The Answer is at your Fingertips: Improving Passage Retrieval for Web Question Answering with Search Behavior Data

Mikhail Ageev*
Moscow State University
mageev@yandex.ru

Dmitry Lagun
Emory University
dlagun@emory.edu

Eugene Agichtein
Emory University
eugene@mathcs.emory.edu

Abstract

Passage retrieval is a crucial first step of automatic Question Answering (QA). While existing passage retrieval algorithms are effective at selecting document passages most similar to the question, or those that contain the expected answer types, they do not take into account which parts of the document the searchers actually found useful. We propose, to the best of our knowledge, the first successful attempt to incorporate searcher examination data into passage retrieval for question answering. Specifically, we exploit detailed examination data, such as mouse cursor movements and scrolling, to infer the parts of the document the searcher found interesting, and then incorporate this signal into passage retrieval for QA. Our extensive experiments and analysis demonstrate that our method significantly improves passage retrieval, compared to using textual features alone. As an additional contribution, we make available to the research community the code and the search behavior data used in this study, with the hope of encouraging further research in this area.

1 Introduction

Automated Question Answering (QA), is an attractive variation of search where the QA system automatically returns an *answer* to a user's question, instead of a list of document results. Passage retrieval is a first critical step of QA system, where candidate passages are identified and scored as likely to contain an answer. While significant progress has been made recently on incorporating syntactic and semantic analysis for improving the QA system performance, this analysis is typically applied only on the (limited) set of candidate passages retrieved. The main reason is that it is generally not practical to perform deep analysis on all documents in a large collection, and not yet feasible for the Web at large.

In the web search setting, automated question answering presents additional challenges and opportunities. On the downside, the questions and queries from real users are often not grammatical or well-formed, differing from the questions used in the traditional TREC Question Answering evaluations (Kelly and Lin, 2007; Sun et al., 2005). On the upside, by interacting with a search engine, the millions of searchers implicitly provide additional clues about usefulness of documents, result ranking, and other aspects of the search process. In this paper, we explore making use of the search behavior data to improve passage retrieval for automated Question Answering on the web.

Our basic observation is that when a user is attempting to answer a question, he or she will more carefully examine the parts of the document that contain an answer. This observation is intuitive, and is strongly supported by numerous eye tracking studies (e.g., Buscher et al. (2008) and Buscher et al. (2009a)). Based on this, we hypothesize that the passages containing the answers can be *automatically identified* from the naturalistic searcher behavior, and this prediction can be subsequently used to improve passage ranking. To the best of our knowledge, our work is the first to successfully incorporate searcher examination into passage ranking for Question Answering.

Our approach is primarily aimed at recurring (repeated) questions, which comprise a large fraction of the search volume (while the exact statistics vary, over 50% of search queries are submitted by multiple users). For such questions, a system would track the clicked result URLs, as well as the user interactions on the landing pages. Then the system would use this information to present the improved results to new users who ask the same (or similar) question. Intuitively, our method uses the same general idea of result click data mining, used by the major search engines to improve result ranking, but takes

*Work done at Emory University.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1011–1021,
Seattle, Washington, USA, 18-21 October 2013.

it a step further to exploit user interactions on the actual landing pages. A key point to emphasize is that our approach exploits the *natural browsing behavior* of the users, not requiring any additional effort from the searchers.

Specifically, our contributions include:

- A novel approach to passage retrieval for question answering, that naturally integrates textual and behavioral evidence.
- A robust infrastructure for connecting fine-grained searcher behavior to precise page contents.
- Thorough experiments over hundreds of search sessions and thousands of page views, demonstrating significant improvements to passage retrieval by harnessing the user's page examination data.

Next we describe related work, to place our contribution in context.

2 Related Work

Our work brings together two areas of research: passage retrieval for question answering, and mining searcher behavior data.

Passage retrieval has long been recognized as the first crucial step of automatic question answering. In some cases, passage retrieval can even serve as the final product of a Question Answering system (Clarke et al., 2000). As another example, redundancy in the retrieved passages has been used by the AskMSR system (Brill et al., 2002) to select answers. Tellex et al. (2003) report a thorough comparison of passage retrieval methods for QA, up to 2003. Additional improvements have been achieved by using deeper analysis of the text. For example, Cui et al. (2005) exploited dependency relations between the question terms, Aktolga et al. (2011) incorporated syntactic structure and answer typing, while Harabagiu et al. (2005) used semantic analysis at all stages of the question answering process. In this paper, we pursue a complementary direction, by exploiting searcher examination behavior, with the assumption that human searchers can easily zoom in on relevant passages as part of normal searching.

