UResearcher

Liu, Christopher
Sharma, Ritesh
Tata, Micheal
Zhao, Jack
Frank, Will

Team Cool Guys
October 25th, 2019

Table of Contents

Project Name: UResearcher
Team Name: Team Cool Guys

Executive Summary

Researchers currently sift through thousands of online research papers regularly to find relevant and useful information pertaining to a particular project. Consequently, an exorbitant amount of time is spent filtering out excessive data, misleading citations, and irrelevant topics, and, sometimes, these efforts prove fruitless because of misinformation and conflicting research results. Consequently, such tedious tasks lead to frustration and impediments. UResearcher remedies these issues plaguing research by significantly reducing the search time for pertinent topics and by efficiently organizing information at search time. The application accomplishes this task by narrowing the search space while finding useful and relevant information through machine learning models.

UResearcher primarily uses unsupervised clustering through public, machine learning libraries to organize the research data. The application retrieves vector information from available abstracts; then, it produces an appropriate model from the gathered information to classify and describe future inputs. Therefore, this feature can easily organize the search space from a user-defined input, reducing not only the search time but also the processing stress. As a result, the program significantly decreases unnecessary research time.

Also, UResearcher identifies previously unidentified knowledge using methods described by researchers from Lawrence Berkeley National Laboratory. This preprocessing, known as Latent Knowledge Analysis, uses the Word2Vec and Gensim libraries to estimate and weight abstract information from papers. From this data, the application constructs a machine-learning based model, which identifies areas of interest for research or gaps in knowledge. Consequently, relevant research becomes readily available without a significant loss in information, ensuring a smooth process.

Furthermore, UResearcher analyzes and predicts trends for fruitful research opportunities by looking at proposal calls for past, present, and future funding insights. The program fetches public funding information as data points and models a predictor for future trends in research. As a result, UResearcher provides invaluable assistance for researchers by reducing menial searching and providing more time for other endeavors.

Therefore, UResearcher is an invaluable application for researchers since the included features reduce unnecessary searching while providing worthwhile outlets. Organizing topics, finding missing information, and predicting are invaluable assets for any researcher, so UResearcher is an essential tool for any research endeavor.

Background & Technical Requirements

Idea Space

The advent of the internet has made accessing academic research easier than ever. From nearly any location in the world, individuals may find research information without a loss in time or quality. In addition, research is no longer confined to isolated singular locations and institutions, so prospective researchers do not need excessive levels of clearance to access papers. However, because of the sheer volume of information available, the process of going through academic research has become even more time consuming and complex. Furthermore, with new research breakthroughs, it is no longer feasible for experts and academics to keep up with the developments in his or her respective fields. This issue also creates problems with redundant information and overlapping research. So, searching the internet for specific information regarding research becomes an arduous task fraught with pitfalls.

Currently, there are several tools that aim to help with online academic research. A few notable programs are Scholarcy, the Science Research Assistant, Mendeley, and Zotero. Scholarcy focuses on summarization and automatic highlighting of key points in an article and functions on an article to article basis. The application primarily expedites sifting through a single article and does not address the problems with searching and filtering. The Science Research Assistant aids with searching by analyzing multiple publishers at once and allows for search customization based on keywords provided by the user. Nevertheless, this application does not address redundant information and only facilitates search in a more organized manner. Mendeley, an organization and social media tool, makes citations and bibliographies easier, while offering the ability to connect with other researchers. Similarly, Zotero offers a way to save, cite, and organize research. However, both Mendeley and Zotero lack features for recommendations.

All of these programs offer features that help with one specific area of online research, but they fail to address the larger problems that make online research difficult. UResearcher addresses these issues and aims to help with multiple aspects of research. By focusing on presenting relevant information in an organized fashion, the program helps with both the scope of the search and the issues of redundant information. It also offers other analytical tools such as viewing trends in grant allocation, and keyword trends within an area of research. Consequently, UResearcher fills an essential niche in the research field by offering more useful features and providing necessary insight.

Technology

The final project contains aspects from several frameworks, libraries, and technologies. Such technologies are quite modular and well-defined, consequently, the structure of UResearcher has several components pieced together. Despite this pre-defined foundation, UResearcher uses at least one algorithm built from scratch.

UResearcher implements three key features: keyword searching, unsupervised clustering, and latent knowledge analysis. To perform searching, the program will use a keyword search algorithm to gather the initial set of articles. The application then performs unsupervised clustering to organize research articles for the user. Finally, the program performs Word2Vec using the Gensim library which will perform latent knowledge analysis within the application.

Another feature is the analysis of past grant money allocation to predict the future of such allocations. In this particular area, UResearcher uses probability and statistics to extrapolate possible graphs. Furthermore, the program accesses the grants.gov API to gather raw information regarding grants and allocations, displays that information in a friendly manner to the user with charts and graphs, and makes predictions about possible outcomes. In addition, the simplicity of this feature allows for a quick and easy implementation.

In the front end, the application uses HTML, CSS, JS/JQuery, and Bootstrap. Aside from the first few technologies, which are nearly essential, Bootstrap aids in rapid prototyping for a quick and clean implementation. Furthermore, the Bootstrap library helps provide a consistent look and feel without contracting any graphic designers.

On the backend, UResearcher uses algorithmic implementations in Python 3, so Django, a python-based web framework, serves as the standard. The long lasting and public nature of Django also yields plenty of supporting middleware packages for development. Django also supplies an MVC architecture, an important aspect to the application. Furthermore, Flask interacts with the various APIs and provides a very lightweight framework for handling requests. It is also used to talk to the University of Utah's Center for High Powered Computing, where our most intensive operations will be performed. Because SQLite is the default for Django and it is a python library, UResearcher uses SQLite for its minimal storage needs.

Hardware/Software Requirements

Because our product is delivered as a website there are virtually no hardware or software requirements. Because we will be using Bootstrap 4.3 we will be targeting, at the oldest, Chrome 45, Firefox 38, Edge 12, iOS 9, and Android 4.4, though these requirements should be very easily met. Even though this is a product rife with complex operations they will be performed primarily on the backend, so there is no high end hardware required for the user.

