Vec embeddings [25, 26]. Word embeddings are high-quality distributed vector representations of words that capture semantic relationships. More specifically, we employed prebuilt Word2Vec models generated from English Wikipedia[6] (non-stemmed words represented as 1000-dimensional vectors) and we utilised Gensim's Word2vec library[7] to load the vectors and compute similarities.

To further increase the recall of on-topic queries, we also included some variants that perform pseudo-relevance feedback [12] from the output of the BM25 search. To meet this aim, we performed query expansion following the Bo1 model from the Divergence from Randomness (DFR) framework [2]. This is a robust expansion approach that obtains new search terms from the top results of the initial ranking[8]. In our experiments, the new terms were sourced from the top 100 results (top 100 queries identified by the initial BM25 search) and we included up to 20 new terms. Overall, this led to 12 different query rankers (3 possible initial queries * 4 possible feedback settings: 0 –i.e., no feedback–, 1, 2 and 3 rounds of feedback). These 12 query rankers will be referred to as BM25(<Q>, <N> PRF), where <Q> is BQ, BQ+15 or BQ+30 and <N> equals 0, 1, 2 or 3.

A second class of query rankers implemented a vectorial representation of the base query and each target query. To meet this aim, we employed Sent2Vect [29], an unsupervised method to learn representations of sentences. Word embeddings allow to represent words as N-dimensional vectors and, depending on the training method utilised, semantically similar (or contextually-related) words are assigned vectors that are close to each other in the projected space. Sent2Vect obtains vectorial representations of sentences by aggregating word and n-gram embeddings and simultaneously training the composition and the embedding vectors. We employed a pre-trained Sent2vec Bigram model[9] produced from the English Wikipedia. Wikipedia articles cover a wide range of topics and therefore have an extensive vocabulary. Using this resource, we represented the base query and each target query as centroid vectors (averages of the Sent2Vec representations of the constituting terms) and ranked the target queries by decreasing cosine similarity. We implemented 3 variants of this embedding-based approach to extract relevant queries: i) a method that only includes the words "food" and "diet" into the base query, ii) a method that includes "food", "diet" and all their hyponyms from WordNet (52 words) into the base query, and iii) a method that includes "food", "diet" and

[5] See https://whoosh.readthedocs.io/en/latest/intro.html. We utilised the Okapi BM25F ranking function with a single field and default parameters: $b = 0.75$ and $K1 = 1.5$.
[6] https://github.com/idio/wiki2vec
[7] https://radimrehurek.com/gensim/models/word2vec.html
[8] We used the default implementation of Bo1 available from Whoosh (https://whoosh.readthedocs.io/en/latest/keywords.html).
[9] sent2vec_wiki_bigrams, 16GB (composed of vectors with 700 dimensions). Available at https://github.com/epfml/sent2vec.

more than 130 additional words related to food and nutrition and collected as part of an open source project[10]. These three methods will be referred to as Sent2Vec(BQ), Sent2Vec(BQ+WordNet), and Sent2Vec(BQ+WordList), respectively.

3.3 Evaluating the Oracle

To evaluate the relevance of the ranked queries we would need human judgements. But human judgements are expensive and, thus, a manual labelling method would not scale. We therefore designed and evaluated an "oracle-classifier", which expands the queries and runs them against an external classification service. The oracle was evaluated against a set of manually labelled queries, obtained from KDD 2005. The details of this validation are provided in this subsection.

As argued in [21], query classifiers face a great challenge. Queries are typically short, noisy and result from subjective user intents. A common approach to mitigate the lack of information consists of augmenting queries by gathering extra information from web resources. To meet this aim, we ran first each query against a state-of-the-art search engine (Google Custom Search API[11]) and augmented the query with the title and snippet of the top k results. Next, the augmented query was sent to a external classification service, the Google Cloud Natural Language API[12]. This external service includes a content-based classifier that assigns labels from a large set of categories[13]. Since we are interested in food or diet-related queries, we considered that on-topic queries are those assigned with the "/Food & Drink" and "/Health/Nutrition" categories. This external service has an associated cost and, thus, a massive use is prohibitive. This is why scientists interested in building query-related resources need to prioritise the queries (e.g., using multi-armed bandits) that are labelled by humans or sent to this type of services.

