Wikoid: An Interactive Tool for Exploring and Searching Wikipedia Documents

Kenneth Emeka Odoh [200334078]

October 16, 2013

1 Motivation

Searching of online documents is an integral part of modern life. Wikipedia is among the largest collection of online documents on the internet. The future success of Wikipedia is dependent on the ability of users to quickly find relevant articles. Users are impatient when the returned results are not relevant to the search term. The search infrastructure of Wikipedia is optimized for well-crafted search queries. User tends to give short query that is lacking in descriptive information [1]. Moreover, in some cases the user may not have an idea of what the appropriate answer may be before making the query. We need to engage the users in recognition rather than recall.

Wikipedia search interface does not have support for exploratory search. The current Wikipedia page provides hyperlinks in the body of document that refers to other pages that may help the user understand the current web page. The lack of visualization of the relevance of the keyword to the search result affects user experience when they are doing specialized searching.

Exploratory search has the following interaction types which includes conversing, manipulating (navigating), exploring (browsing) [2]. This allows the user to get more interaction beyond matching search query to relevant document. Exploratory search does not work by returning irrelevant result and allowing the user to explore endless amount of unrelated pages. This is undesirable from the end user's perspective as it degrades user experience. The addition of exploratory search possibilities is a complementary features and would not replaced the current system for quick fact verification [3].

The current Wikipedia search interface is ideal for fact verification. However, users are subjected information overload on the internet and hence need to compare several results by recognition. Exploratory search is the ideal solution for this problem. This approach is based on recognition (instead of recall). We want to use the visual abilities of the user by allowing the user explore through the search space using visual clues. The visualization of the semantic association among terms can help the user understand relationship among keywords. My graduate research work is focused on exploratory search. We are working on improvement of the the search interface of Wikipedia to enable exploratory search. The current nature of Wikipedia interface has motivated me to research better ways of improving Wikipedia's search interface.

2 Related Work

Most of the earlier search interfaces work by returning search results with the highest relevance to the search string [4]. These systems perform well when the query strings are well constructed. The search engine performance degrades if the string are short and undescriptive. According to a survey reported in the paper [5] which shows that a majority of search query term are between one and three terms. This clearly shows the scale of the problem.

There are several exploratory search engines. They include Wordbars [5] and Hotmap [6]. Interaction design is the core of exploratory search as people want to be guided to the relevant document using visual clues in their everyday working life. Web search can classified as a information retrieval system because of the following features. They includes [7]:

- Document source (e.g. Wikipedia document database).
- Search interface (i.e. mechanism for querying document source.)

The interface to the user is the search box. Many search engine makes use of simple search text box to accept user input. Exploratory search feature can be implemented as a middle layer between the users making query and receiving the search results [7].

3 Method

The current implementation borrows heavily from the open source project named Microsearch [8]. This open source project was aimed at searching within a desktop or intranet. I had to change the architecture and designed a software system that supports

web search. The prototype software implementation of the project was developed over two weeks (from 20th September, 2013). The project was named 'Wikoid'. The goal is to improve the current search interface to support exploratory search by building a thin wrapper around Wikipedia.

This layer of abstraction accepts search queries from the user on on behalf of Wikipedia and sends the request to Wikipedia. The result is obtained by scrapping the Wikipedia page matching the search criteria and returning it to the user in an interface that has constrained the user to the specific item (in accordance with Norman's design principle). The returned page is crawled for possible Uniform Resource Locator (URL) link. This is because according to Gestalt rule of proximity which states that items that are closer are similar [9]. The aim is to recommend URLS of high relevance to the user. The mode of operation can be split into the following. They include:

- Internal
- External

3.0.1 INTERNAL

As pages are retrieved in relation to the search query, the content of first returned page are parsed to check for hyperlinks. The most informative section of the document is tokenized into terms and stemmed to reduce to root form. The terms are used to form the inverted index which is the mapping of frequency of terms(keyword) to documents. This is done to enable fast lookup.

3.0.2 EXTERNAL

This includes the user interface for accepting search query. The search term must undergo the same string manipulation process of tokenization and stemming.The current system is shown in the following figures. They include: Figure 1, Figure 2 and Figure 3.

3.1 PROPOSED IMPROVEMENTS TO CURRENT PROTOTYPE

The current prototype lacks support for information visualization for showing relationship between keywords. This system is slow because of computationally-intensive process of stemming, tokenization and repetitive copying of page content. These processes would be eliminated by design in the improved prototype under development. This would be improved by better software development practice where we avoid unnecessary copying by making page content's variable scope accessible across needed functions.

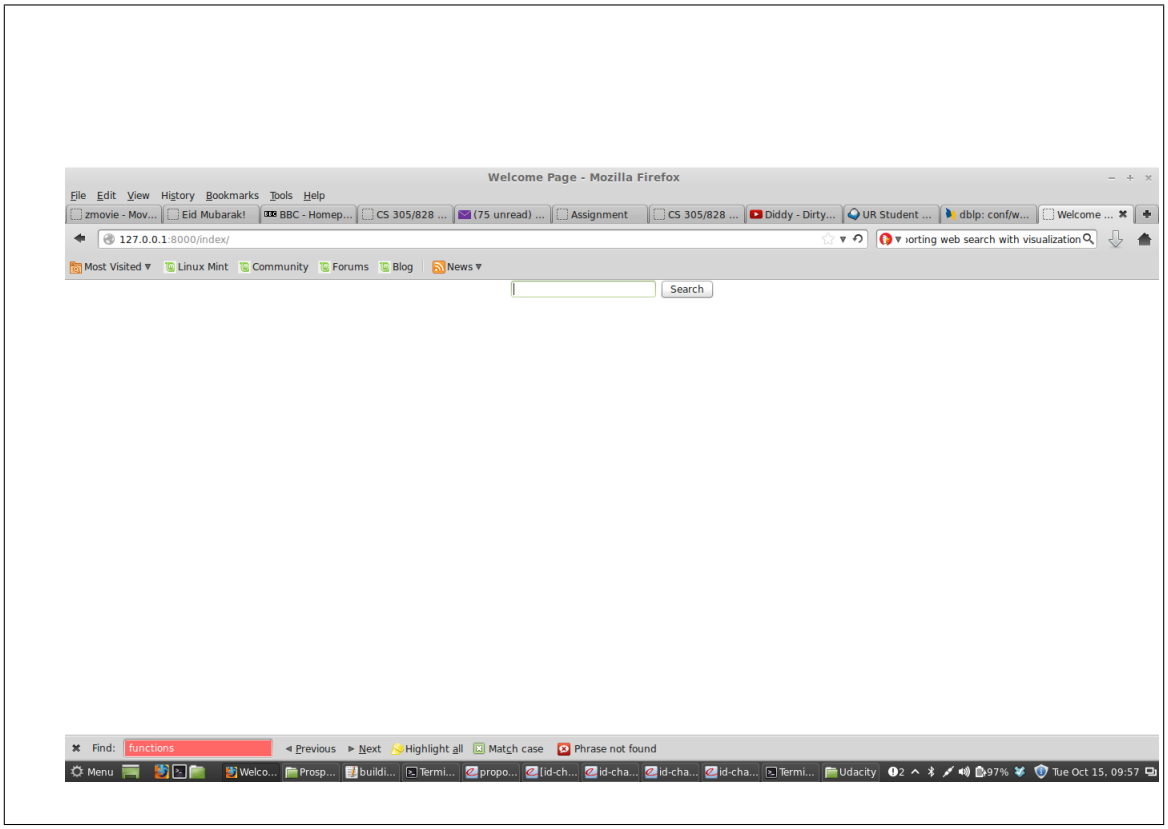

Figure 1: Front page of Wikoid (current protype).

The new system will be similar to Wikiweb [10] . The bulk of the system will be implemented in Python programming language [11]. The relationship between keywords and URLs will be represented as a graph. The graph will be visualized using the Javascript library (vivaGraph.js) [12]. The interface for the new system will be interactive as the relationship between keywords can be identified and visualized.

The keywords in a wikipedia page are highlighted. This makes it easy to obtain the keywords by applying XML related technologies (XPath) [13]. The lookup will be the same as in current version of Wikoid. This system will provide better user experience for the end users as they are able to visualize the reason for the search result that was obtained. The simplification of the search process will make the system accessible to people of all ages and background because visual has a strong effect on all peoples.

