Recommendation Algorithm on Academic Research

Interim Report

Lui Hoi Ching (Vincent)
Chan Erik
Supervisor: Dr. Yiu Siu Ming
Table of Content

Table of Content ........................................................................................................2
Abstract ...................................................................................................................... 3
Acknowledgement .................................................................................................... 3
Abbreviations .......................................................................................................... 3
Background .............................................................................................................. 4
Objectives ................................................................................................................ 6
Project Status .......................................................................................................... 7
Schedule ................................................................................................................... 13
Summary ................................................................................................................... 14
References ............................................................................................................... 15
List of Figures ......................................................................................................... 15
List of Tables .......................................................................................................... 15
Appendix ................................................................................................................ 16
Abstract

In this project, an algorithm that recommends academic research resources is to be implemented. In particular, a server-side database structure for citation storage is designed and citations are generated automatically for target resources. We also discuss the idea of combining existing keyword generation algorithms with knowledge based search in generating abstract keywords for recommendation.

Acknowledgement

We would like to express our appreciation to Dr. Yiu Siu Ming, our supervisor in the project, for his time and effort in guiding us in overcoming many difficulties and discovering new areas that we had not explored. Furthermore we would also like to acknowledge with much appreciation the role of Mr. Patrick Desloge, who gave guidance to our project presentation and the writing of this report that has improved our presentation and writing skills thanks to his comment and advices.

Abbreviations

CSL - Citation Style Language
NLP - Natural Language Processing
Background

Since the invention of the World Wide Web, doing good research on the Internet is an essential part in writing any essays and articles as information on the Internet has a wide coverage on different topics. Several difficulties arises when writing essays. Writers often need to spend a lot of time on searching and filtering for relevant resources on their research topic. Google search and Google Scholar are very popular tools for search for academic research resources. They can significantly reduce searching time once the user had found the right keyword to search for. Despite that, time is still spent on scanning the articles and filtering useful texts. Second, while technological advancements had allowed us to write and do research with the same device, the process of researching and writing had been largely separated. Integrating the research process into the writing had become a time consuming task. Tools like Zotero and Endnote exist to help manage the texts which users find and store from websites. They also provide auto-generated citations. However, these tools do not integrate the process of searching and writing as searching is still largely separate from the writing workflow. Developing software with algorithms that can automatically understands your current writing and provide adequate resources on the Internet can greatly reduce the time required on researching and integrating the research with the writing.

There are several popular tools for academic research, including Google Scholar, Zotero and EndNote. Table 1 shows a summary of advantages and disadvantages of each of the tools. Google scholar is one of the most popular searching tools. It can rank research papers so that users’ searching time is greatly reduced. However, users still need to scan the whole article to determine if it is useful and locate the useful texts. Moreover, the writing and research process is separated. The general approach by most people is to use Google Scholar to search for relevant resources, open new tab for each web page, and then write their essays. This approach has a big disadvantage: writers have to spent time on revising their materials again during writing. Even if Google Scholar is combined with Zotero and EndNote, the above two problems still cannot be solved.
<table>
<thead>
<tr>
<th>Tools</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Scholar</td>
<td>• Search for academic research resources quickly</td>
<td>• Lack a tool to highlight and store the relevant research resources found</td>
</tr>
<tr>
<td></td>
<td>• Search results are ranked by number of citations, helping users get the most relevant resources</td>
<td>• Lack a suggestion of text. Users need to scan the whole article to get useful text and see whether the result is appropriate</td>
</tr>
<tr>
<td>Zotero</td>
<td>• Automatically senses the content in a web browser and allows users to store the content</td>
<td>• Lack a search function to search for relevant resources</td>
</tr>
<tr>
<td>EndNote</td>
<td>• Manage PDF academic research resources</td>
<td>• Lack a method to rank resources</td>
</tr>
<tr>
<td></td>
<td>• Search research resources in PDF form in online database</td>
<td>• Lack searching of internet materials</td>
</tr>
<tr>
<td></td>
<td>• Users can look at comment made on PDF online</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of current academic research tools

The deliverable of this project will be an integration of functions provided by the above mentioned tools, together with functions that recommend relevant texts and integrate writing and searching process. Table 2 shows a comparison of the deliverable and other research tools. The core advantage of the deliverable is that it reduces essay writing time significantly by recommending most relevant resources and texts and this function is not in any of the other research tools.

This project will be built on top of a developing web application named GistNote which provides support of highlighting and storing websites’ texts. At this stage, GistNote is capable of allowing users to highlight any texts on a website. The highlighted texts are stored and can later be referred when users write essays. With the highlight data from users, it is possible to allow the searching of database to find relevant resources.
<table>
<thead>
<tr>
<th></th>
<th>Google/Google Scholar</th>
<th>Zotero</th>
<th>EndNote</th>
<th>Our deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search for relevant website</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Search for research paper</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Search results are ranked</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Auto-generated citations</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow users to store useful results</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Recommend relevant text</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Recommend resources when writing</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2: Comparison of project deliverable and other research tools

**Objectives**

This project is going to achieve five things.

1. Structuring a database to store users’ highlights for searching.
2. Developing an algorithm or integrating existing algorithms to search for relevant recommendations on research resources from keywords.
3. Ranking existing cited research resources by an algorithm.
4. Developing an algorithm to understand users’ essay and recommend resources accordingly automatically.
5. Generating citations in different styles.

These algorithms and functions are mainly developed for use in a web application named GistNote. The integration of the algorithms and the user-interface of the extension depend on the development progress of GistNote (not included in this project)
Project Status

- Implementation of the above mentioned algorithms in executable form(s)
- Recommendation generation requires the program to understand the context of the current writing. Algorithms that tries to achieve this falls into the field of NLP. More specifically, automatic summarization and keyword extraction or generation given a certain length of text is the tasks most relevant to the project. The current technology in NLP is still in development, and most solutions involve the use of machine learning. At the current stage of the project, we have carried out a preliminary testing of implementations of tweaked version of existing TextRank libraries. TextRank is an unsupervised machine learning algorithm derived from Google search ranking algorithm PageRank. (figure 1) TextRank utilize the co-occurrence of words as edges between nodes of words (figure 2).
Comparison between different approaches to the above problem

GitHub repository search was utilized by inputting the keyword “keywords extraction”, resulting in 141 search results. In order to filter out less popular solutions, the search result was sorted by descending order of the number of stars. The top results were examined and five repositories were found to be the most relevant. These five repositories will then be studied in two main categories, namely accuracy and efficiency.

Accuracy will be analyzed by two properties of the extraction algorithm, namely precision and recall, demonstrated in figure 3. They will also be evaluated by the F-score, the harmonic mean of the two.

\[
F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.
\]
Figure 3: Graphical representation of the concept of precision and recall

- **Automatic citation generation**
  - The critical part of automatic generation of citations and bibliography is to obtain the formatting of the target citation style. A cloud-based platform that store different citation styles is preferred. Based on this criteria, Citation Style Language (CSL) is chosen as the platform for the source of citation style.[3] CSL is a open XML-based language to describe the formatting of citations and bibliography. It has many advantages. Firstly, It maintains over 8000 citation styles in the GitHub repository so that addition and
modification of the formatting are convenient. The most popular citation style can be obtained easily and this already fits the scope of the project. Secondly, CSL is currently used by many other research tools, in particular, Zotero. Zotero even developed its own repository to manage the CSL file. It is seen that CSL is reliable to be used.

- There already exist open-source processors that implement CSL. Since a web-based text editor is to be developed in GistNote, a javascript version (citeproc-js) is chosen.[4] It is already well-developed with powerful features like dynamic changing of citation styles so it can reduce the development effort of citation generation feature in this project. The basic logic of generating citations and bibliographies is finished. The engine can generate citations with appropriate inputs, which are specified in the CSL files.

- **Database Structure**
  
- While a client-side database is currently used by GistNote to store information of websites and highlights, a server-side database is planned to be built to facilitate searching of highlights which are grouped by different websites. The server-side database structure is designed and figure 4 shows an entity-relationship diagram of the server-side database structure. It consists of two entity, websites and highlights, in which some useful attributes are stored. Each highlight is associated with one website so that all highlights in one single website can be obtained in the most convenient way. This can help the project engine get the important text in a website more efficiently.
- A web application with integrated functions utilizing the above mentioned algorithms (if possible)
  - Updates on this sub-deliverable is expected to arrive in 1-2 months.

**Difficulties and Limitations**

- This project chose a javascript engine to generate citations automatically. Although the engine chosen can generate citations automatically, it is implemented in a complicated way that requires its user to manage the citation themselves. It becomes difficult to add more powerful features because this requires more effort on understanding the code and programming. As a result, it is planned to include only
the basic function of generating citations and bibliographies for a single piece of resource.

- The preliminary testing of implementations of tweaked version of existing TextRank libraries during an early stage of the project generate unfavorable results (see figure 5). This result stresses the importance of better heuristics [1] and post-processing, including the yet to be implemented knowledge based search in generating more abstract keywords. Further debugging is also required and planned in order to remove the possibility of a faulty implementation. We strive to improve the algorithm in the coming stages of the project, as our latest progress shown in figure 6 generates much more precise keywords, albeit with a lower recall. There is also a bias in returning unigrams as keywords instead of bigrams similar to our preliminary test. This problem should be fixed in a later revision of the algorithm.

