SIRA: TREC Session Track 2014

Brennan Bushee, Drew Pintus, Patrick Smith, and Sharon Gower Small
Siena College Institute for Artificial Intelligence
515 Loudon Road
Loudonville, NY 12211
bbushee@skidmore.edu, dc04pint@siena.edu, psmith4@buffalo.edu, ssmall@siena.edu

1. Abstract

This paper discusses Siena's Interactive Research Assistant's (SIRA) participation in the Text Retrieval Conference (TREC) Session Track of 2014. The overall goal of this track is to improve search results during query sessions based on a user's behavior. Query sessions include many aspects of a search, including query topics, initial retrieved webpages, clicked on links, visit times, etc. SIRA has used several methods to improve search results that will be discussed in this paper. Each method of query expansion utilized clicked-on and non-clicked-on links, pages with the longest visited time, and N-Percent (N%) of each page. Two of our three submissions improved over our baseline results and both of these were equal to the median submission for all participants in the track.

2. Introduction

The Session Track is a program in the Text Retrieval Conference (TREC). TREC is a program co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense and it focuses on supporting research in information retrieval and extraction, and to increase availability of appropriate evaluation techniques. The Session Track [1] is in its fifth year and it focuses on using session information to improve system results during a series of queries. The session information that is available for use includes the top 10 pages returned for a query, their titles, a small preview of the page called a snippet, whether or not that page was clicked on and how much time the user spent on the clicked on pages. The idea is to lead the user to their answers by taking this information and giving them a new set of pages that are more tailored to their specific needs. The official scoring for the track uses Discounted Cumulative Gain for the top 10 pages, or nDCG@10. Normalized discounted cumulative gain (nDCG) is the DCG of the results divided by the optimal DCG. DCG measures the quality of the ranking of the returned documents, based on the assumption that more relevant documents should be placed higher in the returned list. For each session, the NIST judges rated each result by relevance: -2 (spam), 0 (not relevant), 1 (relevant, the content of the page provides some information of the topic), 2 (highly relevant, the content of the page provides substantial information of the topic), 3 (key, the page is dedicated to the topic) to 4 (navigational, the page represents a home page of an entity named in the query).

3. Related Research

Since the Session Track began in 2009, many universities and groups have participated and reported their results and methods in detailed papers. Georgetown University [2] used slightly different approaches in three separate runs for their participation in the 2013 Session Track. Their first run involved a simple retrieval system, their second was based on query modification of the user, and lastly their third run incorporated results that were clicked on. One of the more successful groups, from the University of Pittsburgh [3], also participated in the 2013 Session Track. University of Pittsburgh decided to move away from their 2012 method of using past queries, and instead used the relevance of the returned pages. The reason for the change comes from their study of their system from 2011 and 2012. According to these studies, they reported

that incorporating past queries tended not to affect the search performance significantly. Their system also utilized results that occurred multiple times rather than simply discarding them.

4. SIRA

The main focus of the SIRA system was to utilize "typical search behavior" in order to provide a better ranking of results to a user. This meant analyzing how a user normally utilizes a search engine and identifies what they see as most important/relevant. Such behaviors include skimming for interesting titles, reading only a certain small percentage of a page's content, and investing time on a page that they feel contains information that is useful to them, etc. By using these observations, we were able to take the session information and utilize it to return an improved re-ranking of results.

The remainder of this paper will discuss the modules of our SIRA system in detail as well as the results of our NIST evaluation.

5. SIRA: Document Retrieval

5.1 The Corpus and interface

The ClueWeb12[1] corpus distributed by Carnegie Mellon University consists of about 733 million web pages. Participants are able to access ClueWeb12 through a web search interface that utilizes the Indri search engine [4]. Indri is able to take in many different parameters, from "and-or" statements, to word displacement checks, to frequency of word appearance checks. We utilized Indri to effectively search the collection of web pages to retrieve an initial set of potentially relevant documents for a user's query. SIRA generates a list of parameters automatically after its first run by taking words from both the query and the information about the current session and makes a decision on what parameters should be wrapped around said words. Specifically, our system extracted nouns, verbs, adverbs, and adjectives and finds the highest frequency synonym. After we get a list of those words, we add them to the current query. We then produce a query text file and upload it to the ClueWeb12 Batch Query Service page to retrieve 100 relevant documents.

5.2 Spam Filtering Module

The Spam Module incorporates a list of over 700 million ClueWeb12-IDs paired with a spam score, given by Waterloo Spam Rankings[2] for the ClueWeb12 Dataset. The scores range from 0 to 99 where 0 is most spam-like and 99 is least spam-like. Our original spam threshold was set to 30 (based on what groups had used in the past). After several experiments and manual judging, with spam threshold increments of five, we determined the optimal spam threshold to be 35.

