VIRLab: A Web-based Virtual Lab for Learning and Studying Information Retrieval Models

Hui Fang, Hao Wu, Peilin Yang
University of Delaware
Newark, DE, USA
{hfang, haow, franklyn}@udel.edu

ChengXiang Zhai
University of Illinois at Urbana-Champaign
Urbana, IL, USA
czhai@illinois.edu

ABSTRACT

In this paper, we describe *VIRLab*, a novel web-based virtual laboratory for Information Retrieval (IR). Unlike existing command line based IR toolkits, the *VIRLab* system provides a more interactive tool that enables easy implementation of retrieval functions with only a few lines of codes, simplified evaluation process over multiple data sets and parameter settings and straightforward result analysis interface through operational search engines and pair-wise comparisons. These features make *VIRLab* a unique and novel tool that can help teaching IR models, improving the productivity for doing IR model research, as well as promoting controlled experimental study of IR models.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: virtual lab; IR models; teaching

1. INTRODUCTION

Information Retrieval (IR) models determine how to compute the relevance score of a document for a given query, thus directly affecting the search accuracy. Developing optimal IR models has been one of the most important research problems in information retrieval. Over the past decades, many retrieval models have been proposed and studied. However, there is no single winner, and we end up having a few different retrieval models that all seem to perform equally well [2]. Moreover, it has proven very hard to further improve these state of the art retrieval models [1]. Unfortunately, experimenting with any new retrieval model is inevitably time consuming and requires significant amount of resources available since the new model needs to be empirically evaluated and validated over as many data sets as possible. Thus, it would be necessary to develop a tool that facilitates the development and study of IR models.

IR toolkits such as Lemur, Terrier and Lucene have been developed to enable IR research and successful technology transfer of IR models to industry and various applications. They are often designed as off-the-shelf systems that allow users to take advantage of the implemented retrieval functions to their own applications. In particular, users can use command lines to build indexes, retrieve documents based on a specified retrieval function, and evaluate the retrieval results.

Unfortunately, existing IR toolkits do not offer an easy solution for users to implement, evaluate and analyze new retrieval models, which often pose unnecessary extra burdens to the users. For example, users need to learn how to access various term or document statistic information from the index, and read through source codes or API documentations to figure out how to implement a new retrieval function. After implementing the function, the users often need to write their own scripts to evaluate and analyze the search results over multiple collections. This kind of process adds unnecessary burden to the users and could discourage them from experimenting with more functions over more collections. Naturally, it is also very challenging to use such toolkits for a course assignment.

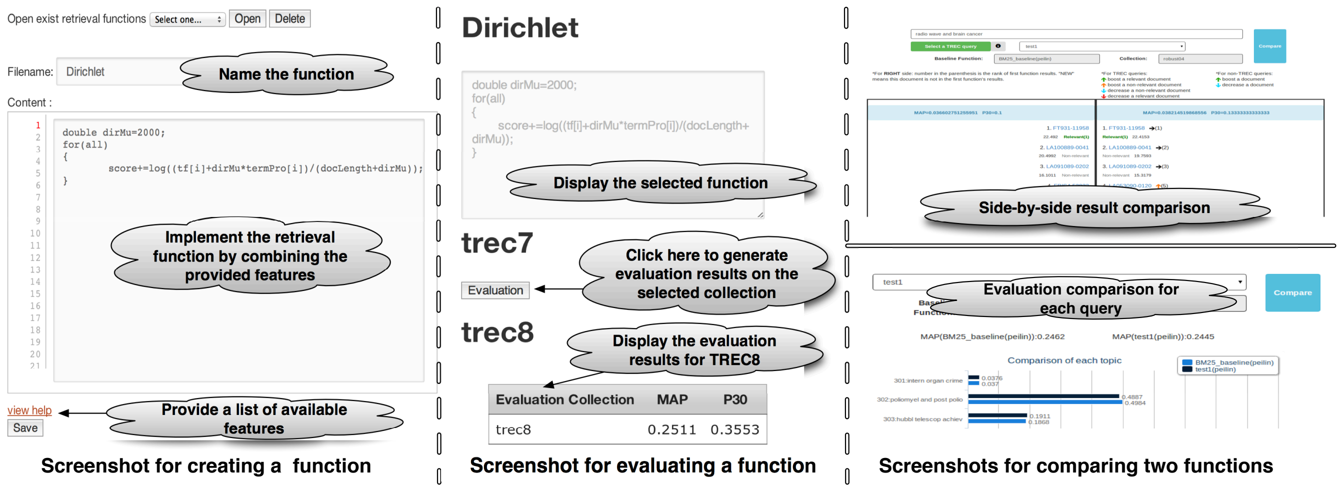
This paper describes our efforts on developing a web-based tool for IR students or researchers to study retrieval functions in a more interactive and cost-effective way. We will demonstrate that the developed system, i.e,. *Virtual IR Lab (VIRLab)*, can offer the following new functionalities:

- *Easy implementation of retrieval functions:* Users only need to write a few lines of code through a Web form to combine statistics retrieved from the indexes without worrying about how to access the indexes. The code will be automatically checked for syntax errors and translated to an executable, which will be used for ranking documents, by a dynamic code generator.

- *Flexible configuration of search engines:* Users can configure a search engine by selecting a retrieval function and a test collection. Multiple search engines can be easily created at the same time. The users can either submit their own queries or select queries from a set of topics associated with the corresponding document collection. Moreover, the users can also compare the search results of two search engines side by side to figure out their ranking differences.

- *Tight connections among implementation, evaluation and result analysis:* After creating a retrieval function, the users can evaluate its effectiveness over a few provided test collections by simply clicking a button.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
ACM 978-1-4503-2257-7/14/07.
http://dx.doi.org/10.1145/2600428.2611178.