In order to evaluate this oracle-based approach, we obtained food and diet related queries from the 800 labelled queries available from the KDD cup 2005 [21]. The organisers of this campaign recruited three human editors to label a sample of 800 queries manually using 67 categories. Since we focus on two-class classification, we divided the queries into food/diet related and food/diet unrelated[14]. In total 43 of the 800 queries were considered to be related to food or diet. We augmented all queries with search results snippets (top k results, with k ranging from 1 to 10) and sent them to the external classification service. The optimal performance was achieved using $k = 5$ (acc=0.98, P=0.97, F1=0.83). This suggests that this oracle-based approach effectively identifies on-topic queries and we therefore adopted this method as a proxy of human judgements.

3.4 Bandit Experiments

We ran the 15 query rankers against a large sample of 59, 673 queries, obtained from the TREC million query tracks[15]. Next, we started a multi-armed bandit process where the rankers play the role of arms or machines and where the allocation policy selects a given ranker (arm), from which to extract the next (unjudged) query from the top of the corresponding ranking. The extracted query is sent to the oracle classifier and the output is used to update the state of the bandits (see section 3.1 for further details about the multi-armed bandit strategies tested). To put the results into perspective, we also experimented with a rank-based method (RANK) where queries are simply selected in decreasing order of rank (top 1 queries go first and, next, top 2 queries, and so forth). The process was run until depth $k = 700$. The pool of judged queries (union of the top 700 ranked queries) contains 2, 160 unique queries (of which 913 were relevant according to the oracle).

Table 1: Number of relevant queries found at different number of judgements performed. For each judgement level, the highest number of relevant queries found is bolded.

	Number of Judgements					
Method	100	500	1000	1500	2000	All
RANK	88	344	589	756	874	913
RANDOM	89	337	534	667	827	913
ϵ-Greedy	**91**	349	530	674	825	913
UCB	86	339	533	666	824	913
BLA	90	333	500	582	820	913
MM	**91**	344	513	617	882	913
BLA_NS	89	311	479	596	820	913
MM_NS	90	**384**	**637**	**789**	**909**	**913**

4 RESULTS

Table 1 reports the number of relevant queries identified by each bandit allocation strategy at varying judgement levels. At the end of the process, all strategies identify the same number of relevant queries (all pooled queries judged). However, some strategies are much quicker than others to identify on-topic queries. Identifying queries earlier leads to a reduction in required effort in judgement as we can simply stop the assessment process when a sufficient number of search queries are found.

The following conclusions can be drawn from these experiments:

- Unsurprisingly, randomly selecting the next ranker from which to extract a query is the worst performing allocation strategy. Applying this naive method does not exploit evidence gained with respect to which rankers are good sources of relevant queries and, thus, tends to identify relevant queries slowly compared to the other, more principled approaches.
  The number of on-topic queries found by randomly selecting rankers is not very low, particularly after judging the first 100 queries. This is also the case with all of the allocation strategies tested. Note that rankers –regardless of how their type– are designed to order queries by the likelihood that they related to food or diet. Thus, despite rankers being selected at random, the queries themselves are drawn in order leading to reasonable performance, reflecting the overall performance of the rankers.

[10] https://github.com/imsky/wordlists
[11] https://developers.google.com/custom-search
[12] https://cloud.google.com/natural-language/
[13] See https://cloud.google.com/natural-language/docs/categories
[14] We included into the food/diet set those queries assigned to the KDD category "Living/Health&Fitness" by at least one of the judges.
[15] We downloaded all queries used in the TREC 2007, 2008 and 2009 Million Query Tracks (https://trec.nist.gov/data/million.query09.html) and removed duplicates.