It has been previously recognized that searcher interactions could be valuable for question answering, and a task on Complex Interactive QA has been ran as part of TREC 2007 (Kelly and Lin, 2007). Our work goes much further by considering not only explicit interactions, but also the searcher examination behavior (i.e., detailed information on which text passages were examined) – which, as we show, provides additional valuable information for passage retrieval. Furthermore, it has been recognized that the questions used in traditional TREC QA evaluation may not be reflective of the "real" questions, posed by users (Bernardi and Kirschner, 2010). Our paper uses a subset of the real questions posted by users on Community Question Answering (CQA) sites, and searches and interactions from real users – which makes our task unique and more challenging than the previous settings.

In particular, our work builds on the rich history of using eye tracking technology to identify areas of interest and attention, and to study reading behavior. In the context of web search document examination, Buscher et al. (2008) extracted sub-documents by tracking eye movements as implicit feedback and expanded search queries to improve the search result ranking. Buscher et al. also studied the prediction of salient Web page regions using eye-tracking (Buscher et al., 2009a). This work, and others, have shown that user attention can help identify regions of documents of particular relevance or usefulness for the query. While eye tracking equipment limits the applicability of these findings to lab studies, these studies served as inspiration to our work to detect the *inferred* areas of interest. Specifically, we use mouse cursor tracking as a natural proxy for user's attention, to replace the requirement for eye tracking equipment. As originally reported by Rodden et al. (2008), the authors discovered the coordination between a user's eye movements and mouse movements when scanning a web search results page. This work was further extended by Huang et al. (2012) to predict the gaze position from mouse cursor movement, with mean error of about 150 px. In summary, there is mounting evidence that the user's attention in web search can be approximated using mouse cursor, scrolling, and other interaction data. In particular, Hijikata (2004) proposed a method to extract text passages of Web pages based on the user's mouse activity and found that extracted passages based on mouse ac-

tivity such as *text tracing, link pointing, link clicking and text selection* enable more accurate extraction of key words of interest than using the whole text of the page. Recently, White and Buscher (2012) proposed a method that uses text selections as implicit feedback for document ranking. Most closely related to this work is a contemporaneous effort on improving web search result summaries, or snippets, by exploiting searcher behavior on the examined documents, described by Ageev et al. (2013). However, to the best of our knowledge, there has been no prior work on modeling searcher interaction on result documents to improve Question Answering performance, and in particular the passage retrieval step.

3 Problem Statement and Approach

This section first states the problem we are addressing more precisely. Then, we describe the key parts of our approach (Section 3.2), and the required infrastructure we had to develop to accomplish the required data collection (Section 3.4).

3.1 Problem Statement

Our goal is to incorporate the searcher behavior (in particular, page examination) into passage retrieval. That is, by analyzing the searcher behavior data, we aim to identify the parts of the page that contain relevant passages for answering a question. Specifically, given a question, a set of queries generated by searchers attempting to answer this question, and a set of documents retrieved by a search engine for each of the queries, our goal to retrieve a set of *passages* that contain correct answers for the question.

That is, our goal is to identify, from searcher behavior, the passages in the documents most likely to contain correct answers to a question, which could then be incorporated into a fully automated question answering system, or returned to the user directly, for example, by incorporating these passages into the result abstracts or "snippets".

3.2 Approach

Our approach accomplishes the goal above by incorporating both textual and behavioral evidence. Specifically, we combine together traditional text-based passage retrieval features, and the inferred user interest in specific parts of a document based on searcher behavior.

First, a passage score is obtained from the QA-SYS system (Ng and Kan, 2010), resulting in a strong text-only baseline that generates *candidate passages*. Separately, examination behavior data is collected over the landing pages, using our logging infrastructure described in the next section. Then, a behavior model is trained to identify the passages of interest to the user, based on user examination data (Section 4.2). Finally, the behavior-based prediction of interest in each candidate passage is combined with the original (text-based) passage score, in order to generate the final *behavior-biased* passage ranking (Section 4.3). Note that by decoupling the behavior modeling from the candidate generation method, our approach can be used with any other passage retrieval approach that provides scores for the candidate passages (that could be combined with the behavior scores for the final ranking step).

While general and flexible, our approach makes two key assumptions, resulting in potential limitations. First, our approach is primarily targeted (and evaluated for) informational questions – that is, questions for which the user expects to find an answer in the text of the page. For other question classes (e.g., opinion), passage retrieval might have to be optimized differently. We also assume that the user interactions on landing pages can be collected by a search engine or a third party. This is not far-fetched: already, browser plug-ins and toolbars collect some form of user interactions on web pages, major organizations can (and sometimes do) use proxies, and common page widgets like banner ads and visit counters commonly inject JavaScript to monitor basic user interactions – and can be easily extended to collect the examination data described in this paper. The privacy and security of these methods are beyond the scope of this paper, we merely point out that these behavior gathering tools, assumed by our approach, already exist and are already widely deployed. The interested reader can obtain an overview of the relevant privacy issues and proposed solutions in references (Mayer and Mitchell, 2012; Krishnamurthy and Wills, 2009).

3.3 Acquiring Search Behavior Data

Our infrastructure for acquiring search behavior was developed with two goals in mind: (1) to obtain behavior data similar to real-world search, with the ability to track fine-grained search behavior such as

a mouse cursor movement (as there are no publicly available data of this kind); (2) to create a controlled and clean ground truth set, to train our system and evaluate the effectiveness of our approach.

To collect sufficient amount of search behavior data, we adapted for our task the publicly available UFindIt architecture, described in reference Ageev et al. (2011). The participants played several search contests, or "games", each consisting of 12 search tasks (questions) to solve. The stated goal of the game was to submit the highest possible number of correct answers within the allotted time. After the searcher decided that they found the answer, they were instructed to type the answer together with the supporting URL into the corresponding fields in the game interface. Each search session (for one question) was completed by either submitting an answer or clicking the "skip question" button to pass to the next question.

Participants were recruited through the Amazon Mechanical Turk (MTurk) service. As a first step, the workers had to solve a ReCaptcha puzzle to verify that they are human and not an automated "bot". A browser verification check was performed to confirm that the browser was compatible with our JavaScript tracking code. During the data postprocessing stage, we filtered out the users who did not answer even the easy, trivial questions, as it indicated either poor understanding of the game rules, or an attempt to make a quick buck without effort.

In order to capture all of the participants' search actions, they were instructed to use only our search interface (and not a separate browser window). The search interface performed the web searches using the public API of a popular web search engine, and showed result pages to the users using the original page design, layout and stylesheets, so the user's search experience is not affected.

3.4 Page Examination Behavior Logging

A key part of our system is a mechanism for collecting searcher interactions on web pages, and tying them precisely to the page content at the word level. As the HTML page passed through the proxy, a JavaScript code is embedded to track the user's interactions, including mouse movements and scrolling, as well as the properties of the visited page. The behavioral (interaction) events are logged by the search interface proxy and written to the server log.

To connect the tracked mouse cursor positions to exact text passages we employed the following trick. After the HTML page is rendered in the browser window, our JavaScript code modifies the page DOM tree so that each word is wrapped by a separate DOM Element. Then for each DOM Element, the window coordinates of that element are evaluated and saved in an Element's attribute. The processed HTML page is then saved to the server by an asynchronous request. The saved coordinates are updated if the page layout is changed due to *resize* window event or AJAX action.

As a result of this instrumentation, for each page visit we know the searcher's intent (question), a search engine query that the user issued, a URL and HTML page, the bounding boxes of each word in the HTML text, and all of the searcher actions, e.g., mouse movement coordinates, mouse clicks, and scrolling.

4 Behavior-Biased Passage Retrieval

We now present the details of our behavior-biased passage retrieval algorithm (BePR). First, we describe the text-only retrieval system. Then, we introduce our method for inferring the most interesting or useful parts of the document from user behavior (Section 4.2).

4.1 Text-Based Passage Retrieval

We adopt an open-source question answering framework QANUS (Ng and Kan, 2010) (version v29Nov2012). The QANUS distribution contains the fully functional factoid QA system QA-SYS that we use as a baseline for our experiments. QA-SYS implements many of the state-of-the-art question answering techniques, and is similar to a top-performing QA system from TREC (Sun et al., 2005). The QA-SYS distribution is configured for processing documents and questions in TREC QA format, and we adopted QA-SYS for answer extraction from web documents. QA-SYS takes a set of documents and a question as an input, and processes the input in three stages: (1) information source preparation, (2) question processing, and (3) answer retrieval.

In the first stage, the downloaded HTML pages are pre-processed with Natural Language Tool Kit (NLTK, Bird (2006)). Extracted text is divided into sentences using *Punkt* unsupervised sentence split-

ter (Kiss and Strunk, 2006). The QA-SYS performs Part of Speech tagging using Stanford POS tagger (Toutanova et al., 2003), and Named Entity Recognition using Stanford NER (Finkel et al., 2005), and then builds a Lucene index over the set of input documents. In the second stage the QA-SYS performs POS tagging, NE recognition, and question type classification for an input question.

To answer a question, QA-SYS creates a query from the question, performs the search over the indexed text collection, and retrieves top 50 documents. Each document is split by sentences, and for each sentence a *QA-SYS Passage Retrieval Score* ($TextScore$) is computed as a linear combination of term frequency score, proximity score, and term coverage score. After that 40 passages with the highest $TextScore$ are retrieved, for each passage QA-SYS performs pattern based answer extraction based on the identified expected answer type of the question.

As the focus of this paper is to improve Passage Retrieval performance, we use the $TextScore$ sentence ranking as a baseline, and improve on it by adding the new search behavioral features indicating the passage relevance, as described next.

4.2 Inferring Relevant Passages from Search Behavior

To rank passages by their "interestingness" – that is, to identify the passages that have been carefully examined by the searcher, we use a learning-to-rank approach, and apply regression algorithms to predict the probability that a specific passage is interesting for a user. A passage is labeled as "interesting", if the user submitted an answer in the current session, and both the passage and the answer have at least one common word, after stemming and stop-word removal.

For each passage, a set of behavior features that could represent passage interestingness is created. To associate behavioral features with a given document passage, we match the sequence of behavior events and the set of bounding boxes for each word and DOM Element of a page. For efficiency, we build a spatial R-Tree index of these bounding boxes, which allows us to quickly find the matching DOM Elements for each event.

One key feature is the duration of the time interval when a mouse cursor was hovering over the specific text passage, or very close to the passage. We also take a scrollbar and event count features from papers (Buscher et al., 2009b), and (Guo and Agichtein, 2012) to detect evidence of "reading" vs. "skimming" behavior, and adopt those features to represent the behavior near the specific location of a page. The full set of our passage behavior features are reported in Table 1.

Feature	Description
MouseOverTime	Time duration when the mouse cursor was over the text passage
MouseNearTime	Time duration when the mouse cursor was close to the text passage in the window $(x \pm 100px,\ y \pm 70px)$
MouseOverEvents	The number of mouse events during *MouseOverTime*
MouseNearEvents	The number of mouse events during *MouseNearTime*
DispTime	Time duration when the text passage has been visible in the browser window (depends on scrollbar position)
DispMiddleTime	Time duration when the text passage was visible in the middle part of the browser window

Table 1: Behavior features for text passages

To implement the passage ranker, we experimented with a variety of learning-to-rank (LTR) algorithms, and chose two implementations of Regression Trees, due to their strong performance for general web search ranking tasks. The first algorithm is Regression Tree (Friedman et al., 2001), and the second is Gradient Boosting Regression Tree algorithm (Friedman, 2001). They are named *BePR-BTree*, and *BePR-GBM* respectively.

The dataset consists of a set of questions, with associated search behavior data collected from all the users who tried to find an answer to this question, the answers submitted by the users, and a set of validated answers. These sets are divided into training, validation, and test, so that the training and validation set URLs are disjoint, and the test set have no intersection with training and validation set by URLs, questions, and users. The training set is created from only those page visits where the document text has non-empty intersection with the user's answer, and the answer is correct. The trained regres-

sion algorithm is applied to all page visits in the test set. When the trained model is applied at test time, it has no information about the user's intent, the correct answer, or the current query, but rather uses only the behavioral features of the current page visit to identify the "interesting" passages.

The predicted probability of passage interestingness is averaged over all the users and page visits, and the resulting passage interestingness is then used as the $BScore$ of the passage. Note that $BScore$ is defined for only *visited* pages; to incorporate the overall clickthrough information (i.e., the fraction of the time a page was visited, indicating relevance), we introduce a generalized version, designated as $BSscore_{All}$, defined as: $\gamma \cdot CTR + (1-\gamma) \cdot BScore$, where CTR is the clickthrough rate for the page, defined as the fraction of time the result was clicked for all searches. Intuitively, this version reduces the weight of the behavior score for the pages with insufficient behavior data by "backing off" to the document clickthrough rate, according to the parameter γ. For the cases where only the visited pages are considered (ignoring the searches when the page was not visited), γ is set to 0, reverting the score to the original $BScore$ definition. The resulting behavior-based passage score is then used as the aggregate value of searcher interest in the passage for the combined passage retrieval step, described next.