<u>Appendix</u>

Use-Cases

For User

Number	1
Title	Searching for a Paper or Subject
Description	A user wishes to investigate or find a specific area of research papers on the website.
Steps	1. Type a query into the search bear and click "Search." 2. Optionally use a refined search or cluster to dive deeper into the website. 3. Choose paper from the results .
Related UI	Figure 1, Figure 2, Figure 3, Figure 4

Number	2
Title	Reverting Searches and Starting a New Search
Description	A user desires to backtrack to a previous result page or start a search for a new subject space.
Steps	1. Continue to click the back button, until you get to your desired page. 2. Stop reverting after reaching the home page. 3. Use new search box contained in each page to search for new search result.
Related UI	Figure 1, Figure 2, Figure 3

Number	3
Title	Doing Latent Knowledge Analysis
Description	After doing a search, the user requests latent knowledge analysis in the results.
Steps	1. Perform a search. 2. Optionally refine the search results. 3. Select to perform latent knowledge analysis on the results. 4. View latent knowledge analysis.
Related UI	Figure 2, Figure 12

Number	4
Title	Analyzing Grant Trends
Description	After performing a search, the user wishes to analyze grant trends.
Steps	1. Perform a search. 2. Optionally refine the search results. 3. Select to perform grant trend analysis on the results. 4. View the grant trend analysis graph.
Related UI	Figure 6

Number	5
Title	Finding No Articles in an Area.
Description	The user does not find any articles with a search request.
Steps	1. Perform a search. 2. Potentially refine the search further 3. Arrive at the "No Articles Found" page. 4. Search or revert from that point.
Related UI	Figure 5

Number	6
Title	Saving Search Results
Description	The user has performed a search, potentially refined it, and wants to save a list of all the articles.
Steps	1. Perform a search 2. Click the Save Search button.
Related UI	Figure 2 - Figure 12

Number	7
Title	Error Messaging
Description	The user encounters an unforeseen error in the application.
Steps	1. Encounter the error in the application. 2. Optionally reply to a feedback system. 3. Revert or restart a search.
Related UI	Figure 11

Number	8
Title	Help, Feedback, and Documentation
Description	The user desires to learn more about the application and provide input to the developers.
Steps	1. Click on the "Help" icon on the application. 2. Choose "Help" to view directions for running the application. 3. Choose "Feedback" to send feedback to the developers. 4. Choose "Documentation" to view the documentation documents. 5. Revert or delete to exit these pages.
Related UI	Figure 1 - Figure 13

Number	9
Title	Saving Grant Analysis Results
Description	The user wishes to save either the raw data or the displayed graphic of the grant analysis.
Steps	1. Perform grant analysis on a selected search result 2. Click "Save" to download the generated figures or the raw data.
Related UI	Figure 6

Number	10
Title	Saving Latent Knowledge Analysis Results
Description	The user wishes to save either the raw data or the displayed graphic of the latent knowledge analysis.
Steps	1. Perform latent knowledge analysis on a selected search result. 2. Click "Save" to download the generated figures or the raw data.
Related UI	Figure 12

Number	11
Title	Advance Search
Description	A user can add more specific filters for their search (like publish date, author, etc.)
Steps	1. From any page, click the "Advanced Search" button. 2. Fill in any desired fields and click "Search."
Related UI	Figure 13

Developer Use Cases:

Number	1
Title	Search Results API
Description	We provide APIs to get search results as JSON rather than HTML, as well as plaintext of specific articles.
Steps	1. Send a GET request to the API using the correct parameters
Related UI	Example API Call: http://uresearcher.com/api/search/your-search-query

Number	2
Title	Configuration file
Description	Configuration file (in .yaml format) 1. Add key for other publishers 2. Add preset for advance search
Steps	1. Download the template configuration file. 2. Make any necessary changes. 3. Upload the file to apply your changes.
Related UI	ELS_API_KEY: some_keys_here PRESET_SEARCH: some_keyword_here

