Visual Analytics for Sensemaking in Attribute Ranking Context

Submitted April 2023, in partial fulfilment of
the conditions for the award of the degree **BSc Hons Computer Science.**

20214735
School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text

Signature: /

Date: 04/24/2023

Acknowledgement

I would like to express my gratitude to my supervisor, Prof Kai Xu, for his guidance and advice, encouragement, and patience in guiding me to complete my development and achievements of this project.

I would also wish to thank all my friends for always saving me a seat in the lectures and labs and leading an ear whenever I was stuck. Finally, I would like to thank my parents for their unwavering belief in my ability to succeed.

Abstract

A novel application is presented in this project. CloudChoice, a Chrome extension designed to aid users in sensemaking tasks through effective data visualization tools. CloudChoice features three panels: Word Cloud, Edit, and Compare. The Word Cloud panel employs NLP techniques to generate a dynamic word cloud of topic and attribute names while browsing web pages. The Edit Panel enables users to select and update keywords from the Word Cloud Panel for attribute ranking. The Compare Panel displays item names and user-selected attributes ranked according to values that can be customized by adjusting and inverting attribute weights. Techniques such as web-scraping, NLP models, and d3 libraries were utilized in the extension's development. User evaluation feedback reported positive responses, although technical difficulties arose, leading to identification of areas for improvement. This study's findings contribute to the field of information visualization and provide direction for future research to expand CloudChoice's potential. CloudChoice offers a unique solution to sensemaking tasks by integrating NLP techniques and data visualization tools.

Table of Contents

1. Introduction ***5***

1.1 Sensemaking **5**

1.2 Aims and Objectives **6**

2. Motivation ***6***

2.1 Overview **6**

2.2 Prior Software **6**

3. Related Works ***7***

3.1 Attribute Ranking Interfaces **8**

3.2 Keywords Extraction **8**

4. Description of the Work ***9***

4.1 Functionality Specification **10**

5. Methodology ***10***

5.1 Overview **10**

5.2 Data Management **11**

5.2.1 Data Collection 11

5.2.2 Data storage 12

5.3 Understanding the Data **12**

5.4 Attribute Ranking Interface **13**

6. Design ***13***

6.1 Data Handling **15**

6.1.1 Data Collection 15

6.1.2 Data Analysis 16

6.2 Word Cloud **17**

7. Implementation ***18***

7.1 Scope Narrowing **18**

7.2 Web-scraping **18**

7.3 Data Storage and Interactive Table **20**

7.4 Keywords Extraction **21**

7.5 Attribute Ranking **22**

8. Evaluation ***23***

8.1 Keyword Extraction Method Evaluation **23**

8.2 User Experience **24**

9. Summary and Reflections ***25***

9.1 Project Management **25**

9.1.1 Motivation and Drive 25

9.1.2 Time Management 25

9.1.3 Gantt Chart 27
9.1.4 User Data Management 28

9.2 Contributions and reflections.. 28
9.2.1 Innovation..28
9.2.2 Computer Laws, Social, Ethical and Professional issues 29
9.2.3 Reflection and Future Considerations..30

Bibliography...31

1. Introduction

The objective of this project is to build *CloudChoice*, a set of effective data visualisation tools that facilitate users in making decisions based on the result of searching vast datasets online. Currently, these tasks are often performed manually, and difficulties arise because of the limitations of human cognition. Visual analytics tools, working as a mindmap could provide a feasible solution to aiding people in organizing thoughts and shaping problems more systematically. A serviceable visual analytical toolkit can effectively increase the efficiency and support the decision making process.

1.1 Sensemaking

The process of making decisions based on large dataset of information involves making sense of the information available and is called sensemaking. The stages of sensemaking often consist of understanding the data, generating hypotheses, selecting analysis methods, creating novel solutions, and critical thinking and learning where necessary. [1] Multi-tier characterization is established to analyse user activities, recording the provenance of *events*, *actions*, *sub-tasks* and *tasks*, from low-level to high-level. [2] While *events* and *actions* are more semantic, *sub-tasks* and tasks contains more human comprehensible information.

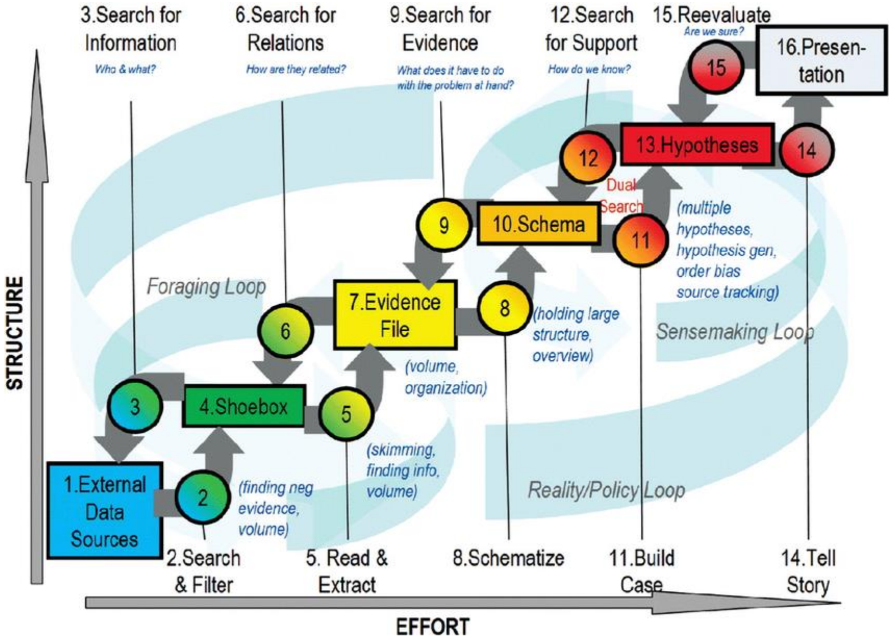


Figure 1- Stages of Sensemaking

Sensemaking tasks go through a number of stages, usually commencing with foraging and understanding the provenance of data. On a browser, the user's click-stream usually underlies the intention, namely, the sub-task and task the user is performing. At this stage, data is collected and organized using SensePath, a web-based visual analytical Chrome extension that enables scholars to retrieve user click-stream data in the form of user logs.[3]

Making sense and understanding data comes next and is the higher stage of recording *events* – capturing sub-tasks and tasks. Capturing these high-level descriptions is more challenging, especially without direct user input. [3] I propose a web-crawler methodology, starting with collecting texts from browsed webpages, then extract key words from input texts using Natural Language Processing (NLP) techniques. Python libraries like spaCy, YAKE, rake-nltk, PKE and KeyBERT are selectively applied to check for better performance. More detailed methods will be introduced in Section 5 - methodologies. Having extracted the related keywords, the visualisation interface is then designed to fit them into automatic feature ranking systems that facilitate comparison tasks by users . The interface would require users to choose from a number of keywords to form an attribute ranking system. Should auto-detection fail to detect the correct number, users can modify the attributes and their corresponding figures. Considering the difficulties in various context and use cases, this project will initially focus on the use case of searching for an ideal world's-top graduate schools.