6. SIRA System

SIRA's first process the TREC session file. It extracts all of the relevant information and organizes it in terms of a session's individual searches, their results, and how the user interacted with those results. The SIRA controller then takes that information, checks and runs each session through both the clicked and longest visit time modules or the framing module, depending on what was specified in the Control File. SIRA then produces a query expansion file formatted for clueWeb12 search engine batch search.

---

1 http://www.lemurproject.org/clueweb12.php/

2 http://www.mansci.uwaterloo.ca/~msmucker/cw12spam/

A manual retrieval of the batch results is required and after that's done, we run the program again. This time the controller uses our HTML to text program to extract each result retrieved from the batch result. Next our spam module compares the retrieved pages to a list of possible website IDs and removes any that match. Finally, Garbage Collect removes any duplicates and makes sure everything is in order before finally outputting a final list of top 10 results for each session.

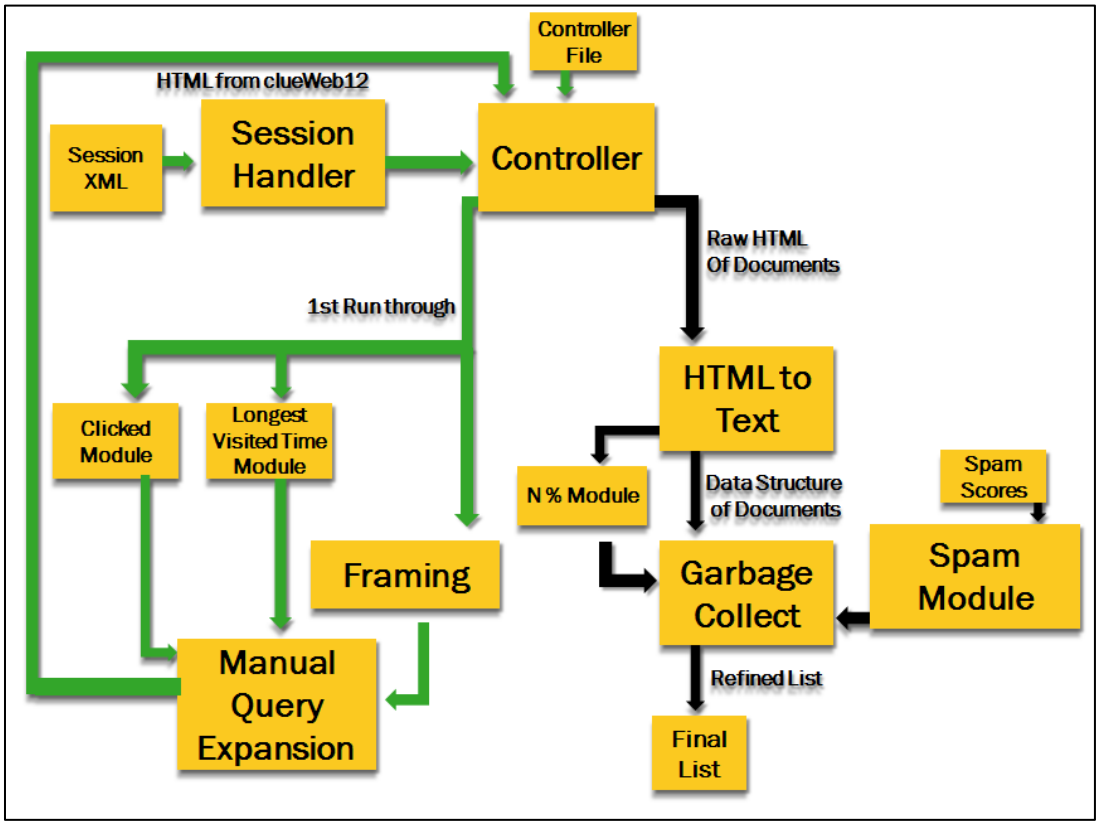

Figure 1: SIRA System Architecture

6.1 N% Module

The N% Module involves extracting the first N% of a document. This module was developed to model a user's typical web searching behavior. When a user enters a query and is presented with results, they normally scan or read the first few sentences of the document they clicked on to quickly judge its relevance. We developed a module that simply takes the given percent of each document and analyzes it from there. This method is much more efficient than searching through an entire document. We determined through our experiments that 10% is the optimal amount of a page to be analyzed by SIRA.