Figures
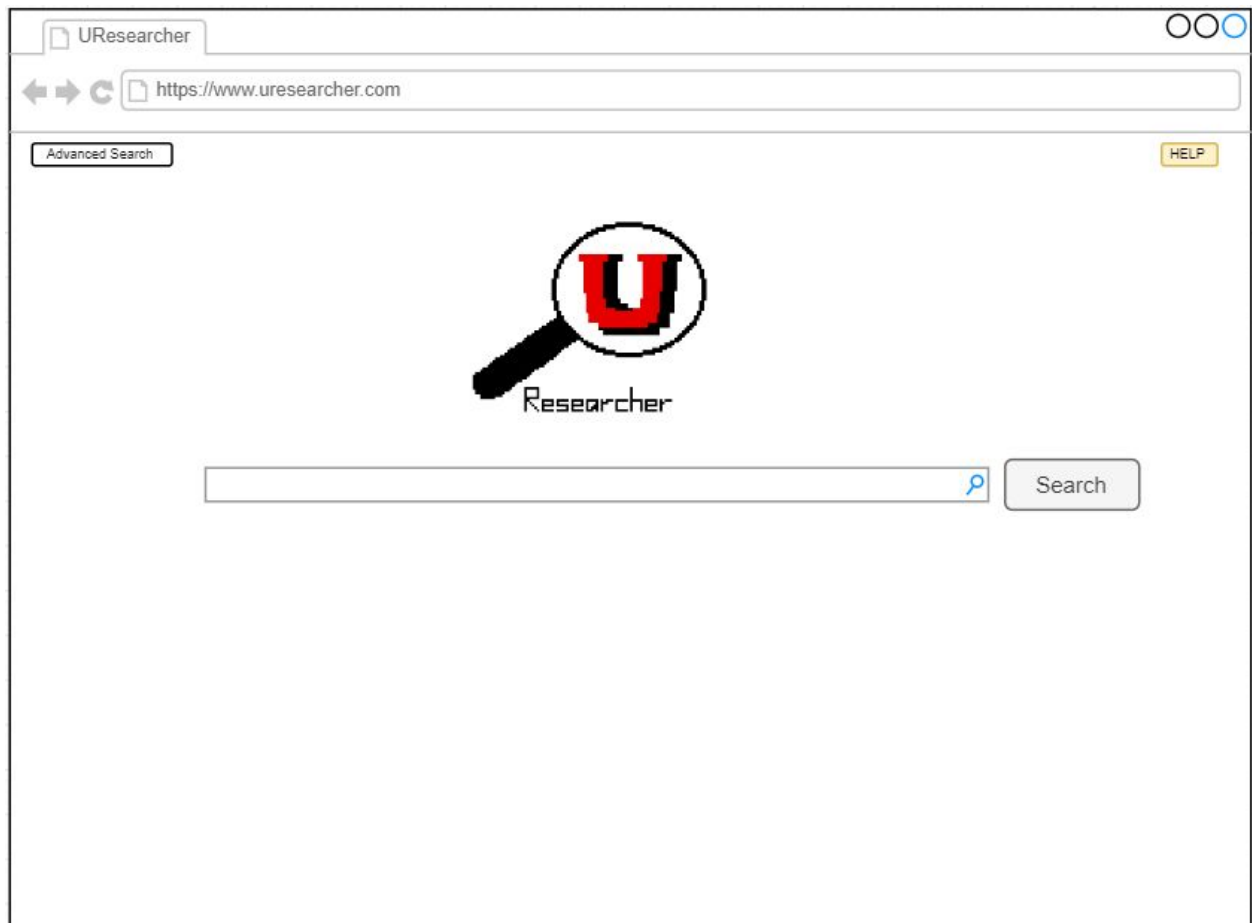
Figure 1 - Home Page
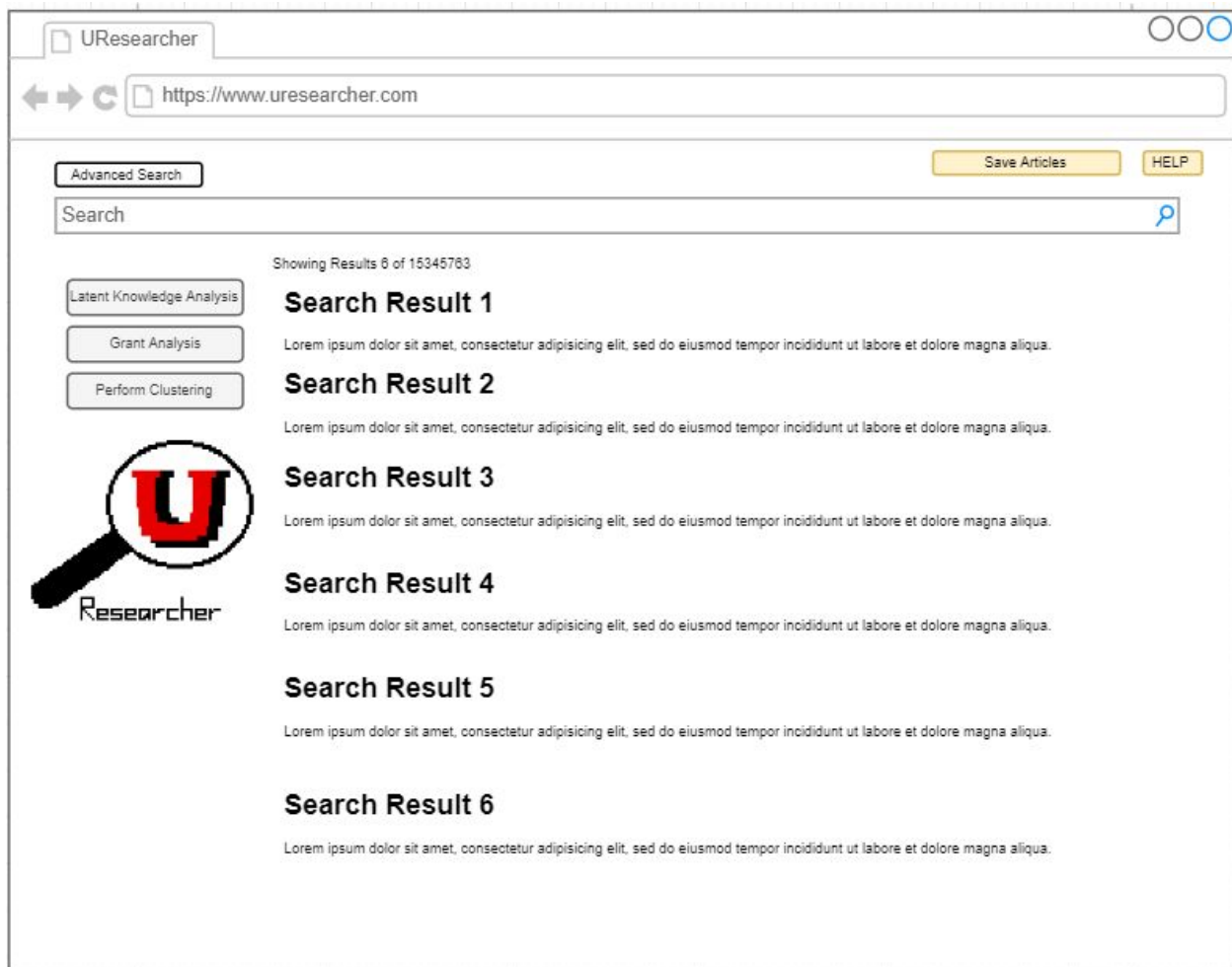