1.2 Aims and Objectives

The **aim** of this project is to design and develop a visual analytics tool to facilitate all stages of sensemaking tasks that are applicable to real world user cases. An example of a tool designed for one stage is SenseMap, which is intended for the 'Information foraging and triage' stage.

The key **objectives** of the research are:

- Visualise the actions in the information foraging and triage stage, including search and filtering the information into a shoebox.
- Schemitize all the evidence found into a clear integrated structure, and aid the discrimination stage where information are to be examined for close relation with the use case.
- NLP models of keywords extraction tasks are applied in this stage to support visualisation more effectively.
- Visualise information filtered by user and keywords extraction models. The visualisation interface enables interactive attribute ranking functionalities.

2. Motivation

2.1 Overview

In seeking to promote the efficiency and manipuility of sensemaking tasks under the context of complex comparison jobs, an interactive and automatic visualisation tool is proposed. This will free users from the need to switch between browser tabs to compare the location, price, or cleanness of different accommodations, allowing the decision to be made in a more rational way with reasons more trackable and attributes more clearly presented.

2.2 Prior Software

Users carrying out decision-making tasks by visualising the search evidence obtained from a number of information websites in the form of a map or a path [3]. Marian Dörk *et al.* argue in *PivotPaths: Strolling through Faceted Information Spaces* that insufficient use is made of data seen from structural and semantic perspectives while presenting and exploring information spaces. When navigating from one item to another, or comparing several items, it is essential to know their data attributes and relations in order to search for information resources and to make final decisions. Emphasis is therefore placed on the related attributes in a sensemaking task. For example, users designing a travel itinerary cannot avoid deciding on accommodation. Among the features of a hotel or an Airbnb homestay, location, price, wifi and cleanliness will influence the final choice. Being able to visualize attributes of items in a search could prevent users from overloading human cognition capacities in the comparison tasks.

Visualising search results in the form of maps or paths[3] has become increasingly popular for information websites aiming to facilitate decision-making tasks. Presenting information in an easy-to-discern form helps users quickly identify and evaluate relevant options. However, research by Marian Dörk et al. [4] suggests that current approaches to presenting and exploring information spaces may be overlooking important structural and semantic perspectives of the data being presented.

In their study of PivotPaths, Dörk et al.[4] argue that effective decision-making is impossible without an understanding of the data attributes different items and relationships between them . When navigating from one item to another, users wishing to make informed choices must find it easy to compare and contrast the relevant attributes. This is particularly relevant in tasks such as planning a travel itinerary, where decisions about accommodation can have a significant impact on the overall experience. In such cases, attributes such as location, price, WiFi, and cleanliness may be key factors in determining the user's final choice.

3. Related Works

A considerable body of literature related to the visualization of sensemaking already exists and draws upon concepts from the fields of human-computer interaction and machine learning. Notable examples include SenseMap[5], mentioned above, and CoVA, a visual analytics system designed to address conflicts that may arise during asynchronous collaborative data analysis [6]. In 2021, a novel visualization approach was proposed to facilitate provenance data visualization to evaluate applicability of algorithms.[7] While SenseMap(2022) and Ilkay Melek Yazici (2021) have focused more on tracing and tracking provenance data, SensePath places greater emphasis on providing users with a timeline interface that displays the various browser windows opened in the course of a sensemaking task.

Academics have proposed projects focused on specific use case or condition. Examples include Visual Bibliographies (VisualBib), a real-time tool that assists researches in bibliographic tasks[8]. Similarly, PivotPaths is a system that navigates through various interlinked academic resources. It enabled faceted relations and filterings to better facilitate academic sensemaking tasks. [4]

The works discussed above represent a growing trend in the field of sensemaking visualization. Researchers are keen to exploit advances in natural language processing and human-computer interaction to give users fresh tools and interfaces to effectively comprehend and interpret complicated data sets. With the ability to track data sources, these visualization tools can ensure data analysis processes' transparency and accountability while identifying areas that require optimization for researchers. Ultimately, creating new and creative visualization approaches for sensemaking has the potential to revolutionize how we interact with and understand intricate data sets across various domains, from scientific research to business analytics.

3.1 Attribute Ranking Interfaces

This project will be built on the above related works from stages of foraging the data and analysing the user intentions, but the core part is the generated interface. Scholars have developed multiple heterogeneous attributes ranking systems.

The study has been developing since no later than 2013 when LineUp was implemented. [9] LineUp is successful not only in the aspect of human-computer interaction, with well-designed learnability and guessability, but is also powerful in realising complex camparison requirements. It allows users to flexibly combine, refine and scale attributes to visually observe the effect of changes visually.

While LineUp is a scalable visualisation tool for bar charts, VisualBib provides an interface of spider charts. In fact, my work is largely inspired by VisualBib by Dattolo and Corbatto[8]. This very recent visualisation design has disassembled features of a paper and designed a core set of bibliographic tasks, incorporating panels presenting metadata, timeline, command, exploration and analysis. The analysis panal is designed in the form of a spider charts, which is ideal for items to be compared with a certain number of essential attributes.

3.2 Keywords Extraction

Natural Language Processing (NLP), as a nonnegligible branch of machine learning and artificial intelligence, has been systematically established for some time. Studies of various problem-solving techniques concerning keywords extraction have appeared at least since 2009. [10]

More recently, scholars have developed python toolkits and libraries that are suitable for the use case scenarios in this research. One essential component of natural language processing (NLP) systems is the identification of Named Entities (NER), which enables the identification and categorization of different types of entities such as individuals and places, among others, within a given text.. Scholars compared studies on NER softwares in 2019, [11] and found that the correct literature identified is approximately 50%. StanfordNLP is usually the winner with the highest correctness, other players being StanfordNLP, NLTK, OpenNLP, SpaCy and Gate.

Explosion launched spaCy in 2016. SpaCy is an open-source NLP toolkit that can perform multiple functions such as recognizing parts of speech, analyzing dependencies between words, and identifying named entities, among others. Its performance in these tasks is exceptional.[10] It has been practically applied more recently in named entity annotation tasks more recently. [12] SpaCy has automatically labeled hundreds of scientific and technological example sentences, and successfully produced word clouds.