Open exist retrieval functions
Select one...
Open
Delete
Filename:
Dirichlet
Content :
Name the function
Implement the retrieval function by combining the provided features
view help
Save
Provide a list of available features
Dirichlet
Display the selected function
trec7
Evaluation
Click here to generate evaluation results on the selected collection
trec8
Display the evaluation results for TREC8
radio wave and brain cancer
Select a TREC query
test1
Compare
Baseline Function:
BM25_baseline(peilin)
Collection:
robust04
Side-by-side result comparison
test1
Compare
Evaluation comparison for each query
Comparison of each topic
BM25_baseline(peilin)
test1(peilin)
301:intern organ crime
302:poliomyel and post polio
303:hubbl telescop achiev
0.0376
0.037
0.4887
0.4984
0.1911
0.1868

```
double dirMu=2000;
for(all)
{
        score+=log((tf[i]+dirMu*termPro[i])/(docLength+dirMu));
}
```

Screenshot for creating a function

```
double dirMu=2000;
for(all)
{
        score+=log((tf[i]+dirMu*termPro[i])/(docLength+
dirMu));
}
```

Evaluation Collection	MAP	P30
trec8	0.2511	0.3553

Screenshot for evaluating a function

*For **RIGHT** side: number in the parenthesis is the rank of first function results. "NEW" means this document is not in the first function's results.

*For TREC queries:
↑ boost a relevant document
↑ boost a non-relevant document
↓ decrease a non-relevant document
↓ decrease a relevant document

*For non-TREC queries:
↑ boost a document
↓ decrease a document

MAP=0.036602751255951 P30=0.1	MAP=0.038214519868556 P30=0.133333333333333
1. FT931-11958 22.492 Relevant(1)	1. FT931-11958 →(1) Relevant(1) 22.4153
2. LA100889-0041 20.4992 Non-relevant	2. LA100889-0041 →(2) Non-relevant 19.7593
3. LA091089-0202 16.1011 Non-relevant	3. LA091089-0202 →(3) Non-relevant 15.3179
4. [illegible]	4. LA053090-0120 ↑(5)

MAP(BM25_baseline(peilin)):0.2462 MAP(test1(peilin)):0.2445

Screenshots for comparing two functions

Figure 1: Screenshots of function creation (left), function evaluation (center) and function comparison (right)

If a retrieval function contains multiple parameter values, the users may select to evaluate all of them. If a search engine is configured using an existing test collection with relevance judgments, the official queries and judgments will be displayed so that the users can easily analyze the search results to figure out when the search engine fails and why.

- *Performance comparison through leader-boards:* A leader-board is created for each collection so that the most effective 10 retrieval functions are displayed. Users can see how their retrieval functions are compared with others, and they can also leverage the comparison functionality described earlier to figure out how to revise their retrieval functions to improve the performance.

Empowered by these new functionalities, the *VIRLab* system is a novel IR tool that can (1) help teaching IR models to students with limited programming experience; (2) improve the productivity for doing research on IR models; and (3) promote controlled experimental study of IR models by establishing baselines on various data collections.

Our prototype system is available at `http://infolab.ece.udel.edu:8008`. Please contact the authors to obtain the login information.

2. SYSTEM OVERVIEW

Figure 1 shows the screenshots of three major functionalities including creating a retrieval function, evaluating the function and comparing the results of two functions. We now provide more details about these functionalities.

The front end of the system is a Web interface that allows users to create retrieval functions. Specifically, a user can implement a retrieval function by simply combining multiple features (i.e., collection statistics) from a provided list based on C/C++ syntax. As an example, the left part of Figure 1 shows how the Dirichlet prior retrieval function [2] is implemented. Moreover, instead of specifying a single parameter value, the users can also specify a set of values for retrieval parameters, and then the system will automatically create a group of functions with these parameter settings. Once a retrieval function has been created, the user can select test collections and evaluate the effectiveness of the retrieval function over the collections (as shown in the middle part of Figure 1).

The front end also enables users to use or evaluate the retrieval function through a Web-based search interface. The user first needs to create a search engine by selecting a retrieval function and a document collection. After that, the user can either enter his or her own query or select a query from existing test collections when queries are available. If the query is from the test collections, we will display not only search results but also the relevance judgment of these results as well as the evaluation results for the query. This feature would allow users to easily see when their search engines fail or succeed and encourage them to identify the problems and try to fix them by changing the retrieval function. Moreover, we also empower users to compare the search results of two search engines side by side so that they could analyze them and identify how to revise one of the search engines accordingly. The right part of Figure 1 shows the screenshot of this functionality.

To promote controlled experimental study of IR, we generate a leader-board to report the best performed retrieval functions for each collection. Users can compare the results of the best system with their own retrieval functions through both side-by-side search result comparison and quantitative evaluation comparison.

The back end of the system includes several basic components such as indexer, ranker and evaluation script. The indexing process is done offline. Several standard TREC ad hoc collections have been indexed and ready for users to choose from. The ranker is determined by the retrieval function that the user provided through the front end.

3. DEMO PLAN AND FUTURE WORK

We expect to demonstrate all these functions of VIRLab, particularly, how one can easily (1) modify/implement a retrieval function and immediately evaluate its performance on the fly; (2) create an operational search engine with the implemented function with one click; and (3) compare the search results of two retrieval functions side by side to analyze their relative weaknesses and strengths.

We plan to extend the developed system by enabling flexible implementations of other system components.

4. REFERENCES

[1] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proceedings of CIKM'09*, 2009.

[2] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of SIGIR'04*, 2004.