Figure 2 - Results Page
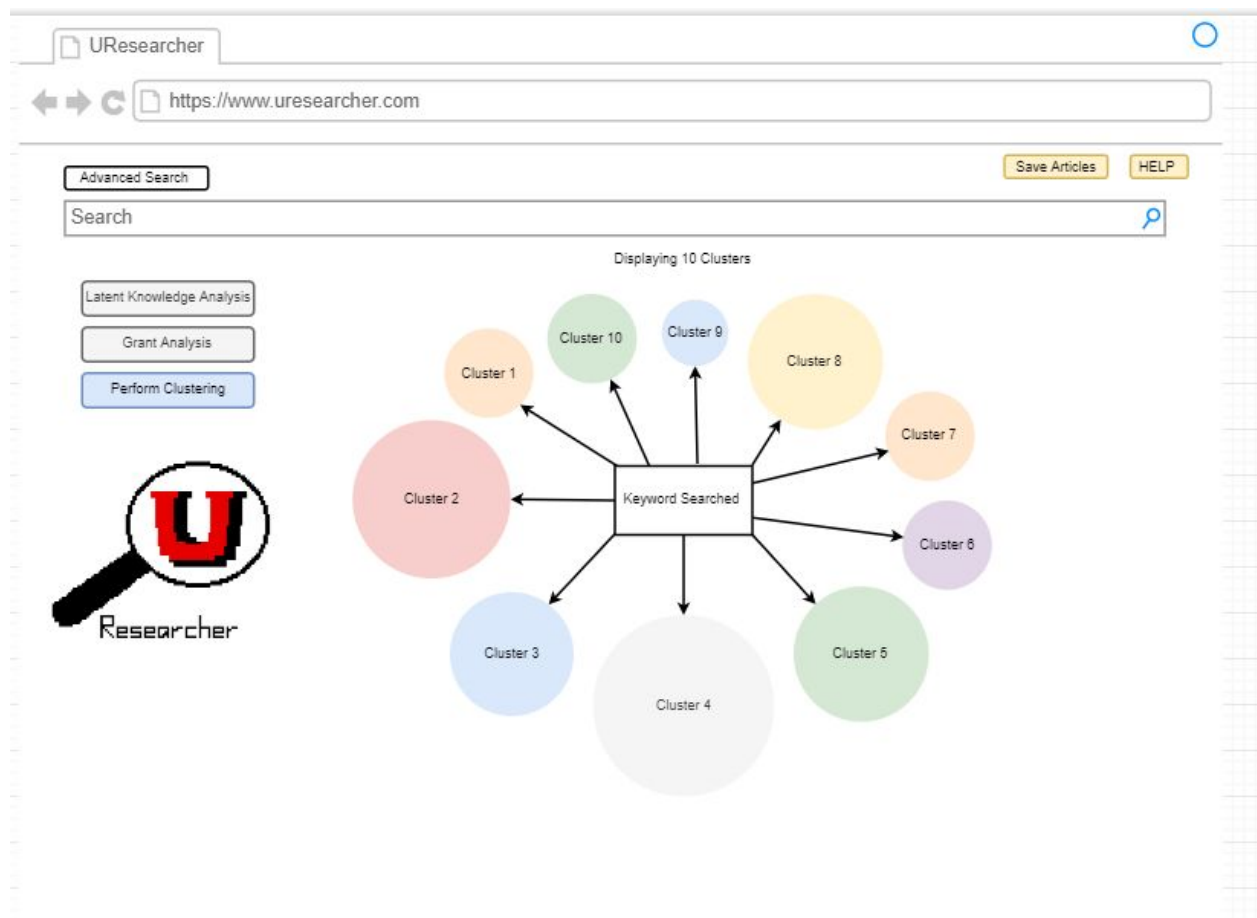
Figure 3 - Clustering Page
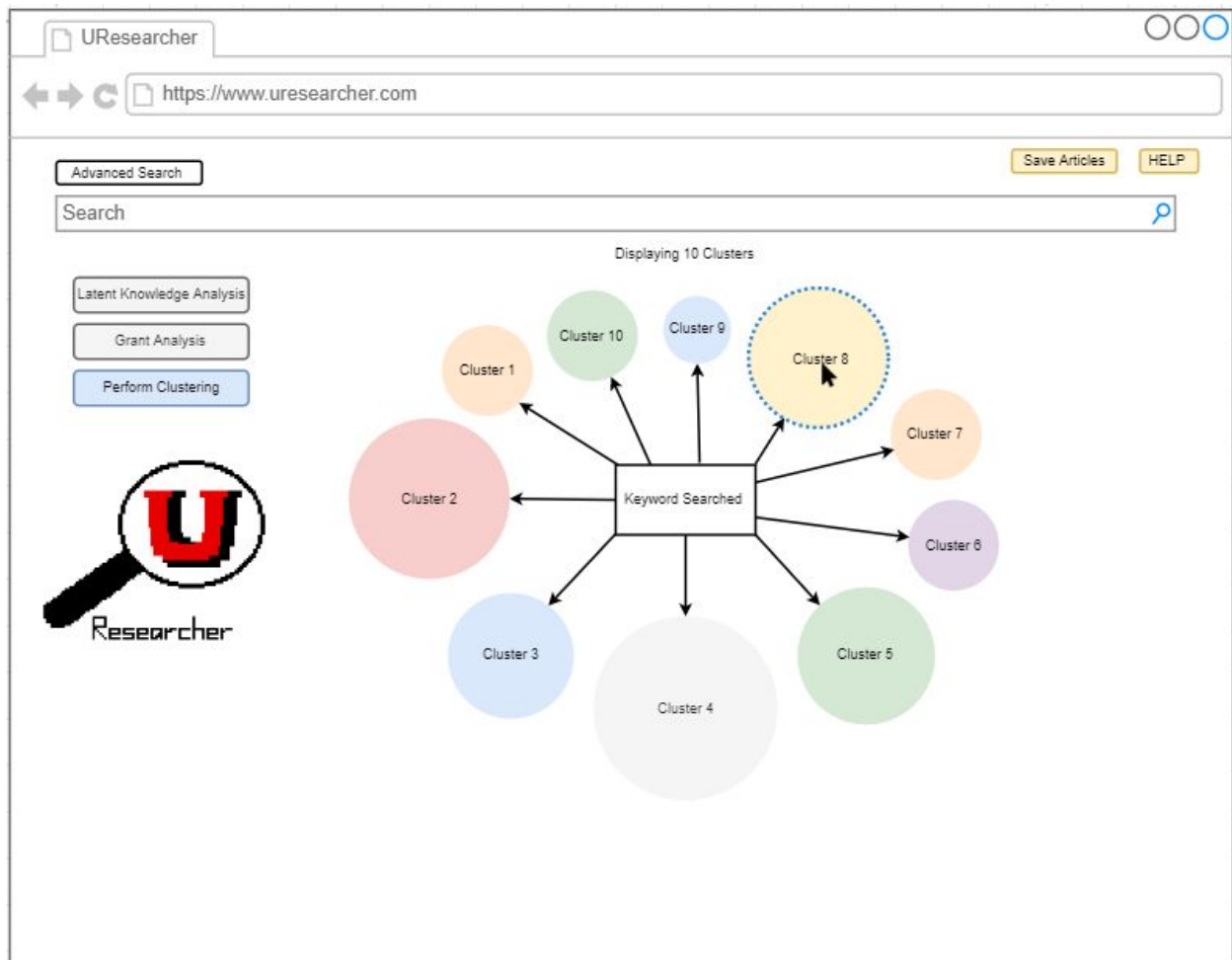