Other than spaCy, Muhammad K. et al. made a more recent breakthrough . KeyBERT was published to better facilitate processing and extracting knowledge from unstructured textual data.[13] It uses contextual word embedding and, focusing solely on abstract data, outperformed conventional techniques like Rake, Yake, Text Rank, Gensim, and TF-IDF in generating keywords similar to those provided by the authors.

CloudChoice integrates the aforementioned keyword extraction models to facilitate users with attribute selections in sensemaking and decision-making tasks.

4. Description of the Work

In the current era, it is common for individuals to engage in sensemaking tasks involving comparing multiple items based on various attributes. For instance, on Black Friday, consumers may spend time comparing discounts for similar products on different websites. Similarly, individuals planning a vacation may need to switch between multiple tabs in order to compare accommodations. Undergraduate students, meanwhile, may conduct research on world-renowned graduate schools and manually record information in an Excel spreadsheet. In response to this need, the proposed project aims to develop a web-based tool that can facilitate sensemaking tasks within a multi-attribute ranking context.

Initially, the tool will be designed to cater to the needs of students searching for graduate schools around the world. Specifically, the tool will provide a comprehensive platform for organizing and comparing relevant information on different schools, including details such as programme offerings, faculty expertise, funding opportunities, and other relevant attributes.

The interface is composed of three panels: **Word Cloud Panel**, **Edit Panel** and **Compare Panel**.

The **Word Cloud Panel** displays the output of the “Keywords Extraction” process, which uses NLP techniques to generate a word cloud of topic and attribute names of which users can select and update keywords from. These words can be updated in real-time while the user is browsing webpages while browssing webpages. Each word is an interactive object that can be clicked and chosen to be either an item name or an attribute name.

Words selected from the Word Cloud Panel become part of the **Edit Panel** and are used to construct axes for the attribute ranking interface. For instance, "Tuition" may be an important attribute for an undergraduate with a limited budget who is browsing webpages containing information about tuition. If this word appears in the word cloud, the user can select it as **an attribute name**. Similarly, if the user is browsing webpages related to Carnegie Mellon University, the name of the university will appear in the word cloud and can be set as **an item name**. The user can then fill in corresponding numbers to allow for further comparison.

Once the form is partially completed, the user can jump to the **Compare Panel**, which is the core design of the tool. The Compare Panel is composed of item names and various attribute names chosen by users in previous steps. Each attribute is presented in a unique colour, and each item (in this case, the university) is ranked according to the values of the chosen attributes. The weights of the attributes can be adjusted and inverted, allowing for flexible customization. For example, , students with a limited budget may prioritize a lower tuition cost and invert the weight of this attribute, while those with a larger budget may scale down the weight of tuition.

A list of functional specifications is provided.

4.1 Functionality Specification

To better manage the working progress, functionality specification is listed in Kanban Board. Tasks are listed as below.

1	**Users want to be able to switch between three panels**
1.1	*The system needs to provide a user interface.*
1.2	*The system needs to have three panels independently.*
1.2	*The system needs to store data that is consistent in the three pages.*
2	**Users want to be able to edit attributes interactively: from word cloud, or directly by typing**
2.1	*The system needs to have interactive word cloud section.*
2.2	*The system needs to be able to transfer information from word cloud to edit panel.*
3	**Users want to be able to edit values in edit panel**
3.1	*The system needs to support data overwriting in front-end interface.*
4	**Users want the system to automatically generate keywords that can be considered as attributes.**
4.1	*The system needs to implement NLP models based on the userlog.*
4.2	*The system needs to integrate output of keywords with word cloud visualisation.*
5	**Users want to assign and edit weights of attributes**
5.1	*The system needs to have an attribute ranking interface in Compare panel should be able to adjust attribute values according to different weightings*
5.1	*The system needs to enable attributes in Compare panel to be inversely rated.*
6	**Users want the comparisons between items list to be clearly visualised**
6.1	*The system needs to have a Compare panel where all attributes of items are presented in bar chats, annotated with different colours.*
7	**Users want to able to edit the attributes and items intuitively.**
7.1	*The system needs to have an Edit Pane where attributes can be dragged to form different orders.*

Requirements
Specifications

5. Methodology

5.1 Overview

The aim of the project is to design and develop a visual analytics tool to facilitate decision-making in sensemaking tasks that are applicable to real world user cases, and is especially targeted towards users researching graduate schools around world. To realise the purpose, the essential steps are as follows.

1. Effective **Data Management** for Enhanced Utilisation and Manipulation.

In order to leverage user behaviour and preferences for improved decision making, it is necessary to collect and manage user clickstream data along with their userlog information. The process of collecting data is followed by retrieving webpage content that users have browsed. This raw data is then stored and organized by *CloudChoice* for further utilisation and manipulation. Such data management practices enable organizations to gain valuable insights into user preferences and behaviour, which can help inform decision-making processes.

2. Comprehensive **Data Understanding** through Schematization and NLP

Data understanding is a critical data analysis stage. Schematization into a clear integrated structure of all evidence found with an overview provides a foundation for further analysis. This step aids in the keyword-searching stage where information is examined for close relevance to the use case. NLP techniques can be applied to support visualisation more effectively. These techniques enable data analysts to identify patterns and relationships within the data efficiently, facilitating the development of data-driven solutions.

3. Innovative **Attribute Ranking Interface** for Enhanced User Experience

Based on the evidence collected and understood, multiple hypotheses of topics and attributes can be formed and presented to users in a visually intuitive interface. This approach uses word clouds to present topics and attributes in an engaging manner, making exploring the identified topics and attributes both immersive and user-friendly. This attribute ranking interface allows users to select the keywords most relevant to their needs so that they can access the information they require quickly and easily. Such an approach enables organizations to enhance the user experience and increase user engagement by providing targeted information and insights.

5.2 Data Management

5.2.1 Data Collection

In this research endeavour, the data collection process involves acquiring userlog information, which stems directly from users. The userlog data comprises the Uniform Resource Locator (URL) addresses, as well as the corresponding time duration used in browsing the pages. SensePath has already implemented this data collection technique on the Chrome browser.

Following the acquisition of the userlog data, web-scraping techniques are employed to extract further insights from the visited websites. Four distinct methodologies are employed, namely:

1. puppeteer,
2. puppeteer-web,
3. browserify,
4. Python libraries.