- The 15 query rankers tested are reasonably effective at finding on-topic queries. The top 100 positions, in particular, contain a large proportion of relevant queries and 42% of the pooled queries are relevant.
- The best performing strategy overall was MM-NS. This is a non-stationary Bayesian method that quickly reacts to the presence of non-relevant queries in the rankings. Other methods, such as UCB or ϵ_n-greedy, are slower to abandon rankers that offered good performance initially but deteriorate with rank. The superiority of MM-NS over alternative bandits (and over the RANK baseline) aligns well with the results found in pooling document judgments [22, 23]. The increasing evidence for the utility of such Bayesian Bandits justified their adoption in TREC 2017 when creating the relevance judgments of the Common Core Track [1]. Our results add to this evidence and support the potential of this reinforcement learning methods to create labelled judgments in a cost effective way.

Table 2 reports the first 50 queries judged following the MM-NS approach. These show that the multi-armed bandit method does a good job at extracting on-topic queries with a high percentage of food or diet related queries being present. Note that the oracle classifier is not perfect (particularly in terms of recall, as reported in section 3.3) and, thus, a few on-topic queries (e.g., "diverticulosis diet") are wrongly classified as off-topic. On the other hand, queries that are clearly off-topic (e.g., "electromagnetic spectrum") are correctly labelled as non-relevant by the oracle. Although we miss some queries that are potentially relevant, the quality of the set of queries estimated as relevant is very high. Indeed, for most use cases, the precision of the resulting query set is the most crucial aspect. For example, in online advertisement, we typically do not want to show advertisements that are totally unrelated to the user's needs and, thus, a precise identification of the target queries is crucial (otherwise the advertisements might be perceived as confusing or annoying by the web users).

Figures 3 and 4 provide final insights into how different bandit allocation strategies function in practice. Figure 3 helps to illustrate the exploitation vs exploration behaviour of the diverse bandit algorithms by depicting the percentage of times they "jump" to explore new possibilities (i.e., the next machine is different to the last played machine). The MM algorithms show a clear exploitation-oriented behaviour and they tend to explore less often at the beginning of the process. At the initial stages, queries are extracted from high positions of the ranked lists, where relevant queries abound. The MM algorithms select the ranker with the highest average and, thus, at the beginning of the process, they tend to stay on the winners (no reason to leave a good supplier of queries). Note also that MM_NS jumps more than MM because it is non-stationary (MM averages the full history of previous rewards –relevant/non-relevant queries extracted– while MM_NS has a highly stringent notion of history where only the last extraction counts). On the other hand, the other bandits (UCB, ϵ-Greedy, BLAs and random) explore many more options and do so continuously until very late in the process. It is only after rank 1800 that exploring reduces.

Figure 4 shows which rankers provided input to the most effective algorithm (MM_NS) and how this changed as the algorithm progressed deeper in the process. Initially, traditional IR approaches (several BM25 variants) were deemed to be the best source of relevant queries (at high ranking positions they are good suppliers of relevant queries). Indeed, in the first 100 plays the Sent2vec rankers were never chosen at all. However, as the process continues, the Sent2vec variants become important and eventually dominate. This is a natural consequence of the respective characteristics of these methods. The BM25 variants are essentially term matching models and, even with expansion and pseudo-feedback, their ability to retrieve lots of relevant queries is limited. The Sent2Vec variants, instead, have a recall-oriented nature and, by employing embeddings, they can identify target queries that have no overlapping at all with the seed terms. In this way, when the top positions of the BM25 rankers become exhausted of relevant queries the process concentrates more on variants such as Sent2Vec(BQ+WordList) which, at the end of the process, was the most frequently selected ranker.

Table 2: First 50 queries extracted by MM-NS. The first column shows the search queries extracted and the second column reports the relevance value according to the oracle classifier.