Figure 4 - Select Cluster Page
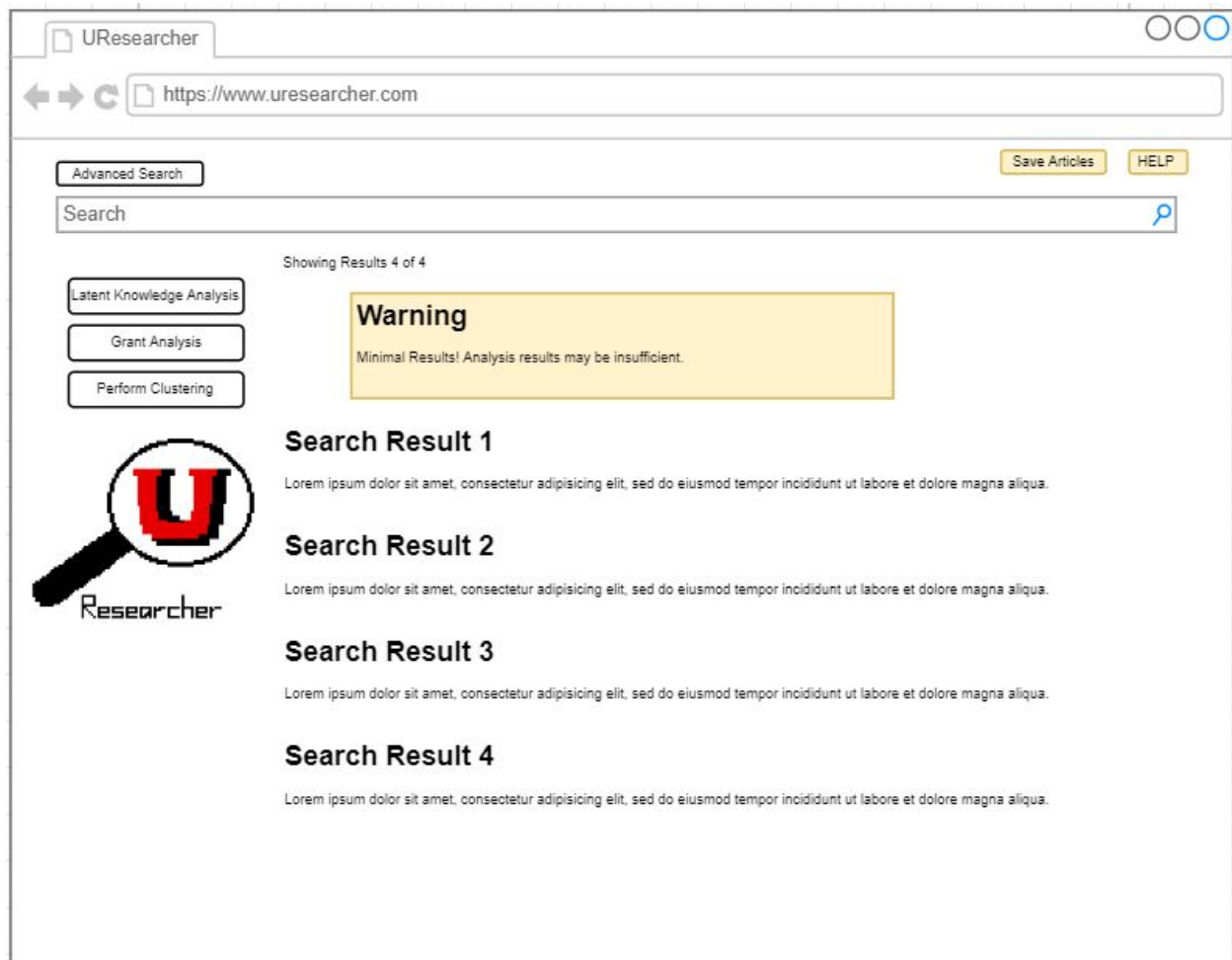
Figure 5 - End of Search Page