Puppeteer is a server-side node.js application that enables automation and control of Google Chrome, while puppeteer-web was created solely for client-side web development. [14]Unfortunately, puppeteer-web is now deprecated, but the raw code remains accessible. Browserify, on the other hand, bundles the puppeteer files.[15] The Python libraries used in this research include the HTML parser and urllib, an extensible tool for opening URLs.

After careful consideration of the various methodologies, the python library urllib was deemed the most appropriate for extracting all textual information from designated web pages. The extracted textual data is recorded in a text-based format and subsequently used as input for the subsequent NLP stage.

5.2.2 Data storage

Modern web applications frequently need to be able to store data on the client-side, either to enhance user experience or to facilitate offline functionality. Three of the most commonly used client-side storage options in modern web development are the Chrome storage API, IndexDB, and localStorage. These solutions are examined for practicality in this project in Section 7.3.

The Chrome storage API allows web developers to store and retrieve data in a structured manner. It is available for use on Chromium-based browsers and provides three different types of storage areas: local storage, sync storage, and managed storage, each of which has its specific use case.

IndexDB is another client-side storage system and provides a more advanced storage solution than the traditional key-value storage model used in localStorage. IndexDB enables developers to store and retrieve data in an object store, providing enhanced querying capabilities and the ability to store complex objects.

localStorage is a simple key-value pair storage system available in all modern web browsers. It provides developers with an easy-to-use storage solution for small amounts of data that need to persist between sessions or page refreshes but is simple in nature and lacks the more advanced features provided by the other two storage options.

5.3 Understanding the Data

Keyword extraction is an important aspect of NLP that involves identifying and extracting relevant keywords or phrases from a large corpus of text. This process is particularly useful for analysing user-generated data such as social media posts, customer reviews, and feedback forms. To perform this task efficiently, developers often rely on machine learning and NLP libraries such as SpaCy and KeyBERT, which are implemented in Python.

Python’s simplicity, flexibility and ease of use have made it a popular programming language in the field of machine learning and NLP. Most state-of-the-art libraries and toolkits for NLP and machine learning are written in Python. It is therefore no surprise that Python has become the preferred language for NLP applications, particularly in the context of keyword extraction.

A number of studies have evaluated the performance of various NLP libraries and toolkits, including a review of StanfordNLP, NLTK, OpenNLP, SpaCy, and Gate by Schmitt *et al*. in 2019. Gensim, SpaCy, and Keras are applied and have been written into a practical guidance of text analysis. [16] SpaCy has been identified as one of the best-performing NLP libraries for data-driven labelling tasks, particularly in identifying cutting-edge terminologies. [12] Additionally, KeyBERT, a more recent keyword extraction model, has been found to outperform older models in this domain. [13]

In light of these findings, SpaCy and KeyBERT have become increasingly popular for keyword extraction in recent years. In this research, these libraries will be used to generate a Word Cloud from the input text collected in the Data Collection stage. This process involves leveraging the power of machine learning algorithms to extract relevant keywords and phrases from large volumes of text data.

5.4 Attribute Ranking Interface

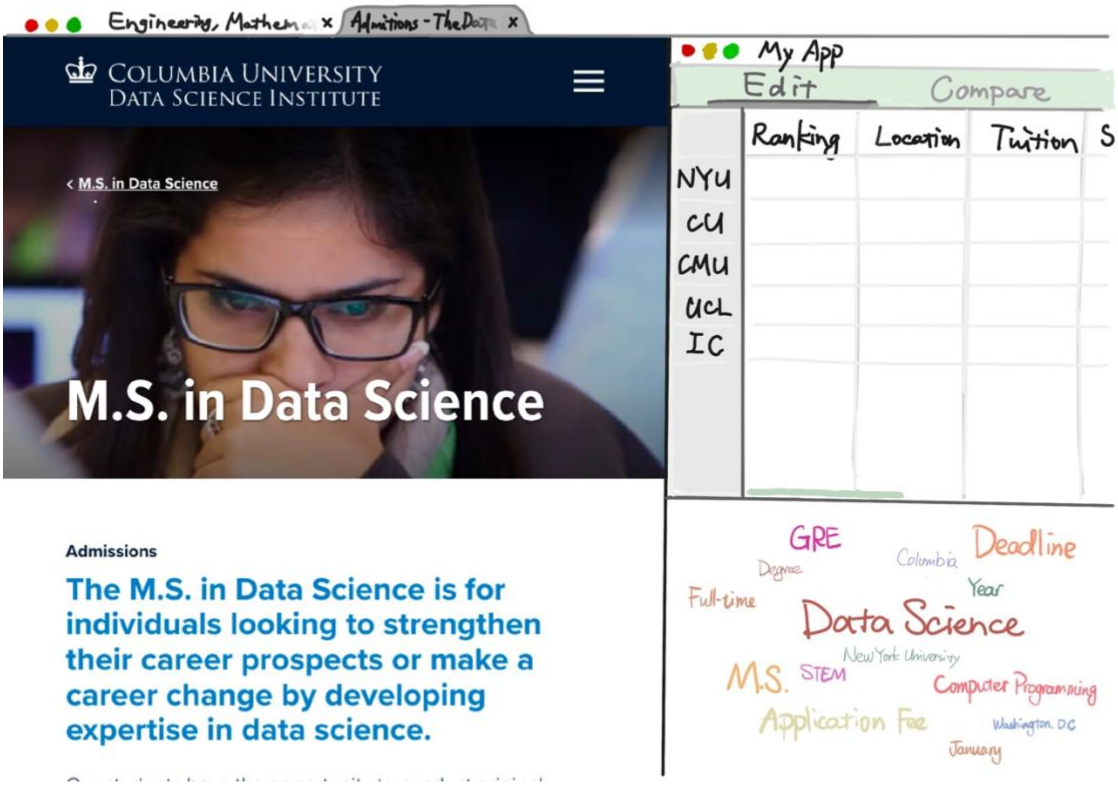
The project's front-end interface is constructed using pure JavaScript and CSS augmented with the component library Chakra to access additional building blocks. The interface allows users to perform interactive actions such as dragging keywords to designated positions. The project's data visualization component is implemented using libraries such as d3.js by means of which a wide range of bar chart models can be deployed.

In instances where the collected data of chosen attributes may be incorrect, a panel is provided to temporarily store all figures in an organized form. Corrections can be made in this panel, after which the resulting data will be displayed in the ranking panel in the form of bar charts.