4.3 Combining Textual and Behavioral Evidence

The final step in our approach is to *combine* the text-based score $TextScore(f)$ for a sentence (Section 4.1) with the interestingness score $BScore(f)$ (Section 4.2), inferred from the examination data. In our current implementation we combine these scores by linear combination:

$$FScore(f) = \lambda \cdot BScore(f) + (1 - \lambda) \cdot TextScore(f)$$

Other more sophisticated ways to combine text and behavior evidence are possible, such as jointly learning over both text and behavior features. However, we chose to follow the simpler linear approach for interpretability of the results (e.g., by varying the λ parameter).

5 Data Collection and Experimental Setup

This section presents the methodology used for selecting the questions (Section 5.1), the corresponding search behavior data (Section 5.2), and the experimental collections and metrics (Section 5.3).

5.1 Questions

The search tasks were selected from community question answering sites such as wiki.answers.com and Yahoo! Answers by the researchers. The criteria used were that the question should be clearly stated, had a clear answer, and that finding this answer was not a trivial task, that is, the answer was not retrieved simply by submitting the question verbatim to Google, Bing, or Yahoo! Search engines. Overall, 36 such questions were selected, posing (as it turned out) greatly varying levels of difficulty for participants. These questions were randomly split into three game rounds of 12 questions each.

5.2 Browsing Behavior Dataset

The search behavior data for each of the questions above was acquired as described in Section 3.3. A total of 270 participants finished the game. After filtering out users who did not follow the game rules, we have 3047 search sessions performed by 265 users. Our data for these users consists of 7800 queries, 3910 unique queries, 8574 SERP clicks on 1544 distinct URLs. For 5683 page visits (66%) and 883 distinct URLs the on-page behavioral data is collected. For the rest 34% of page visits the behavioral data were not collected due to conflicts between our JavaScript tracking code and other code presented on the page. For each page view there are about 400 atomic browsing events (mouse movements, scrolling, key pressing) on average. All the source and derived data are available at `http://ir.mathcs.emory.edu/intent`.

The dataset is divided into training, validation, and test set in the following way. The behavior dataset for the first game is divided randomly into equal-sized training and validation sets that are disjoint by URLs. The training set was used to train the regression algorithm for predicting passage attractiveness, and the validation set was used to explore the influence of behavior weight λ on passage retrieval performance, and to select the parameter λ for using on a test set. The validation set consists of 254 different URLs spread over 11 questions, and

for each of them there is a collected browsing behavior.

The test set consists of 441 URLs spread over 24 questions, and the test set has no intersection with training and validation set by URLs, questions, and users.

5.3 Candidate Document Selection Strategies

The first step for question answering is a selection of a candidate document set. In our settings, we may select a subset of web documents in a different way. We explore passage retrieval effectiveness using three different strategies of document set selection.

- For each question select *All* documents that are in top 10 documents returned by a search engine for any query that was issued during search for the specific question. For our dataset this gives around 500 candidate documents per question on average.

- For each question select only documents that were *Clicked* by a user. This restricts a candidate document set to set of most promising documents. For our dataset this gives around 25 candidate documents per question on average.

- For each pair of question and *Relevant* document apply passage retrieval to the specific document. In this experiment we label a document "Relevant" if a correct answer was extracted from it. In a real-world scenario, while document relevance could be estimated by a variety of click-based methods, we address the challenge of how to actually extract the correct answer from the document, automatically, with the help of the natural behavior data. We perform this experiment to estimate the performance of passage retrieval for the case when relevant documents are known with high confidence.

Evaluation Metrics: We evaluate passage retrieval performance by standard Mean Reciprocal Rank (MRR), and Mean Average Precision (MAP) metrics for top 20 retrieved sentences (Voorhees and Tice, 1999). We also evaluate ROUGE-1 metric (Lin, 2004) for the first retrieved passage.

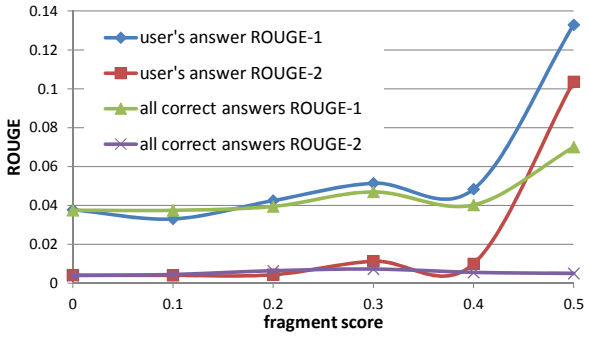

Figure 1: The actual passage interestingness, measured by intersection with user's answer, vs. the passage relevance score $BScore$ predicted from behavior data

6 Results

We now present the empirical results. First, we report the intermediate result of using behavior data to infer the interesting (useful) passages in the document. Then, we report the main results of the paper where the quality of the generated snippets with and without using behavior data is compared using human judgments.