Search Query	Relevance Value
daily nutrition calories and fat	1
healthy snacks	1
dietary supplement fact sheet vitamin c	1
vegetable calories	1
vitamins and supplements	1
vitamins supplements	1
vitamin e supplement	1
healthy cooking	1
healthy nutrition	1
nutrients in food	1
list dietary supplement	1
risk of dietary supplement	1
healthy meal recipes	1
why are vegetables healthy	1
food calories	1
sat fat trans fat and fat from calories	1
eating food	1
how much fat and calories do vegetables and fruits have	1
healthy eating plan	1
learning to eat healthy	1
diet supplements with ephedra	1
nutrition and eating right	1
lact-enz dietary supplement	0
assay of magnesium	1
93150 pill	0
diet	1
dieting	1
food	1
lupus diet	1
diet exchanges	1
brat diet	1
watermelon diet	1
diverticulosis diet	0
diabetic diet exchange list	1
electromagnetic spectrum	0
low fat low sodium heart healthy foods	1
list of low fat foods	1
heart healthy diets	1
heart healthy meal recipes	1
calories in food list	1
low calorie cook books	1
low protein dog food	0
diabetic exchange diet	1
pet food recall	0
mediterranean diet	1
hepatitis a diet	0
diabetic diet vs dash diet	1
himalayan diet	1
vlcd diet	1
atheletes diet	1

5 DISCUSSION & CONCLUSION

In this work we have described experiments, which show that multi-armed bandit algorithms can utilise signals from several divergent query rankers resulting in improved performance in extracting on-topic search queries. In this section we discuss what these results mean in the context of past and future work in different fields.

Firstly, our experimental findings contribute to the multi-armed bandit literature, generally. As shown in Section 2.3, even within the research domain of information retrieval, a multitude of use-cases have been found for which multi-armed bandit algorithms offer utility. We add one further use-case –on topic query identification– to this literature.

Our results align strongly with those published by Losada and colleagues for the pooling problem [22] which, we argue, exhibits a great deal of similarity to the query classification problem as we set it up. We found the non-stationary Bayesian allocation strategy was the dominant approach beyond a judgement level of rank 100. This was also the case in the experiments by Losada et al.

We investigated the query classification problem in a slightly different form to how has been studied previously in the IR literature (i.e., as a multi-class classification problem). In our work, we instead employ a binary classification setup where a single query topic needs to be isolated. This makes more sense for our needs (identifying queries related to food and diet) and –as we argue– for other purposes, too. This kind of task would be important, for example, when systems are required to provide specific support for queries relating to a particular topic. One hypothetical example would be applying fact-checking to results relating to political queries. Another could be red-flagging or sensoring queries relating to child pornography or other sensitive or illegal domains. As mentioned in the motivation for this paper, a binary topic-based query classification setup is also a common use-case in fields such as public health, where many scholars have the aim to study or evaluate commonly accessed web-pages relating to a particular subject [9, 27, 33]. Sampling from naturalistic query logs –which our methods facilitate– is likely to provide a much more representative pool of queries than the approach commonly applied where it is typical to gather sample queries from a small convenience sample of users. Our work, therefore, opens up the potential for researchers to better identify representative web pages for their research.

Although our experiments and the discussion of the results have focused on IR related problems, it is important to recognise that this approach could be exploited in diverse situations in science, generally. Evaluation is crucial to scientific progress. In many disciplines, particularly those that employ data driven approaches, creating a gold standard set of judgements is very often a major bottleneck when building a test collection or benchmark for evaluation. Gold standard or ground truth annotations are typically produced by humans and are, therefore, expensive and time consuming to collect.

In terms of future work we plan to apply the bandit methods discussed here to query classification as a multi-class problem to see if similar benefit can be obtained. A second line of planned future work is to apply pooling methods in Machine Learning and Data Mining. More specifically, we will study pool-based ways to prioritise unlabelled items in different contexts. We expect –and our results here endorse our thoughts– prioritisation strategies to be helpful in building robust and effective ground-truth data.

ACKNOWLEDGMENTS

This work was supported by Ministerio de Ciencia, Innovación y Universidades, which funded a research stay of the first author at the University of Regensburg (ref. PRX18/00096, "Salvador de Madariaga" research stays programme). The first author also acknowledges the support provided by FEDER/ Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/ Project (ref. RTI2018-093336-B-C21). He also thanks the financial support supplied by the Consellería de Educación, Universidade e Formación Profesional (accreditation 2019-2022 ED431G-2019/04, ED431C 2018/29) and the European Regional Development Fund, which acknowledges the CiTIUS-Research Center in Intelligent Technologies of the University of Santiago de Compostela as a Research Center of the Galician University System.