The project is web-based and the comparison process is conducted online without a backend database to store relevant data. As a result, only the outcome of the completed comparison can be downloaded. The results are presented on the "compare" page, and various visualization methods can be selected depending on the use case. For instance, to compare several items with equal importance and bias the generated graph would display a ranking based on the sum of attribute values. On the other hand, if the user assigns different emphasis to different attributes, such as prioritizing "tuition" or "scholarship" for undergraduate students with financial difficulties, the ranking would be adjusted accordingly to reflect the user's priorities.

6. Design

In accordance with the specifications of the functional requirements and prototype design, development of the CloudChoice application has been structured into three main phases: data handling, word cloud visualization, and attribute ranking interface construction. A low-fidelity prototype is shown in figure 2 and 3. A comprehensive elaboration of each stage given in the discussion that follows.



COLUMBIA UNIVERSITY
DATA SCIENCE INSTITUTE

‹ M.S. in Data Science

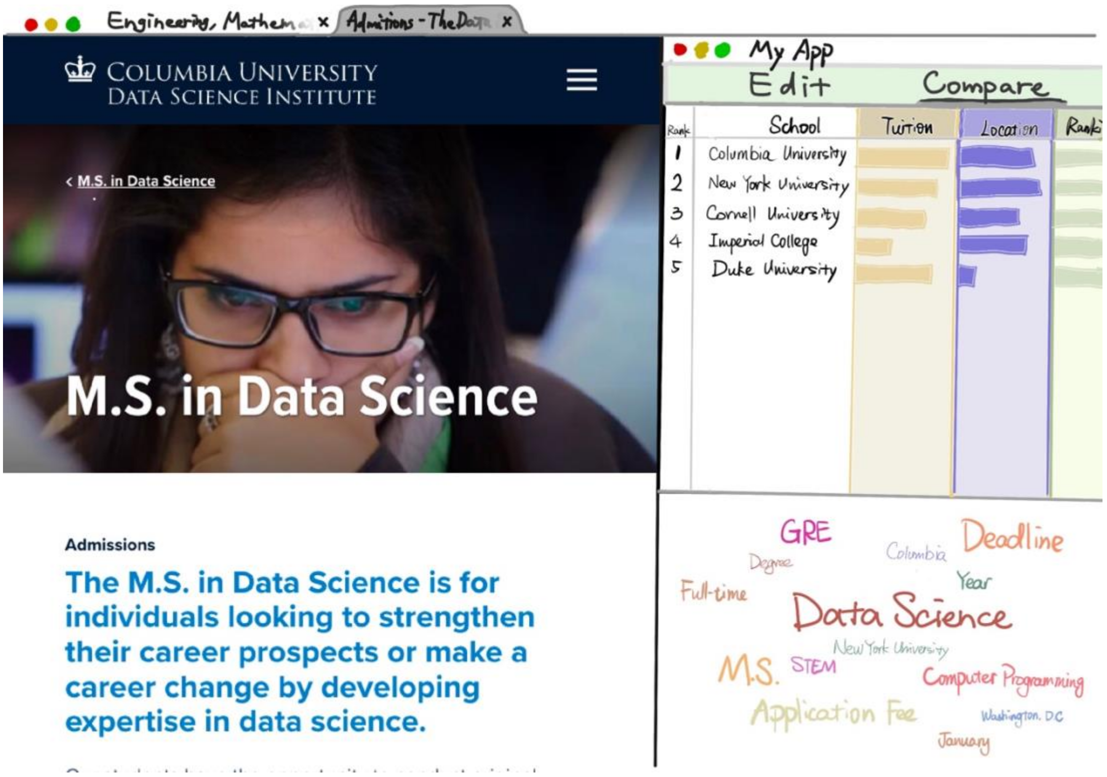
M.S. in Data Science

Admissions

The M.S. in Data Science is for individuals looking to strengthen their career prospects or make a career change by developing expertise in data science.

	Ranking	Location	Tuition	S
NYU				
CU				
CMU				
UCL				
IC				

Figure 2 – Low Fidelity Prototype (Edit Panel)



COLUMBIA UNIVERSITY
DATA SCIENCE INSTITUTE

‹ M.S. in Data Science

M.S. in Data Science

Admissions

The M.S. in Data Science is for individuals looking to strengthen their career prospects or make a career change by developing expertise in data science.

Rank	School	Tuition	Location	Ranki
1	Columbia University			
2	New York University			
3	Cornell University			
4	Imperial College			
5	Duke University			

Figure 3 – Low Fidelity Prototype (Compare Panel)

6.1 Data Handling

6.1.1 Data Collection

Before performing data analysis, the userlog data must be pre-processed to refine it and make it amenable for analysis. The proposed process of data management is illustrated in figure 4.