6.1 Prediction of Passage Interestingness

This experiment evaluates how well we can predict interesting passages by observing a user's on-page behavior. We suppose that the passage is interesting if it is related to the answer for the question. For each visited page, we collect the user's answer (if submitted), and all correct answers from all users who answered this question. Then, we compare those answers to each text passage in the document using ROUGE metrics (Lin, 2004).

Figure 1 shows the relationship between the interestingness of a passage and behavior score. The graph shows that when the score is high ($\geq$ 0.5), then average intersection between the passage and user's answer is much higher than those when the passage score is low. All ROUGE-N metrics significantly grow when the behavior score grows, although ROUGE-2 over all correct answers are always very small (it grows from 0.003 to 0.007). ROUGE-1 is much greater than ROUGE-2 for high scores, as the interesting passage might contain useful information for the answer, but the user reformulates the obtained information and submits reformulated answer. The ROUGE-N metrics for a user's answer are much greater than those for all correct

Feature	Feature Importance
DispMiddleTime	0.51
MouseOverTime	0.34
DispTime	0.12
MouseNearTime	0.02
MouseOverEvents	0.01
MouseNearEvents	0.01

Table 2: Feature importance for behavioral features, as measured by Gini coefficient

answers, as other users might obtain valuable information from other documents, and some questions have distinct correct answers.

Behavior Feature Importance Analysis: To estimate relative importance of behavior features we evaluated the Gini importance index (Breiman, 1996) for each behavior feature from the Table 1. The Table 2 shows that the most important features are the time duration when the text passage was visible in the middle part of the scrolling window, and the time duration when the mouse cursor was over the text passage. The first feature has been shown to be a good feature for re-ranking search results in reference (Buscher et al., 2009b), and we have shown that it is also useful for passage retrieval. The *MouseOverTime* feature has been previously shown to be correlated with examination time, measured by eye-tracking experiments (Guo and Agichtein, 2010), and it helps us detect local behavior in the neighborhood of a specific text passage.

Analysis of Searcher Attention: In order to better understand what characteristics of the textual passages attract the searcher's attention, we explored 21 linguistic features for each sentence. Our features were designed to estimate text readability, and the overlap of a passage with the query that was used to find the document. We implemented the readability features from (Kanungo and Orr, 2009), and query matching features from (Metzler and Kanungo, 2008). Table 3 reports the top 10 features with the highest absolute value of the correlation coefficient with passage interestingness score $BScore$. Interestingly, the most highly correlated features are related to readability, while query matching features are less important.

Feature description	*corr*
Number of distinct words in the passage	0.31
Total number of words in the passage	0.28
Number of letter ([a-zA-z]) characters	0.27
Relative location of the passage in the document	-0.25
Number of unique words in the passage divided by total number of words	-0.24
Number of punctuation characters	-0.20
Number of words with first letter capitalized	-0.17
Overlap of query terms expanded with synonyms and the passage	0.15
Absolute count of query terms matched in the passage	0.15
Average position of query term within the passage	-0.14

Table 3: Correlation of passage interestingness $BScore$ with linguistic properties of a sentence

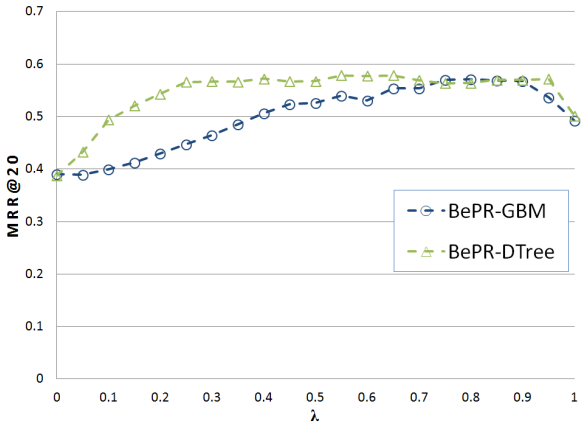

Figure 2: MRR for passage retrieval for varying behavior weight λ and interestingness prediction algorithms *BePR-DTree*, and *BePR-GBM*

6.2 Passage Retrieval with Behavior Data

This section reports the main results of the paper. First, we describe the parameter tuning, followed by the main performance results.