6.2 Clicked Module

We hypothesized that if a page was clicked on, then there was something attractive about the link that led the user to believe the full page would be relevant. A logical answer would be that the title contained words that the user found useful or relevant. The Clicked Page module takes information found in the current session and extracts the most clicked page (s)' titles. It then takes those titles and creates a batch query text file which is used to retrieve 100 documents.

The results of this module turned out to be better than anticipated. In the original test data, SIRA's clicked module produced an average nDCG@10 score of 0.15. Official results from our TREC submission had an average nDCG@10 score of about 0.17.

6.3 Longest Visited Time Module

We also hypothesized that a page that was visited for a long time must have some information that the user found interesting or important. With that in mind, we built a module that automatically determined which results had been visited for the greatest length of time and extracted the snippets that accompanied it. To see what information was relevant to our search, we extracted the most frequent words including synonyms. For this process, words were extracted from the document's snippet. These words were then added to the batch query text file for query expansion. Using this method, we scored an average nDCG@10 of 0.169, which was just below the average of all teams participating, 0.1702. In the future, we would like to try and refine what parts of the snippet we decide to place in the batch query text file so that our query expansion is directed more accurately.

7. Framing

In order to impose some order on the large amount of unstructured data, SIRA stored all the "important" words in a structure called a frame. Frame semantics have been used before in special-purpose question answering systems, such as HITIQA, which was designed for intelligence analysts researching weapons of mass destruction [5]. In these domains, a small number of specific frames can be added to the basic general frame to describe all the events of interest to the user, but due to the wide range of possible search topics for our task, we decided to use just the general frame consisting of nouns, verbs, people, locations, and organizations recognized in the text. The system created a "goal frame" for each query and then created a "data frame" for each search result.

7.1 Entity Recognizer

The Stanford Named Entity Recognizer (SER) [6] was one resource used in the framing of text. In the RL2 run we framed the current query and the description of the topic in each query. Using SER aided in this process. SER searches through a string of text and tags or labels sequences of words that are entities. These include people, locations, organizations, etc. This information was used as elements of our frame.

7.2 Frame Scoring

Documents were given relevance scores based on how well their data frame matched the goal frame. For example, in Session 11 from the 2012 Session Track, the user searched for information about the sinking of the Russian nuclear submarine *Kursk.* In the RL1 run, only the current query was framed.

```
Current query: "kursk UK subs in area"
Current query framed:
       Nouns: [kursk, subs, area]
       Verbs:  [none]
       Locations: [UK]
       People: [none]
       Organizations:  [none]
```

In our RL2 run, the system added previous queries and snippets from clicked links to the goal frame. Similarly, the N% module trimmed each document to 10 percent of its original size, and multiple snippets were taken from each result page. The one that generated the highest-scoring data frame was used as the data frame for that page. Here is an example of a frame created from a snippet:

```
 Clicked Snippet:
    "Russian Submarine Sinks, Killing 9 Crew VLADIMIR ... to boost the prestige of the
Russian navy, badly hurt by the August 2000 sinking of the Kursk nuclear submarine"

Clicked snippet framed:
       Nouns: [Submarine, boost,  prestige, navy]
       Verbs: [Kill, sink, hurt]
       Locations: [Russian]
       People: [none]
       Organizations: [none]
```

The frames are awarded one point for each field in which there is a word matching one in the corresponding field of the goal frame. For nouns and verbs, synonyms and hypernyms count as matches, and all verbs are converted to infinitive form before they are stored in the frame for ease of comparison. The framing module showed some promise in training, but did not perform as well as we had hoped on the real data. From the results, we were able to identify some areas in which it could improve:

- It is possible for a document to receive a high score if it contains words that are only tangentially related to one in the goal frame. It may be better to change our scoring system that takes into account distance between the matches in a semantic network.

-Another unforeseen problem was that sometimes common web words such as "search" and "contact" were added to the goal, creating false matches. Thus the system needs a way of knowing whether these words are part of the meaning of the text. Simply adding these common web search words to a stop word list would fix this.

-On average for framing, the RL2 results were worse than those of RL1. This suggests that a frame that is too large may be too general and generate false positives. The optimal size of a goal frame is open for experimentation.

-The system sometimes erroneously splits up phrases into different categories. For example, "Russian navy" in the above example may be better categorized as an organization; the system instead stored "Russian" as a location and "navy" as a noun.

8. Garbage Collecting

Our garbage collecting module had the purpose of producing a list of ranked documents that did not share a common base URL. For example, if one of the documents has the URL “www.amazon.com/scooter”, another document cannot have a URL starting with “www.amazon.com”. The base URL of each document in question was compared to the rest of the URLs of documents gathered by the program. If any of the base URLs matched, the ranked list only returns the top rated document with the corresponding base URL. Although it seemed

like this method wasn't useful at first, through closer inspection it removed many duplicated pages and gave our user more unique pages in their results.