REFERENCES

[1] J. Allan, D. Harman, E. Kanoulas, D. Li, C. Van Gysel, and E. Voorhees. 2017. TREC 2017 Common Core Track Overview. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*. https://trec.nist.gov/pubs/trec26/papers/Overview-CC.pdf

[2] G. Amati. 2003. *Probability models for information retrieval based on divergence from randomness.* Ph.D. Dissertation. University of Glasgow.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. 2002. Finite-time analysis of the multi-armed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[4] S. Beitzel, E. Jensen, O. Frieder, D. Lewis, A. Chowdhury, and A. Kolcz. 2005. Improving automatic query classification via semi-supervised learning. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 8–pp.

[5] M. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. 2012. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 237–246.

[6] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. 2007. Bias and the Limits of Pooling for Large Collections. *Inf. Retr.* 10, 6 (Dec. 2007), 491–508. https://doi.org/10.1007/s10791-007-9032-x

[7] J. Callan. 2012. The Lemur project and its CLUEWEB12 dataset. In *Invited talk at the SIGIR 2012 Workshop on Open-Source Information Retrieval.*

[8] L. Chilton and J. Teevan. 2011. Addressing people's information needs directly in a web search result page. In *Proceedings of the 20th international conference on World wide web*. ACM, 27–36.

[9] M. Chung, R. Oden, B. Joyner, A. Sims, and R. Moon. 2012. Safe infant sleep recommendations on the Internet: let's Google it. *The Journal of pediatrics* 161, 6 (2012), 1080–1084.

[10] G. Cormack and T. Lynam. 2007. Power and Bias of Subset Pooling Strategies. In *Proc. of the 30th Annual Int. Conf. on Research and Development in Information Retrieval* (Amsterdam, The Netherlands). ACM, USA, 837–838. https://doi.org/10.1145/1277741.1277934

[11] G. Cormack, C. Palmer, and C. Clarke. 1998. Efficient Construction of Large Test Collections. In *Proc. of the 21st Annual Int. Conf. on Research and Development in Information Retrieval* (Melbourne, Australia). ACM, USA, 282–289. https://doi.org/10.1145/290941.291009

[12] W.B. Croft and D. Harper. 1979. Using Probabilistic Models of Document Retrieval without Relevance Information. *Journal of Documentation* 35, 4 (1979), 285–295.

[13] C. Davidson-Pilon. 2015. *Probabilistic Programming & Bayesian Methods for Hackers.* Addison-Wesley Data & Analytics Series. http://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/

[14] O. Granmo. 2008. A Bayesian Learning Automaton for Solving Two-Armed Bernoulli Bandit Problems. In *Proc. of Seventh Int. Conference on Machine Learning and Applications (ICMLA '08)*. 23–30. https://doi.org/10.1109/ICMLA.2008.67

[15] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. 2003. Categorizing web queries according to geographical locality. In *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 325–333.

[16] K. Hofmann, S. Whiteson, and M. de Rijke. 2011. Contextual Bandits for Information Retrieval. In *NIPS 2011 Workshop on Bayesian Optimization, Experimental Design, and Bandits*. Granada.

[17] B. Jansen, D. Booth, and A. Spink. 2007. Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 1149–1150.