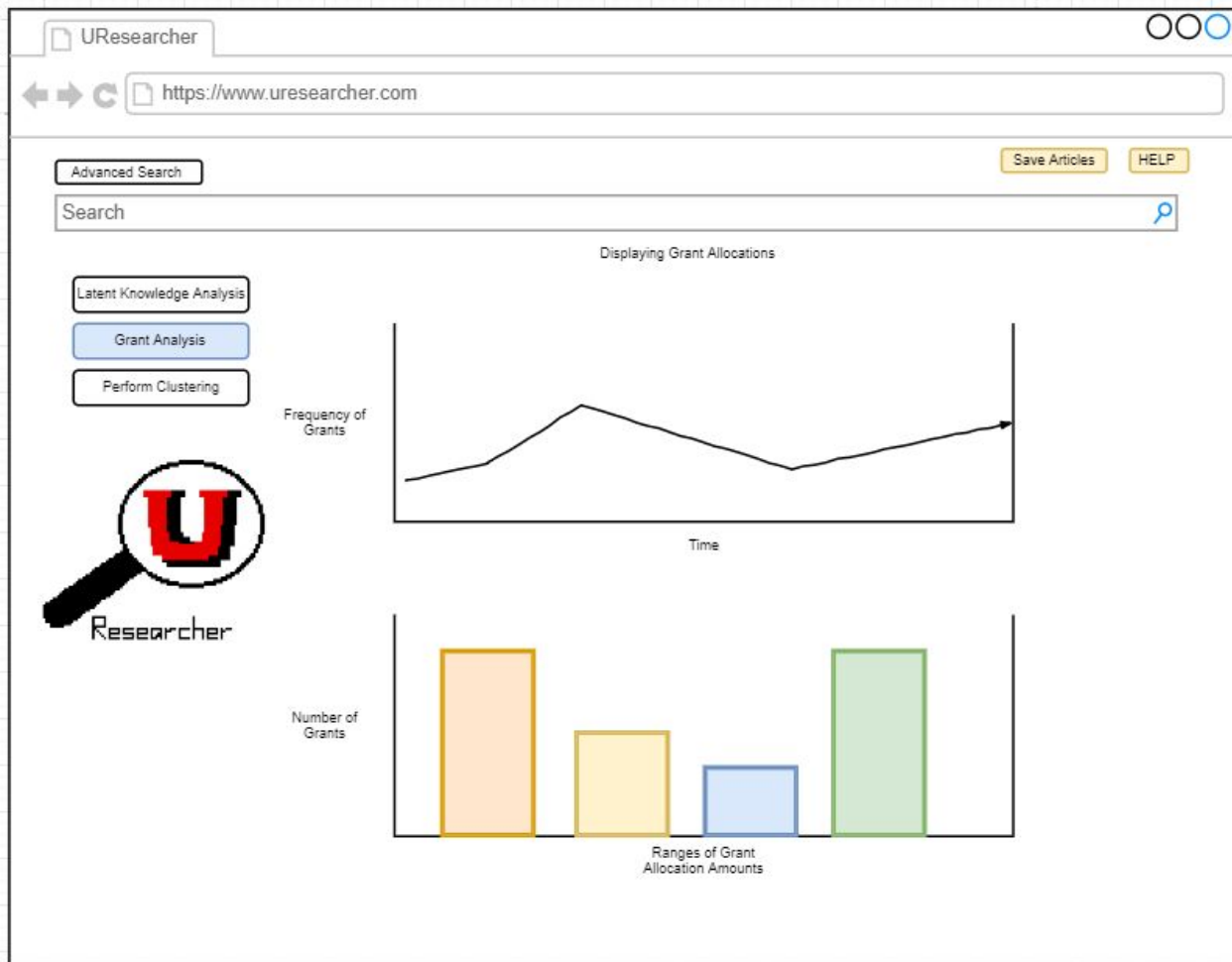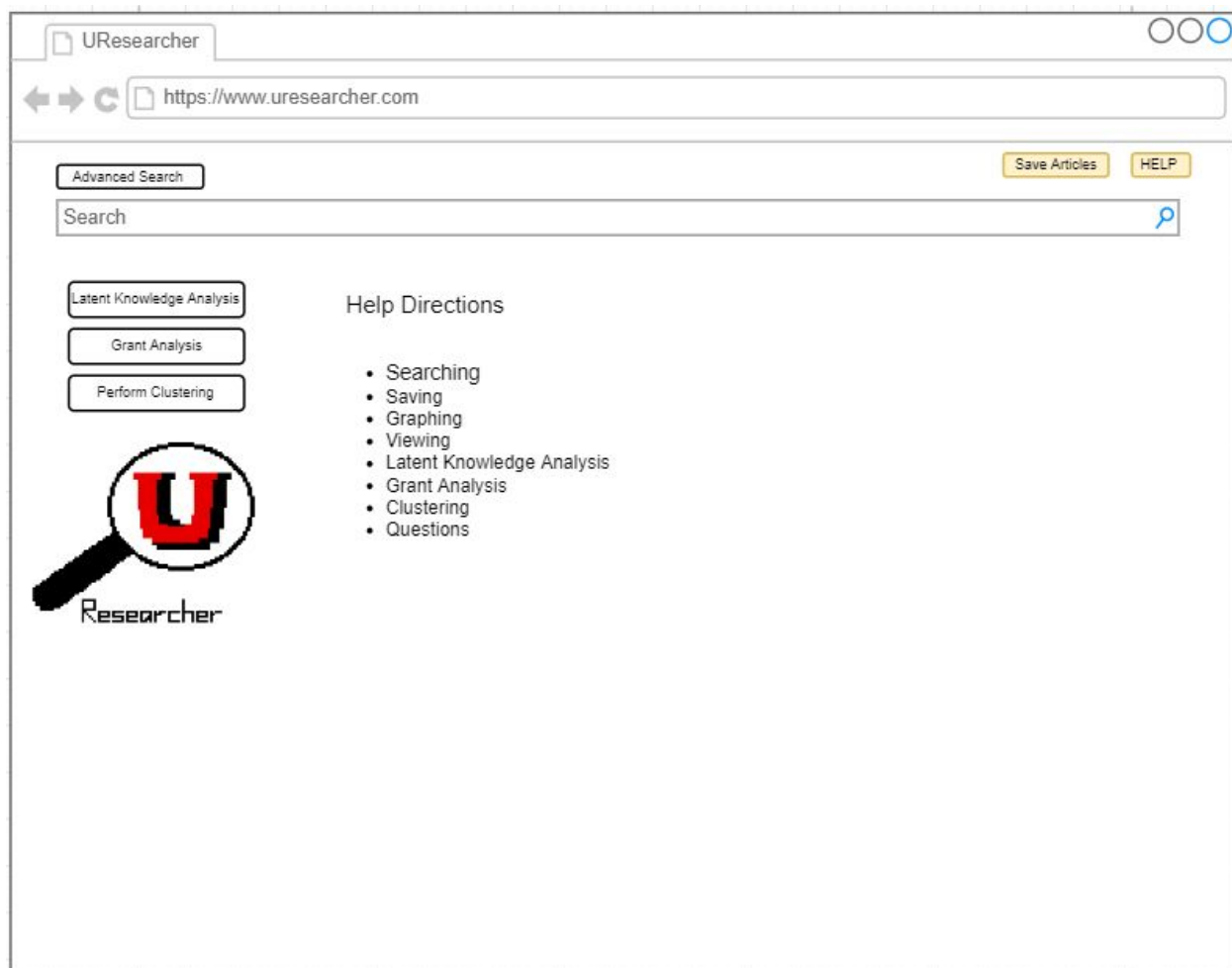
Figure 6 - Grant Analysis Page