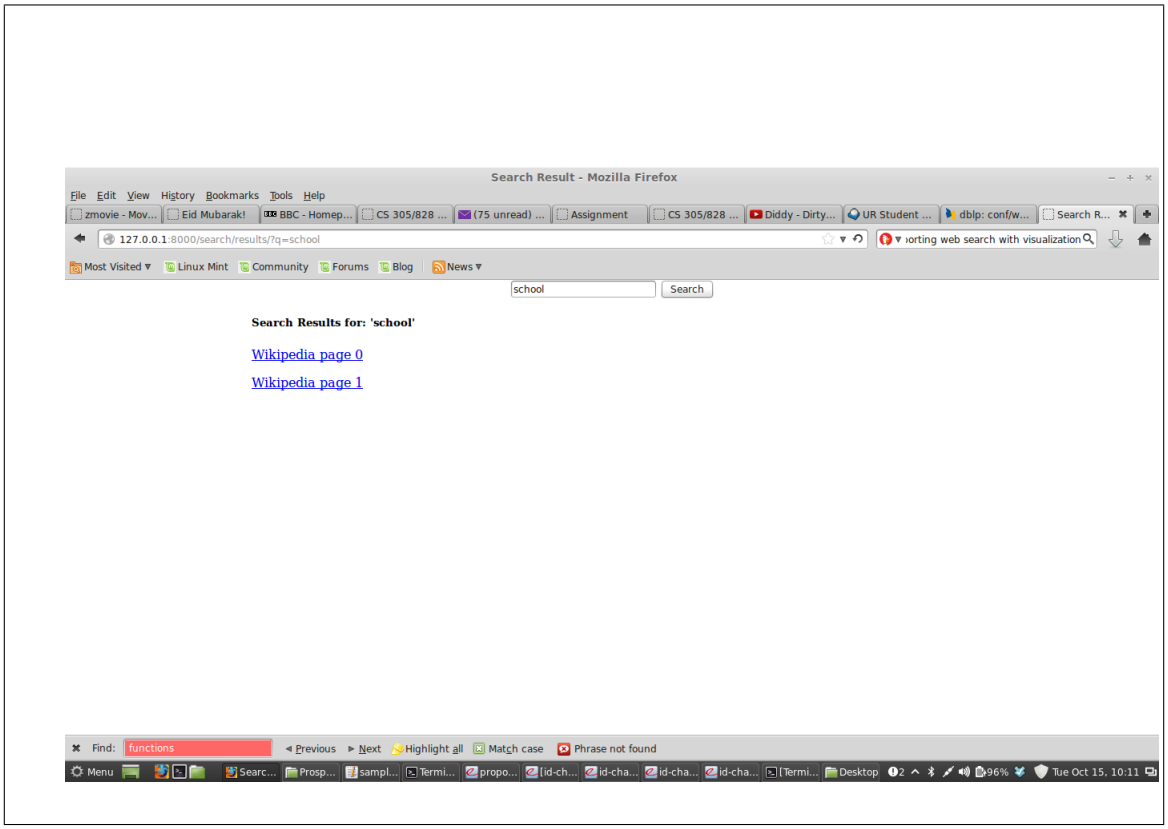

Figure 2: Second page of Wikoid (current protype).

4 EVALUATION

The system will be evaluated using the DECIDE framework [2].

4.1 DETERMINE THE GOALS

The goal is to provide exploratory search feature to wikipedia. This is a needed feature as users want to be guided to the search result especially when recognition is in use. The consistency is maintained as related Wikipedia page is returned. This user can view a constrained wikipedia page. The user can also see relationship among the keywords.

4.2 EXPLORE THE QUESTIONS

This is a complementary feature that is added as a layer on Wikipedia. This would help users who are poor in constructing query to arrive at the required document by explo-

Figure 3: Result page of Wikoid (current protype).

ration.

4.3 CHOOSE THE PARADIGM

Given the time constraint in the course and difficulty of performing usability testing.The evaluation would be based on heuristics. The approach is to use Nielsen's heuristics. The experts that are evaluating the system are two of my thesis supervisor who already know the user requirements. These experts would evaluate the performance and usability of the system. They will provide feedback in the debriefing section. We would also use walkthrough(cognitive and pluralistic) to identify usability problems

4.4 IDENTIFY PRATICAL ISSUES

These are some of the identified practical issues that can affect usability evaluation.

- Select users (This is not needed as we are using predictive models.)
- Find evaluators (My thesis supervisors will serve as evaluators)
- Select equipment (My personal computer will serve the purpose.)
- Stay on budget (zero cost project.)
- Stay on schedule (finish before the end of course.)

4.5 Decide about ethical issues

This is non-existent because privacy of user is not a problem as we are using predictive models.

4.6 Evaluate, analyze and present data

The project will be evaluated holistically and findings will be reported in the final project documentation.

References

[1] Taraneh Khazaei and Orland Hoeber. Metadata visualization of scholarly search results: supporting exploration and discovery. In *I-KNOW*, page 21, 2012.

[2] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction Design: Beyond Human-Computer Interaction* . JohnWiley & Sons Limited., 2011.

[3] Orland Hoeber and Xue Dong Yang. Hotmap: Supporting visual exploration of web search results. *JASIST*, 60(1):90–110, 2009.

[4] Orland Hoeber. Exploring web search results by visually specifying utility functions. In *Web Intelligence*, pages 650–654, 2007.

[5] Orland Hoeber and Xue Dong Yang. Evaluating wordbars in exploratory web search scenarios. *Inf. Process. Manage.*, 44(2):485–510, 2008.

[6] Orland Hoeber. A longitudinal study of hotmap web search. *Online Information Review*, 37(2):252–267, 2013.

[7] Orland Hoeber and XueDong Yang. Supporting web search with visualization. In JingTao Yao, editor, *Web-based Support Systems*, Advanced Information and Knowledge Processing, pages 183–214. Springer London, 2010.

[8] Microsearch. `https://github.com/toastdriven/microsearch`. Date accessed: October 15, 2013.

[9] Gestalt laws of perceptual organization. `http://psychology.about.com/od/sensationandperception/ss/gestaltlaws_4.htm`. Date accessed: October 15, 2013.

[10] Wikiweb. http://www.wikiwebapp.com/. Date accessed: October 15, 2013.

[11] Python programming. http://www.python.org/. Date accessed: October 15, 2013.

[12] Graph visualization. https://github.com/anvaka/VivaGraphJS. Date accessed: October 15, 2013.

[13] Xpath. http://www.w3.org/TR/xpath/. Date accessed: October 15, 2013.