9. Results

Our evaluation results can be seen in the graph below. The highest RL2 nDCG@10 score that we received was our Clicked module, which gave a 3% increase over our RL1 score. Our LVT also gave a slight increase with a 2.37% increase from RL1 to RL2. Our RL1 score was significantly higher than last year's median RL1 score, which was 0.1171, and our highest RL2 score was an improvement over last year's median of 0.15305 and equivalent to this years average.

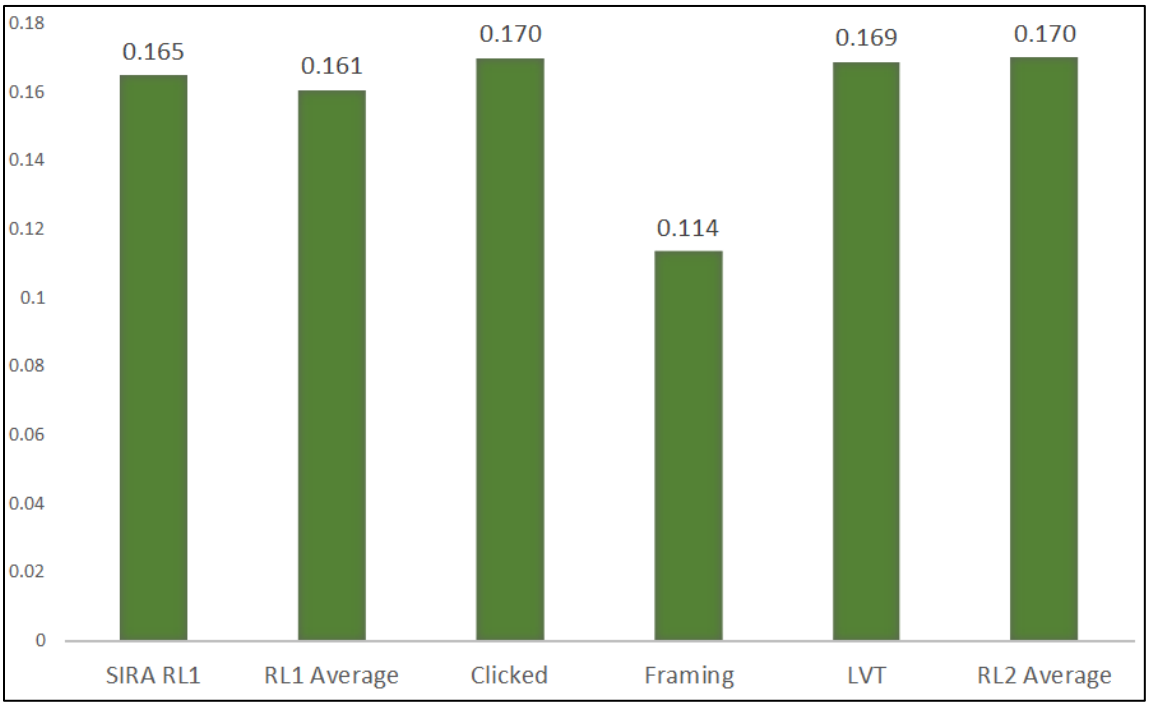

Figure 2: nDCG@10 results of SIRA modules and the TREC averages.

10. Conclusion

This paper reviews our work done for the TREC Session Track for 2014. The results show that it is possible to use past searches to effectively direct a future query with our Clicked and LVT modules. We are currently working on modifying our framing module as identified above and running new experiments utilizing the NIST judgment file to improve its performance.

11. References

[1] Cartette, Ben, Ashraf Bah, Evangelos Kanoulas, Mark Hall and Paul Clough. (2013). Overview of the TREC 2013 Session Track. The Twenty-Second Text Retrieval Conference (TREC 2013) Proceedings.

[2] Zhang, S. and Yang, H. 2013. Applying the Query Change Retrieval Model on Session Search-Georgetown at TREC 2013 Session Track. *The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings*. 2013.

[3] Jiang, J. and Dai H. 2013. Pitt at TREC 2013:

Different Effects of Click- through and Past Queries on Whole-session Search Performance. *The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings.*

[4] Strohman, Trevor and Donald Metzler, Howard Turtle and W. Bruce Croft. (2004). Indri: A Language Model-based Search Engine for Complex Queries. *Proceedings of the International conference on Intelligence Analysis*.

[5] Small, S. and Strzalkowski, T. 2004. HITIQA: A Data Driven Approach to Interactive Analytical Question Answering.

[6] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf