

Information Retrieval and Interaction System (IRIS): A Toolkit for Investigating Information Retrieval and Interaction Activities

Jonathan Pulliza and Chirag Shah
School of Communication and Information
Rutgers University
{jonathan.pulliza, chirags}@rutgers.edu

ABSTRACT

In this demo we present IRIS, an open-source framework that provides a set of simple and modular document operators that can be combined in various ways to create more interesting and advanced functionality otherwise unavailable during most information search sessions. Those functionalities include summarization, ranking, filtering and query. The goal is to support users looking for, collecting, and synthesizing information. The system is also easily extendable, allowing for customized functionality for users during information sessions and researchers studying higher levels of abstraction for information retrieval. The demo shows the front end interactions using a browser plug-in that offers new interactions with documents during search sessions, as well as the back-end components driving the system.

KEYWORDS

Interactive search; Exploratory search; Information synthesis; Sensemaking;

ACM Reference format:

Jonathan Pulliza and Chirag Shah. 2018. Information Retrieval and Interaction System (IRIS): A toolkit for investigating information retrieval and interaction activities. In *Proceedings of 2018 Conference on Human Information Interaction & Retrieval, New Brunswick, NJ, USA, March 11-15, 2018, New Brunswick, NJ, USA*. ACM, NY, NY, USA, 3 pages. DOI: <https://doi.org/10.1145/3176349.3176895>

1 INTRODUCTION

The Information Retrieval and Interaction System (IRIS) is built with the understanding that simple and modular functions, which we refer to as operators, can be combined in various ways to create more interesting and advanced functionality for users

to explore documents. Each operator performs a specific function on a single document or a created collection, including ranking, clustering, and summarization. By building with modularity in mind, the system encourages the development of new operators that then can be easily integrated into the system, further expanding the complexity of possible requests and outcomes. Front-end users can access this functionality through a browser plug-in, which extends possible interactions with documents in real-time, or by making requests to a publicly hosted API. The complete back-end code and requirements are available in a public repository, which allows those interested to quickly host their own local IRIS instance for further exploration, as well as encouraging further development from the information science research community. The open-source project has been available to the community since 2013 and has served as a component of the Coagmento system, a platform for collaborative information seeking [1, 2, 3].

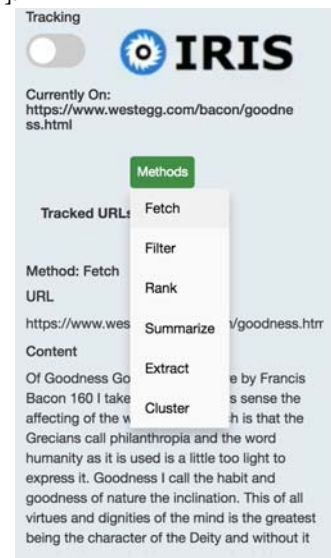


Figure 1 IRIS web browser extension sidebar displaying possible user methods for transforming documents

2 WEB BROWSER PLUG-IN

The IRIS web browser plug-in is designed to provide increased functionality as users try to make sense out of information they encounter on the Web. It monitors users'

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHIIR'18, March 11-15, 2018, New Brunswick, NJ, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-4925-3...\$15.00

DOI: <https://doi.org/10.1145/3176349.3176895>

current activity and keeps track of documents that users collect, which then become the basis for the various operators available through IRIS. The sidebar gives users a snapshot of the current state of their IRIS session, including information on the current webpage as well as documents saved to a Saved Stack. Users can then use the available operators to explore these documents in ways usually unavailable to them, including extracting key terms and summarizing documents into a requested number of sentences, as shown in Figure 1. The browser plug-in is also context-aware, so in the case of search result pages IRIS can be called on to cluster the links together sharing common content and themes. Users can then make judgments on similar documents as a group and move forward appropriately instead of judging links individually in a linear manner. Selected documents can then be added to a Saved Stack where other operators can be used, such as extracting certain terms or ranking documents.

3 ACCESSING THE IRIS PUBLIC API

A publicly hosted instance of IRIS is available for those who wish to access the API programmatically through requests. Users can then send properly formatted XML as laid out by the API guidelines, providing the appropriate documents or URLs to be processed by the indicated operators on the IRIS server. All IRIS operators are available to users making requests through the API, with the added benefit of the UNIX-like pipe feeding the output of one command into the input of another. Conditional requests can also be handled through the use of IF/Then/Else tags, further expanding the possible complexity of output. The request is then processed and returned as an XML object defined by the API guide. This allows users the ability to create and integrate complex workflows into their projects that expand beyond document retrieval to an almost unlimited amount of operations. An example XML response to summarize the content of two webpages is shown graphically in Figure 2.

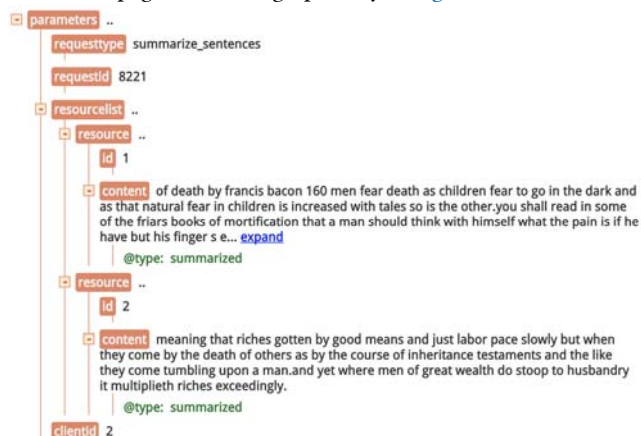


Figure 2 Example XML request for IRIS to summarize the content of a set of webpages

An example user scenario is a Ph.D. student writing a thesis and who would like to utilize previously written essays, but is

unsure of which are relevant to the thesis. Using a current draft of the thesis, the user wants to select the old essays that are the most relevant to the thesis topic. The user begins by utilizing the “extract” operator to find the most frequent words in the current thesis, with the assumption that these relate to the most important concepts. These terms then become the basis for ranking the old essays regarding their similarity to the thesis. A “rank” request is made on the old essays in respect to the extracted keywords, and the result is a ranked list indicating which essays are the most relevant to the user’s thesis. From a collection of documents the user can select the most relevant based on a single target document, and can now further query and manipulate the resulting documents as needed.

Another example outside of academia is a user that is a parent of a 7-year old daughter who wants to learn about sharks, specifically pertaining to their body size and speed. The user collects a list of URLs that relate to sharks and instead of going through each URL, the user decides to upload the URLs to IRIS and query for the terms “body” and “speed.” IRIS returns the documents with those terms, but the user is looking for just the relevant blocks of texts related to body and speed. The user can then request to extract the blocks of text from these documents related only to shark bodies and speed, which the parent can now quickly read over or further process and manipulate. From a list of raw URLs IRIS can extract and transform relevant information into the desired format.

4 HOSTING AN IRIS INSTANCE

The back-end of IRIS is comprised of a PHP server connected to a MySQL database, with queries handled through the Indri Query Language. All of the text processing extensions which correspond to operators in IRIS are written and compiled in C and then called by the PHP instance when necessary. Adding a new operator then means simply creating a new file in the Controllers directory and adding that operator to the top-level PHP config file. This design not only makes adding new operators straightforward but allows for previous extensions to be easily augmented and improved as better techniques and algorithms become available.

To further encourage community contributions to the project the entire IRIS codebase is available GitHub as an open source tool. The repository also contains a guide to the requirements necessary for setting up a local instance of IRIS, as well as example requests and responses. We will continue to host our own instance of IRIS open to the public for API use, but we hope researchers will explore their own custom instances for experimentation and user studies. We hope this will lead to an active community dedicated to IRIS development and further exploration of information abstraction and interaction.

5 DEMONSTRATION

In this demo, we will show some of the front-end functionality of IRIS through the web browser plug-in as well as through API requests. This will include going through some example user sessions and possible outcomes with the assistance

of IRIS. Next, we will show how to install and run a local instance of IRIS locally, as well as creating and utilizing an example custom operator.

ACKNOWLEDGEMENT

This work is supported by the [National Science Foundation \(NSF\)](#) grant [IIS-1717488](#).

REFERENCES

- X. Fu, D. Kelly, and C. Shah. Using collaborative queries to improve retrieval for difficult topics. In *Proceedings of ACM SIGIR '07*, New York, NY, USA, 2007. ACM.
- [1] C. Shah, G. Marchionini, and D. Kelly. Learning design principles for a collaborative information seeking system. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09*, pages 3419–3424, New York, NY, USA, 2009. ACM.
- [2] M. Mitsui, C. Shah. Coagmento 2.0: A system for capturing individual and group information seeking behavior. In *Digital Libraries (JCDL)*, 2016 IEEE/ACM Joint Conference on (pp. 233-234). IEEE.
- [3]