Figure 7 - Feedback Page

Figure 8 - Help Page
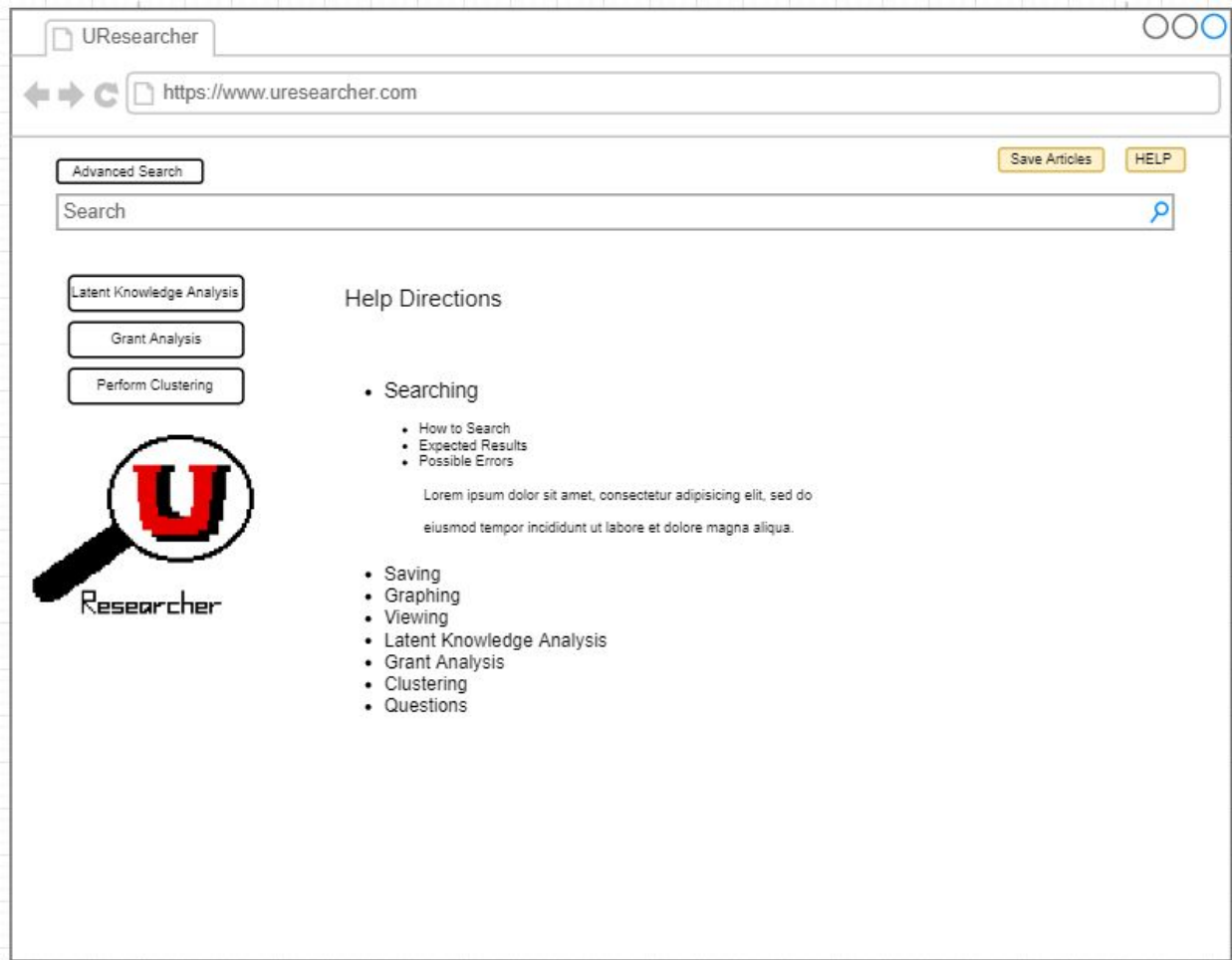
Figure 9 - Help Page (Extended)
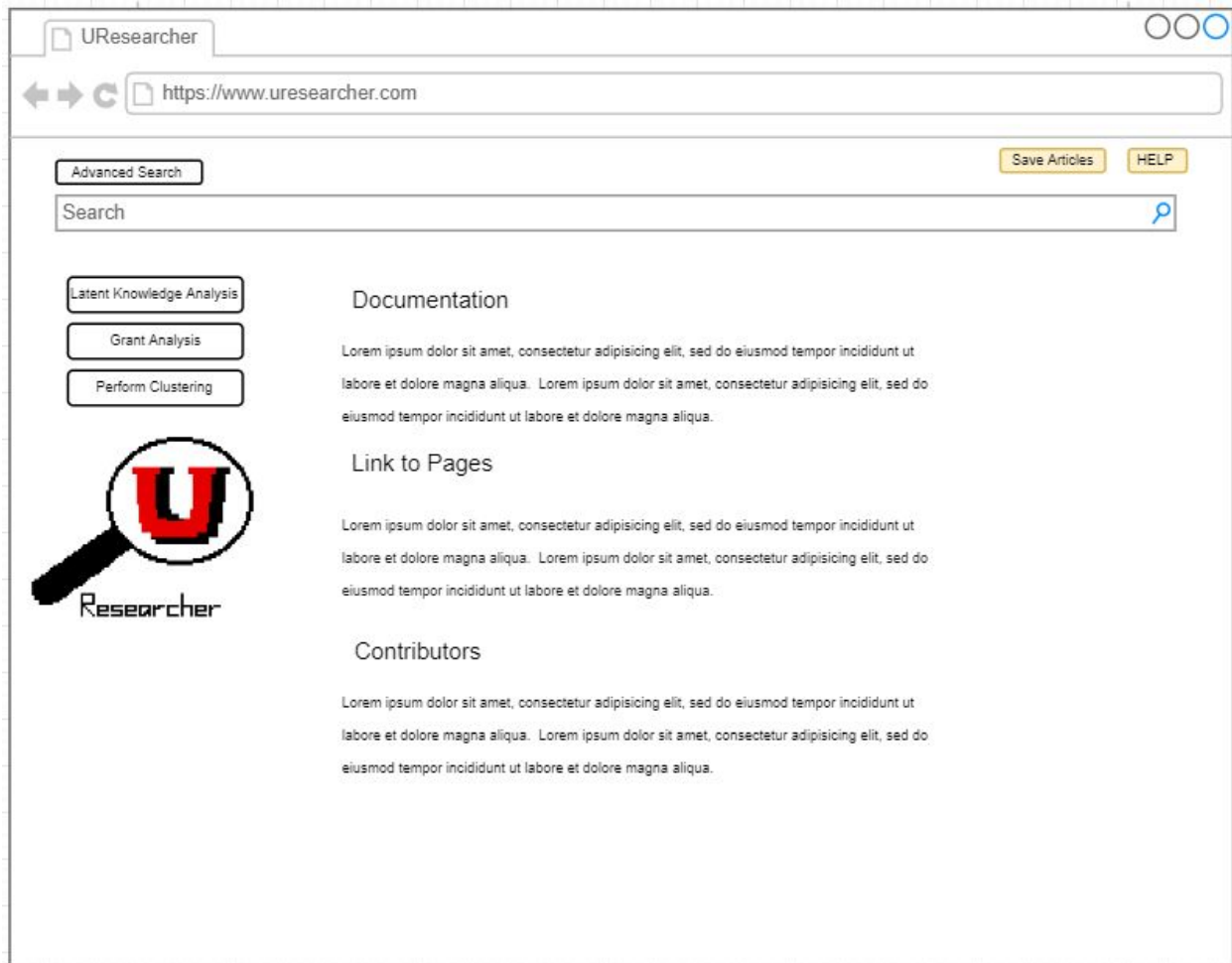
Figure 10 - Documentation Page