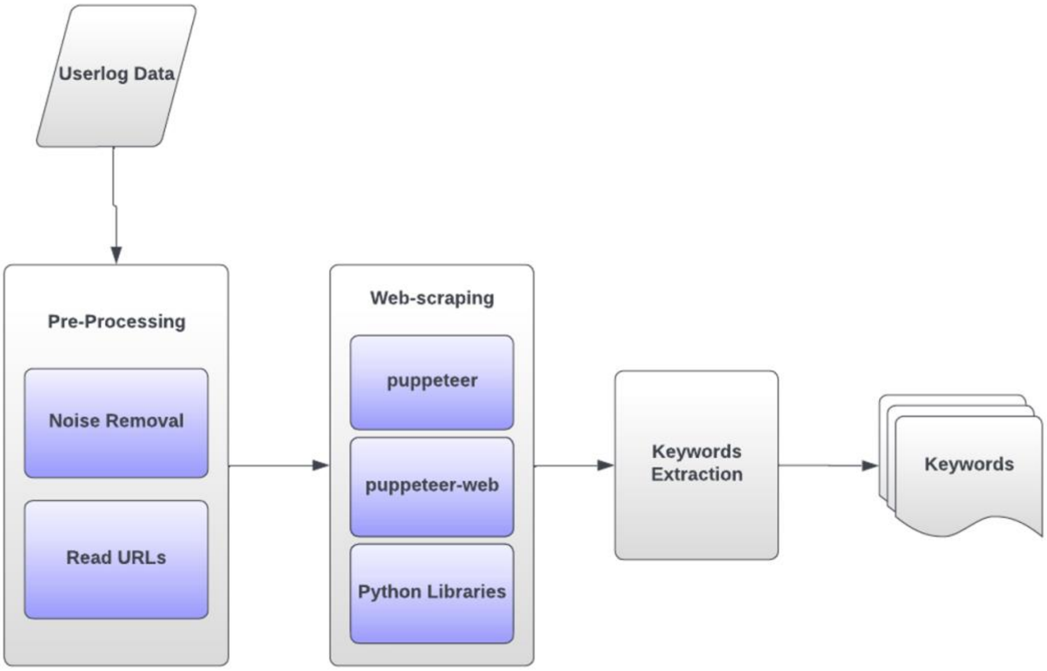
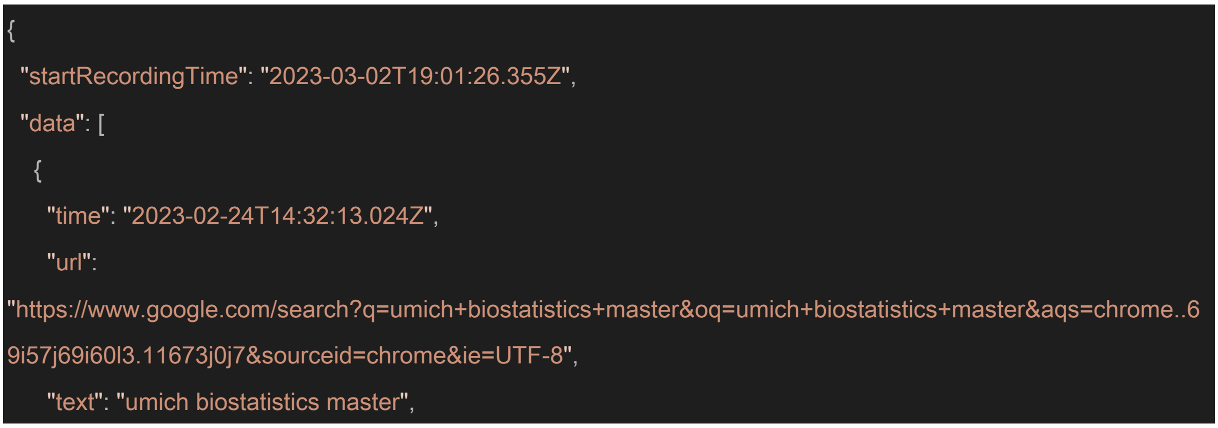


Figure 4 – Flowchart of Proposed Data Management Approach

The raw data is collected from a variety of sources. While the specific details are beyond the scope of this project, it is worth providing an overview of the data pipelines. The main source of userlog is people with fluency in English reading abilities. Section 4.3 contains a more in-depth description of the data collected . The data collection system is fairly simple. The chrome extension ‘SensePath’ is installed in the users’ Chrome browser to collect the userlog information including the current time, webpage title, browsing duration and, most importantly, the URL itself. After a 15 to 20 minute session of researching on a certain task, the userlog is stored and saved in a json file. Here is a code snippet of user researching for ‘umich Biostatistics master’:



```
{
  "startRecordingTime": "2023-03-02T19:01:26.355Z",
  "data": [
    {
      "time": "2023-02-24T14:32:13.024Z",
      "url":
"https://www.google.com/search?q=umich+biostatistics+master&oq=umich+biostatistics+master&aqs=chrome..6
9i57j69i60l3.11673j0j7&sourceid=chrome&ie=UTF-8",
      "text": "umich biostatistics master",
      "type": "search",
      "id": 1677249133024,
      "endTime": "2023-02-24T14:32:14.024Z",
      "zoomLevel": 2,
      "customTranscript": "1 seconds spent in searching 'umich biostatistics master'",
      "transcript": "[01:00:17 - 01:00:18] (web search) \n1 seconds spent in searching 'umich biostatistics master'"
    },
...
```
Code snippet – userlog raw data

After the raw data has been collected it is processed as local files. This second process runs as a python file to extract and dig for more information in the userlog, including calculation of browsing time and scraping texts in the webpage of each URL. As mentioned in Section 5.2.1, several methods are applied to perform web-scraping. Puppeteer is a powerful web scraping tool developed by Google which allows developers to automate web page interactions and data extraction using a headless Chrome browser. Puppeteer provides a JavaScript API that can be used to control the browser, simulate user actions, and extract data from websites.

Puppeteer offers numerous benefits, particularly when dealing with intricate web pages and dynamic content. Its capabilities extend to AJAX requests, client-side rendering, and single-page applications, making it ideal for scraping modern websites. Additionally, Puppeteer can interact with various elements on a page, enabling users to automate complex workflows easily. This feature includes filling out forms, clicking buttons, and scrolling. The second approach is Puppeteer-Web, an open-source web scraping tool built on top of Puppeteer. Puppeteer-Web is a Node.js library that allows developers to control and automate Chrome or Chromium browser tasks. It extends Puppeteer by providing a simplified and easy-to-use API for web scraping, providing a simple and powerful way to extract data from websites using JavaScript and Chrome DevTools Protocol. The Python libraries HTMLParser and urllib are essential tools for web scraping, data mining, and data analysis tasks.

On careful consideration, the optimal solutions for collecting textual information are the HTMLParser and urllib libraries. More detailed implementation and comparisons between solutions are in Section 7.2. Once the data has been effectively gathered, the next step entails rigorous analysis of the acquired information.

6.1.2 Data Analysis

This section describes the use of NLP models for keyword extraction tasks. Specifically, six distinct models for keyword extraction are employed in the analysis of user data, with the aim of assessing and improving performance.

RAKE is an unsupervised keyword extraction model that identifies relevant keywords or key phrases from a given text using statistical measures.[17] It extracts candidate keywords by identifying significant co-occurring terms and scores them based on frequency, degree of word co-occurrence, and phrase length. RAKE is domain-independent and has been shown to perform well in various applications, including text classification, sentiment analysis, and document clustering. YAKE is another unsupervised model that identifies key phrases from a given text corpus using statistical features such as frequency, position, and context of words.[18] Its domain-independent approach (i.e., it does not rely on dictionaries or thesauri) makes it usable across different languages, and it has demonstrated competitive performance in benchmark studies.

In contrast to the two previous models, TopicRank, PositionRank, and MultipartiteRank are graph-based. TopicRank leverages a complete graph, where candidate key phrases are clustered into topics that serve as vertices.[19] A graph-based ranking model is then applied to assign a significance score to each topic, and key phrases are extracted by selecting candidates from the highest-ranked topics. In contrast, PositionRank uses words' position and context to identify salient keywords[20], while MultipartiteRank constructs a graph-based representation of the text using multiple types of information such as syntactic and semantic features. Both models have demonstrated strong performance in text classification, summarization, and information retrieval applications.[21]

KeyBERT, proposed by Van de Cruys et al. (2021), is an unsupervised method for keyword extraction and document classification that uses BERT embeddings.[13] It outperforms existing methods, including graph-based and statistical methods, and is domain-independent. KeyBERT first encodes the input text using BERT, selects the top-n most important sentences based on their cosine similarity with the document's BERT embedding, and then extracts keywords or keyphrases using a fine-tuned BERT model on a large keyword extraction dataset.