Parameter Tuning: To tune the passage retrieval performance, we use the validation set to find the optimal value for λ. Figure 2 reports the passage retrieval MRR for varying λ, for two learning algorithms BePR-GBM and BePR-DTree. The figure shows that both BePR-GBM and BePR-BTree improve over the QA-SYS baseline. BePR-GBM algorithm achieves the best performance with $\lambda = 0.8$, and also exhibits more robust behavior compared to BePR-BTree, so we use BePR-GBM with $\lambda = 0.8$ for the main experiments described next. Similarly, using the training and validation sets, we optimized

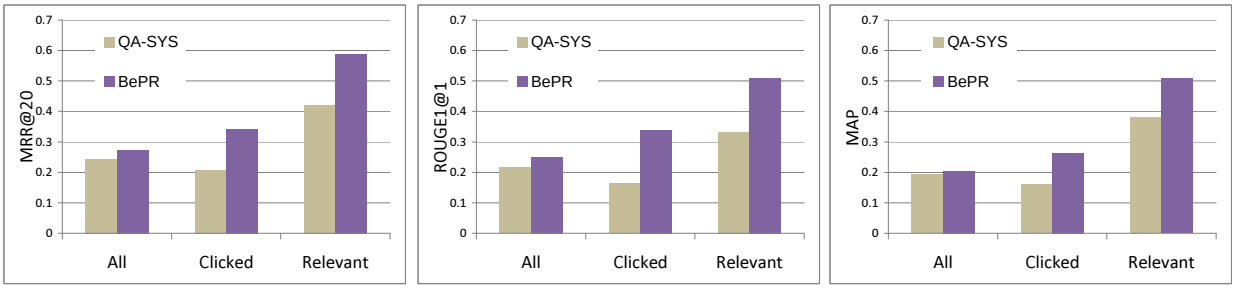

Figure 3: Passage retrieval MRR (a), ROUGE1 (b), and MAP (c) for the BePR and QA-SYS systems, on the test set.

the value of the clickthrough rate weight $\gamma = 0.05$ (used for the $BScore_{All}$ score) for the *All* document set only (as for the *Clicked* and *Relevant* document sets, γ is always set to 0 by construction).

Main retrieval results: We now compare the baseline algorithm for passage retrieval implemented in QA-SYS system and described in section 4.1 with the BePR algorithm (section 4.2-4.3) that combines the textual passage score and the behavior score using the λ parameter for the relative weight of the behavior evidence.

Figure 3 reports the main results of the paper, namely the MRR, ROUGE-1@1 and MAP passage retrieval metrics for the baseline QA-SYS algorithm, and BePR-GBM, on the test set. As the figure shows, BePR achieves higher performance on all metrics, and for all document sets. The improvements are statistically significant ($p < 0.01$) for experiments with Clicked and Relevant document sets. Not surprisingly, the improvements are smallest when *All* documents are considered, as unclicked documents do not provide any associated behavior data. As the results show, our simple back-off strategy (using the document clickthrough rate with the γ parameter) is moderately successful, but could be further refined in the future.

Finally, we illustrate how behavior features affect passage ranking. Let's consider a question "*How many Swedes speak English as a percentage?*". The perfect relevant page for this question is a Wikipedia page "Languages of Sweden". A sentence "Main foreign language(s): English 89%, German 30%, French 11%." contains an answer to the question, but it has only a small intersection with question terms, and QA-SYS ranks this question in the 13th place. Other sentences that contain a country name, a number, or have more terms that match the question are ranked higher. In contrast, as searchers examined this sentence carefully to find the answer, BePR is able to promote this sentence to the second place in the ranking.

7 Resources and Data

All the code and the collected data used in this research are available at `http://ir.mathcs.emory.edu/intent/`. The dataset contains the set of questions used for the experiments, and user's behavior: queries submitted by users to search engine, result pages, visited URLs, downloaded landing pages, on-page browsing behavior (mouse movements, scrollbar events, resize actions, clicks). By sharing our code and data, we hope to encourage further research in this area.

8 Conclusions and Future Work

We presented the first successful approach to incorporating naturalistic searcher behavior data into passage retrieval for question answering. Specifically, we developed a robust method to infer searcher interest in specific parts of the document, which could then be combined with more traditional textual features used for passage retrieval. Our results show significant improvements over a strong baseline, derived from a competitive Question Answering system.

To implement the proposed method in a real-world search engine for Web QA, the proposed infrastructure and/or the released data could be used as a training set for the algorithm that predicts fragment interestingness from user behavior. Such a system would need to track document examination data. This can already be done by incorporating our released tracking code or a similar method into a

browser toolbar, banner ad system, visit counters or other JavaScript widgets that already track user visits. While we acknowledge user privacy as an important concern, it is beyond the scope of this work.