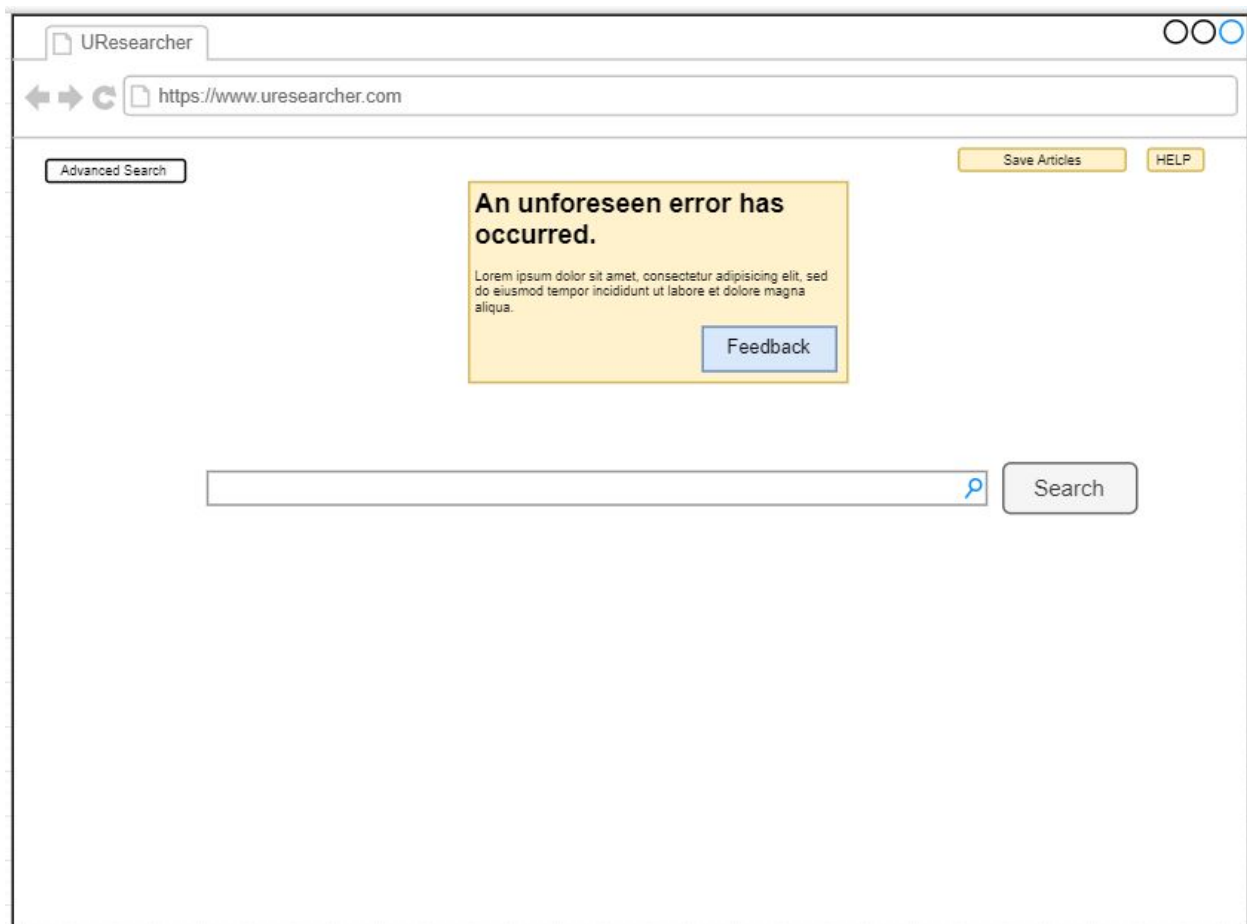Figure 11 - Error Page

Figure 12 - Knowledge Analysis Page

Figure 13 - Advanced Search