Figure 5: Unimportant keywords “original” and “google” with a high ranking

Figure 6: Our latest progress in improving the algorithm
<table>
<thead>
<tr>
<th>Date Range</th>
<th>Task Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Sep - 15 Sep</td>
<td>Research on algorithm and feasibility studies</td>
<td>Completed</td>
</tr>
<tr>
<td>16 Sep - 30 Sep</td>
<td>Project planning and revising specifications</td>
<td>Completed, may be revised in a later stage</td>
</tr>
<tr>
<td>1 Oct - 5 Oct</td>
<td>Design the structure of database that meets the needs of recommendation algorithm.</td>
<td>Completed, may be revised in a later stage</td>
</tr>
<tr>
<td>6 Oct - 31 Oct</td>
<td>Implement the database structure and integrate into the original application. Data will be stored in the way we structured and can now be gathered from users. (Vincent)</td>
<td>In progress</td>
</tr>
<tr>
<td>1 Oct - 20 Oct</td>
<td>Implementation of auto-generated citations (Vincent)</td>
<td>In progress</td>
</tr>
<tr>
<td>21 Oct - 31 Oct</td>
<td>Integrate auto-generated citations to the original application (Erik)</td>
<td>Re-scheduled to a later stage</td>
</tr>
<tr>
<td>1 Nov - 31 Dec</td>
<td>Comprehension of text - get the keywords and their synonyms from texts. (Erik and Vincent)</td>
<td>Started preliminary implementations and comparisons ahead of schedule in order to reduce probability of project failure</td>
</tr>
<tr>
<td>1 Jan - 20 Feb</td>
<td>Searching websites (Vincent)</td>
<td>On schedule</td>
</tr>
<tr>
<td>1 Jan - 20 Feb</td>
<td>Ranking websites (Erik)</td>
<td>On schedule</td>
</tr>
<tr>
<td>21 Feb - 31 Mar</td>
<td>Searching and ranking texts in the same website (Vincent)</td>
<td>On schedule</td>
</tr>
<tr>
<td>21 Feb - 31 Mar</td>
<td>Comprehension of users’ essay, and recommend users on research resources. (Erik)</td>
<td>On schedule</td>
</tr>
<tr>
<td>1st Apr - 15th Apr</td>
<td>Testing</td>
<td>On schedule</td>
</tr>
</tbody>
</table>
We have presented our current status and our different approach to tackling the recommendation problem from keyword extraction compared to other solutions in the market. While our preliminary testing on the TextRank algorithm was not satisfactory, it provided us a new opportunity to examine other natural language processing algorithms. Comparing TextRank, an unsupervised machine learning algorithm, with other supervised machine learning algorithms is possible. However, it likely would not provide valuable results as the lack of learning data may be detrimental to the accuracy of supervised learning algorithms. With more abundant data, comparisons should be carried out. Despite the lack of comparisons between these two types of learning algorithms, we are confident that our final implementation with synonym search would provide superior accuracy and efficiency.

The database structure has already been designed and the citations generation feature is done. This means the search engine can be developed in the later stage.

In contrast to our scheduled timeline, we have moved ahead on preliminary implementation of text comprehension provided that this part contains a high risk and the highest chance of failure. As we predict that citation generation is not as complex and the recommendation algorithm, the project as a whole is still progressing as expected if not better than expected.
References


(3) CitationStyles.org [Internet]. CitationStyles.org, 2016 [retrieved 30 Oct 2016]. Available from: http://citationstyles.org/


List of Figures

Figure 1. Representation of the Google PageRank algorithm Page 7
Figure 2. Example of a co-occurrence graph of a body of text Page 8
Figure 3. Graphical representation of the concept of precision and recall Page 9
Figure 4. Entity-relationship diagram of server-side database Page 11
Figure 5. Unimportant keywords “original” and “google” with a high ranking Page 12
Figure 6. Our latest progress in improving the algorithm Page 12

List of Tables

Table 1. Summary of current academic research tools Page 5
Table 2. Comparison of project deliverable and other research tools Page 6
Table 3. Project Schedule Page 13
Appendices

I. Division of work of the report

Abstract: Written by Lui Hoi Ching (Vincent)
Acknowledgement: Written by Erik Chan
Background: Written by Lui Hoi Ching (Vincent)
Objectives: Written by Erik Chan
Project Status:
  • Implementation of the above mentioned algorithms in executable form(s): Written by Erik Chan
  • Comparison between different approaches to the above problem: Written by Erik Chan
  • Automatic citation generation: Written by Lui Hoi Ching (Vincent)
  • Database Structure: Written by Lui Hoi Ching (Vincent)
  • A web application with integrated functions utilizing the above mentioned algorithms (if possible): Written by Erik Chan
Difficulties and Limitations:
  • Paragraph 1: Written by Lui Hoi Ching (Vincent)
  • Paragraph 2: Written by Erik Chan
Schedule: Co-written
Summary: Co-written