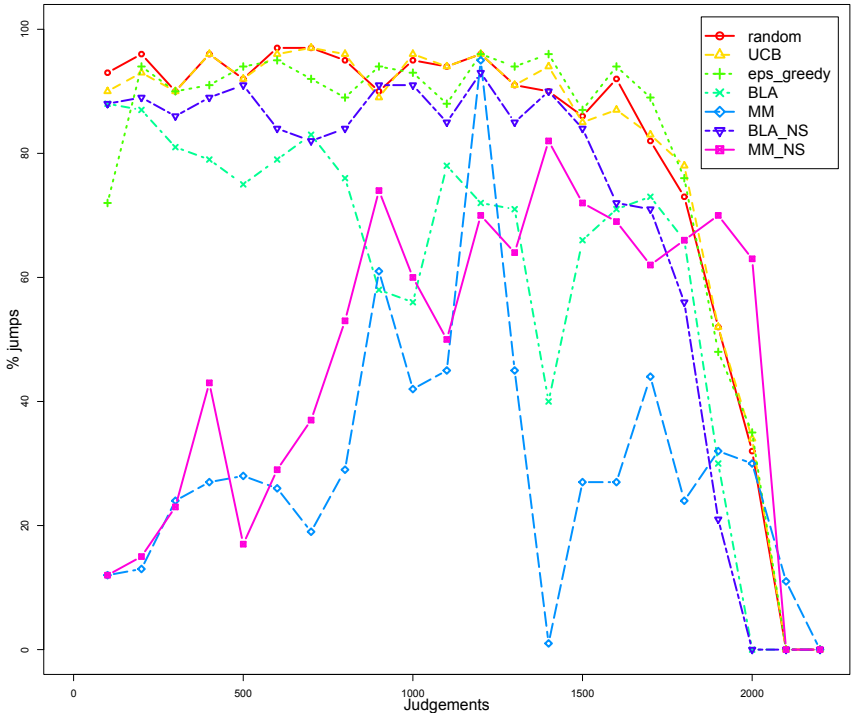

Figure 3: Percentage of Jumps by different Multi-armed Bandit methods.

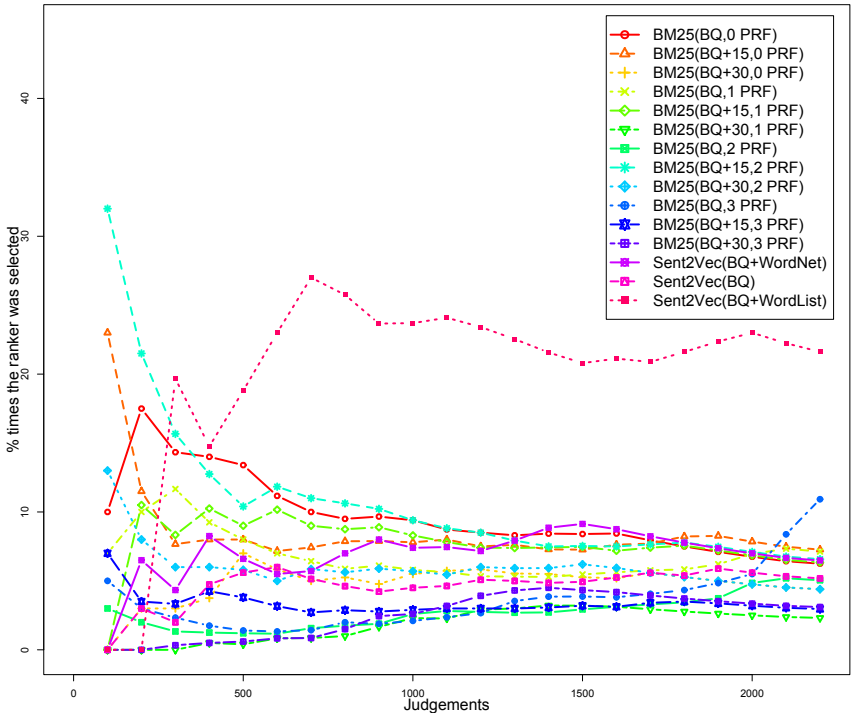

Figure 4: Rankers selected by the MM_NS method (in percentage).

[18] G. Jayasinghe, W. Webber, M. Sanderson, and J. Culpepper. 2014. Extending Test Collection Pools Without Manual Runs. In *Proc. of the 37th Int. ACM SIGIR Conference on Research and Development in Information Retrieval* (Gold Coast,

Queensland, Australia) *(SIGIR '14)*. ACM, New York, NY, USA, 915–918. https://doi.org/10.1145/2600428.2609473

[19] M. Karimzadehgan and C. Zhai. 2013. A learning approach to optimizing exploration-exploitation tradeoff in relevance feedback. *Inf. Retr.* 16, 3 (2013), 307–330. http://dblp.uni-trier.de/db/journals/ir/ir16.html#KarimzadehganZ13

[20] D. Lewis and W. Gale. 1994. A sequential algorithm for training text classifiers. In *Proc. of the 17th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*. 3–12.