KeyBERT's interpretability is a notable advantage. Its extracted keywords and key phrases are derived from the most important sentences, resulting in more interpretable keywords than other unsupervised methods. As a result, KeyBERT is useful for information retrieval, text classification, and document summarization applications. More detailed analysis is in Sections 7.4 and 8.

6.2 Word Cloud

A sensemaking assistant visualisation tool must give users the ability to identify and select attributes that are significant in their ultimate decision-making process. In this regard, word clouds play a crucial role in enabling users to comprehend the essential keywords that have been browsed online during the time the task was performed. In the context of CloudChoice, word clouds serve as an intermediary component, and various JavaScript tools available online are used to integrate the requisite libraries into the application.

The Word Cloud is designed to classify keywords based on their output rank and score generated during the data analysis phase by NLP models. The significance of a particular keyword is directly proportional to its score, with a higher score implying a greater likelihood that the keyword represents an attribute or an item name. Thus, the Word Cloud serves as an effective tool for facilitating efficient attribute ranking.

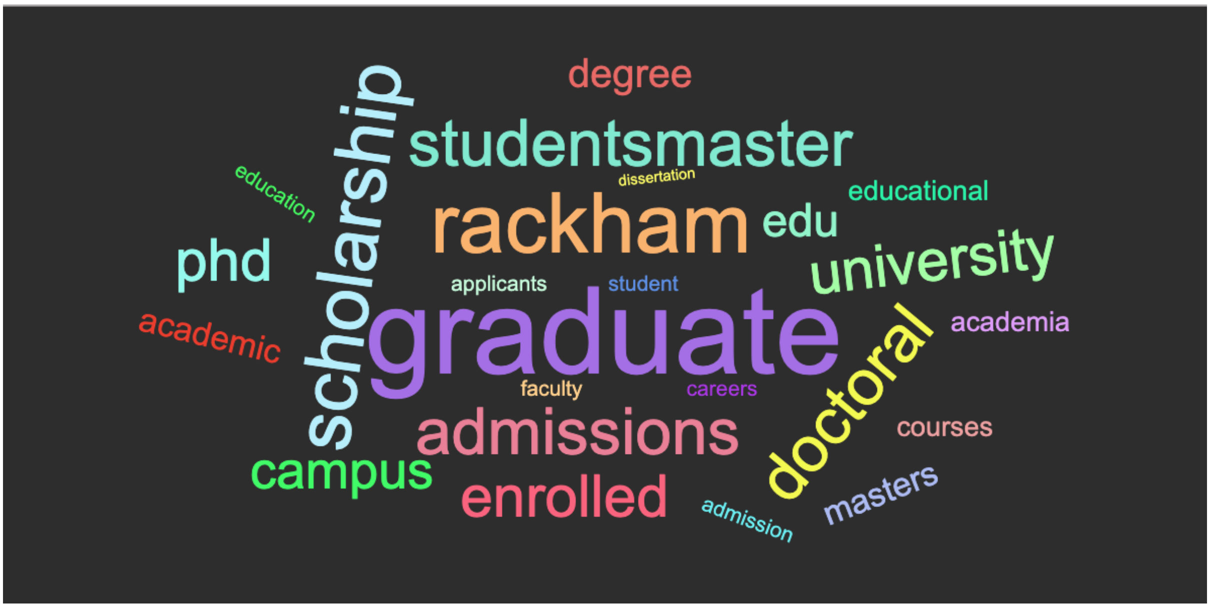


Figure 5 – sample output of Word Cloud

Figure 5 is a sample output of word cloud with a small input dataset. The top 5 generated keywords with rank and scores are: ('graduate', 48), ('graduates', 44), ('rackham', 42), ('scholarship', 41), ('doctoral', 39). This provides empirical evidence to suggest that the proposed solution is viable and practicable in real-world applications.

7. Implementation

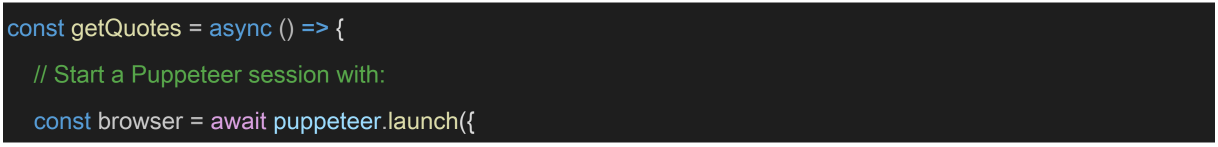
7.1 Scope Narrowing

The primary sources of data emanate from peer students affiliated with the University of Nottingham and possessing a distinguished educational background and ability to comprehend literary works. Data was acquired with limited financial support and within a stringent timeline and resulted in collection of information from 26 participants and a total of 1176 sample webpages. Due to the relatively small pool of interviewees and limited sample diversity, the utilization scope of the current CloudChoice application is primarily confined to users with advanced educational backgrounds and in a younger demographic. The potential expansion of the user base to encompass a more diverse demographic is contingent on the acquisition of additional funding and an extension of the project timeline, which would enable the collection of a more comprehensive dataset.

7.2 Web-scraping

In the forthcoming discussion, we shall delineate the specific advantages and disadvantages of the diverse methodologies described in section 6.1.1.

Primarily, the Puppeteer tool was implemented across multiple webpages within the dataset, as it demonstrated a remarkable ease of use and practicality when applied to real-world scenarios (as evidenced in the code snippet provided below).



```
const getQuotes = async () => {
  // Start a Puppeteer session with:
  const browser = await puppeteer.launch({
    headless: false,
    defaultViewport: null,
  });

  // Open a new page
  const page = await browser.newPage();
  await page.goto("http://example.com/", {
    waitUntil: "domcontentloaded",
  });
```

Code snippet - puppeteer

However, as CloudChoice is built purely on client-side, it is not realistic to integrate puppeteer with the front-end visualisation part and it is therefore not a feasible approach to build a real-time web-scraping chrome extension.