In the future, we plan to extend this work to more precisely pinpoint the answer location on a page, and consequently incorporate searcher behavior into subsequent answer extraction and ranking stages of question answering. We also plan to further investigate the examination data to better understand how searchers find correct (and incorrect) answers using both general web search engines and QA systems – in order to inform and further improve query suggestion, result snippet generation, and result ranking algorithms.

Acknowledgments

This work was supported by the National Science Foundation grant IIS-1018321, the DARPA grant D11AP00269, the Yahoo! Faculty Research Engagement Program, and by the Russian Foundation for Basic Research Grant 12-07-31225.

References

Mikhail Ageev, Qi Guo, Dmitry Lagun, and Eugene Agichtein. 2011. Find it if you can: a game for modeling different types of web search success using interaction data. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 345–354, New York, NY, USA. ACM.

Mikhail Ageev, Dmitry Lagun, and Eugene Agichtein. 2013. Improving search result summaries by using searcher behavior data. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13.

Elif Aktolga, James Allan, and David A. Smith. 2011. Passage reranking for question answering using syntactic structures and answer types. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 617–628, Berlin, Heidelberg. Springer-Verlag.

Raffaella Bernardi and Manuel Kirschner. 2010. From artificial questions to real user interaction logs: Real challenges for interactive question answering systems. In *Proceedings of Workshop on Web Logs and Question Answering*, pages 8–15.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, COLING-ACL '06, pages 69–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proc. of ACL*, EMNLP '02, pages 257–264, Stroudsburg, PA, USA. Association for Computational Linguistics.

Georg Buscher, Andreas Dengel, and Ludger van Elst. 2008. Query expansion using gaze-based feedback on the subdocument level. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 387–394, New York, NY, USA. ACM.

Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. 2009a. What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 21–30. ACM.

Georg Buscher, Ludger van Elst, and Andreas Dengel. 2009b. Segment-level display time as implicit feedback: a comparison to eye tracking. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 67–74, New York, NY, USA. ACM.

Charles Clarke, Gordon Cormack, Derek Kisman, and Thomas Lynam. 2000. Question answering by passage selection (multitext experiments for trec-9). In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 400–407, New York, NY, USA. ACM.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*, volume 1. Springer Series in Statistics.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):pp. 1189–1232.

Qi Guo and Eugene Agichtein. 2010. Towards predicting web searcher gaze position from mouse movements. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3601–3606, New York, NY, USA. ACM.

Qi Guo and Eugene Agichtein. 2012. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 569–578, New York, NY, USA. ACM.

Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, Andrew Hickl, and Patrick Wang. 2005. Employing two question answering systems in trec-2005. In *Proceedings of the fourteenth text retrieval conference*.

Yoshinori Hijikata. 2004. Implicit user profiling for on demand relevance feedback. In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI '04, pages 198–205, New York, NY, USA. ACM.

Jeff Huang, Ryen White, and Georg Buscher. 2012. User see, user point: gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1341–1350, New York, NY, USA. ACM.

Tapas Kanungo and David Orr. 2009. Predicting the readability of short web summaries. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 202–211, New York, NY, USA. ACM.

Diane Kelly and Jimmy Lin. 2007. Overview of the trec 2006 ciqa task. In *ACM SIGIR Forum*, volume 41, pages 107–116. ACM.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32(4):485–525, December.

Balachander Krishnamurthy and Craig Wills. 2009. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 541–550, New York, NY, USA. ACM.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. Barcelona, Spain, July. Association for Computational Linguistics.

Jonathan R. Mayer and John C. Mitchell. 2012. Third-party web tracking: Policy and technology. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 413–427, Washington, DC, USA. IEEE Computer Society.

D. Metzler and T. Kanungo. 2008. Machine learned sentence selection strategies for query-biased summarization. In *SIGIR Learning to Rank Workshop*.

Jun-Ping Ng and Min-Yen Kan. 2010. Qanus: An open-source question-answering platform http://www.comp.nus.edu.sg/ junping/docs/qanus.pdf.

Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. 2008. Eye-mouse coordination patterns on web search results pages. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2997–3002, New York, NY, USA. ACM.

Renxu Sun, Jing Jiang, Yee Fan Tan, Hang Cui, Tat-Seng Chua, and Min-Yen Kan. 2005. Using syntactic and semantic relation analysis in question answering. In *TREC*.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 41–47, New York, NY, USA. ACM.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ellen Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *Proceedings of The Eighth Text REtrieval Conference (TREC-8), http://trec. nist. gov/pubs/trec8/t8_proceedings. html*.

Ryen W. White and Georg Buscher. 2012. Text selections as implicit relevance feedback. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1151–1152, New York, NY, USA. ACM.