[21] Y. Li, Z. Zheng, and H. Dai. 2005. KDD CUP-2005 report: facing a great challenge. *SIGKDD Explorations* 7 (01 2005), 91–99.

[22] D.E. Losada, J. Parapar, and A. Barreiro. 2016. Feeling lucky?: multi-armed bandits for ordering judgements in pooling-based evaluation. In *Proceedings of the 31st annual ACM symposium on applied computing*. ACM, 1027–1034.

[23] D.E. Losada, J. Parapar, and A. Barreiro. 2017. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing Management* 53, 5 (2017), 1005 – 1025. https://doi.org/10.1016/j.ipm.2017.04.005

[24] D.E. Losada, J. Parapar, and A. Barreiro. 2019. When to stop making relevance judgments? A study of stopping methods for building information retrieval test collections. *Journal of the Association for Information Science and Technology* 70, 1 (2019), 49–60. https://doi.org/10.1002/asi.24077 arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24077

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781

[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[27] F. Modave, N. Shokar, E. Peñaranda, and N. Nguyen. 2014. Analysis of the accuracy of weight loss information search engine results on the internet. *American journal of public health* 104, 10 (2014), 1971–1978.

[28] A. Moffat, W. Webber, and J. Zobel. 2007. Strategic System Comparisons via Targeted Relevance Judgments. In *Proc. 30th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands). ACM, NY, USA, 375–382. https://doi.org/10.1145/1277741.1277806

[29] M. Pagliardini, P. Gupta, and M. Jaggi. 2018. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 528–540. https://doi.org/10.18653/v1/N18-1049

[30] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. 2008. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 403–410.

[31] F. Radlinski, R. Kleinberg, and T. Joachims. 2008. Learning Diverse Rankings with Multi-armed Bandits. In *Proc. of the 25th Int. Conference on Machine Learning* (Helsinki, Finland) *(ICML '08)*. ACM, New York, NY, USA, 784–791. https://doi.org/10.1145/1390156.1390255

[32] S. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[33] P. Scullard, C. Peacock, and P. Davies. 2010. Googling children's health: reliability of medical advice on the internet. *Archives of disease in childhood* 95, 8 (2010), 580–582.

[34] D. Shen, R. Pan, J-T. Sun, J.J. Pan, K. Wu, J. Yin, and Q. Yang. 2006. Query Enrichment for Web-query Classification. *ACM Trans. Inf. Syst.* 24, 3 (July 2006), 320–352. https://doi.org/10.1145/1165774.1165776

[35] D. Shen, J-T. Sun, Q. Yang, and Z. Chen. 2006. Building Bridges for Web Query Classification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) *(SIGIR '06)*. ACM, New York, NY, USA, 131–138. https://doi.org/10.1145/1148170.1148196

[36] M. Sloan and J. Wang. 2012. Dynamical Information Retrieval Modelling: A Portfolio-armed Bandit Machine Approach. In *Proc. of the 21st Int. Conf. Companion on World Wide Web* (Lyon, France). ACM, USA, 603–604. https://doi.org/10.1145/2187980.2188148

[37] K. Sparck-Jones. 1971. *Automatic keyword classification for information retrieval*. Butterworths.

[38] K. Sparck-Jones and C.J. Van Rijsbergen. 1975. Report on the Need for and Provision of an Ideal Information Retrieval Test Collection. *Cambridge : University Computer Laboratory* (1975).

[39] R. Sutton and A. Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[40] E. Voorhees. 2002. The Philosophy of Information Retrieval Evaluation. In *Proc. of 2nd Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*. Berlin, Heidelberg, 355–370.

[41] E. Voorhees and D. Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press.

[42] Y. Yue and T. Joachims. 2009. Interactively Optimizing Information Retrieval Systems As a Dueling Bandits Problem. In *Proc. of the 26th Annual Int. Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. ACM, NY, USA, 1201–1208. https://doi.org/10.1145/1553374.1553527