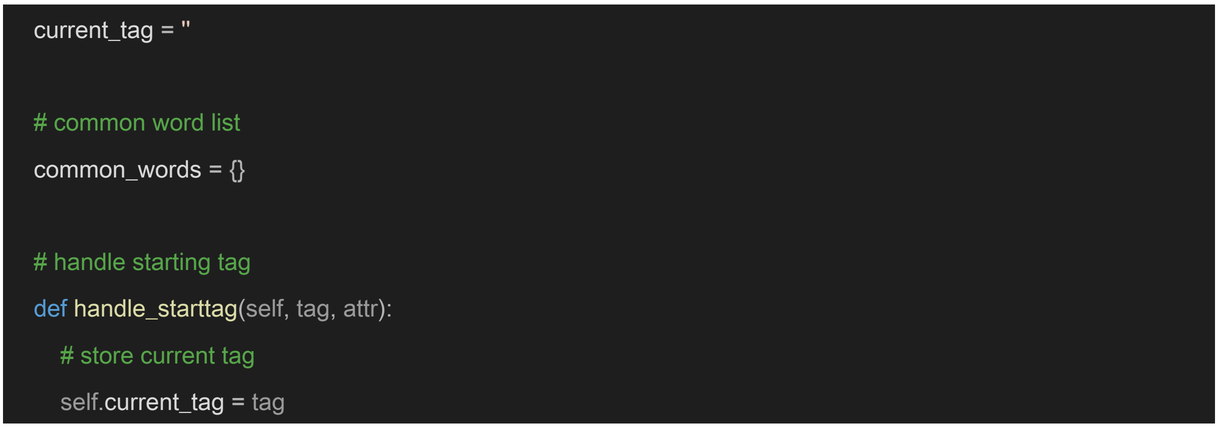
We can only deprecate the fact that Puppeteer-web is deprecated and only copies of it remains available.[22] It is no longer maintained for latest and secure usage.[23]

Browserify is another tool widely-used in the field of web development.[15] It enables programmers to construct Node.js-style modules that can be utilized in web browsers. Using the CommonJS module format, it is possible to compartmentalize code into smaller, reusable modules, mirroring the practice in a Node.js environment. This facilitates the management of complex JavaScript applications while reducing the potential for naming conflicts and other errors. Unhappily, Puppeteer-core is not specifically engineered for web browsers and so cannot be used in JavaScript tools constructed to function beyond the Node.js runtime environments.

Ultimately, it becomes necessary to use the Python libraries HTMLParser and urllib. The HTMLParser library, built into the standard Python library, allows efficient extraction of data from HTML files including text, tags, and attributes. HTMLParser works by defining a subclass of the HTMLParser class and overriding the appropriate methods to handle desired HTML data.

On the other hand, the urllib library, also built into the standard Python library, provides a powerful and convenient way to work with URLs. Its urllib.request module allows URLs to be opened and their contents read, while urllib.parse module provides functions to parse URLs into their components and to create URLs from their components. The urllib.error module handles errors that may occur when working with URLs, such as 404 errors.

```
class WordsParser(HTMLParser):
    # tags to search text within
    search_tags = ['p', 'div', 'span', 'a', 'h1', 'h2', 'h2', 'h3', 'h4']


    # current tag
```



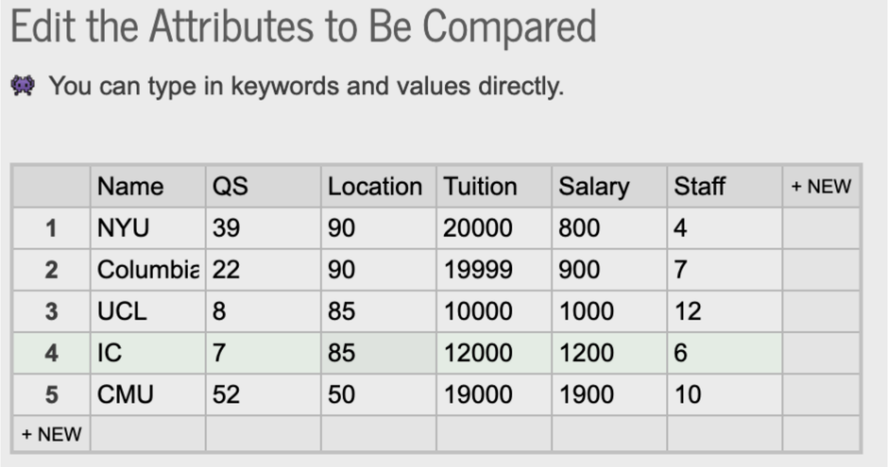
```
current_tag = ''


# common word list
common_words = {}


# handle starting tag
def handle_starttag(self, tag, attr):
    # store current tag
    self.current_tag = tag
```

code snippet - web-scraping process

Despite their usefulness, HTMLParser and urllib are limited in their integration into client-side and real-time systems and so do not provide a comprehensive solution. Nevertheless, this approach stands out as the most efficacious way to extract all words from a diverse range of HTML tags.

7.3 Data Storage and Interactive Table

Before implementing an attribute ranking interface, it is essential to establish the dataset that will be used for the ranking process. An interactive table interface has been devised for this purpose and permits users to input attribute names and values directly or to select them from a word-cloud. When a particular grid is selected, the interface will modify its styles, generating a highlighted effect and so enhancing the user's experience. Real-time updating of the dataset occurs as soon as an attribute name or value is entered or modified, a functionality that is thoroughly explained in the ensuing paragraphs detailing the storage method.



Edit the Attributes to Be Compared

You can type in keywords and values directly.

	Name	QS	Location	Tuition	Salary	Staff	+ NEW
1	NYU	39	90	20000	800	4	
2	Columbia	22	90	19999	900	7	
3	UCL	8	85	10000	1000	12	
4	IC	7	85	12000	1200	6	
5	CMU	52	50	19000	1900	10	
+ NEW							

Figure 6 – screenshot of interactive table (fabricated numerical data is used)

As a chrome extension, *CloudChoice* is developed on client-side scripting. *CloudChoice* currently employs localStorage rather than other options as its primary storage mechanism, a decision that may be based on several factors including simplicity, compatibility, data persistence and storage limitations.

localStorage presents a simple, easy-to-use key-value storage system that requires minimal setup, making it an ideal option for extensions that do not necessitate complex data management and retrieval.

localStorage data is persistent, meaning that it remains even after the browser is closed or the computer is shut down. This feature makes it suitable for storing user preferences or settings that must persist across sessions. In the 'graduate school choices' use cases, users might encounter circumstances where the work is half-done and data should be accessible and editable after navigating back from other webpages.

localStorage has a larger storage capacity than alternative options such as cookies, enabling greater data storage without any apprehension of reaching storage constraints. A user may not browse a very large number of webpages during a 15-minute-long task, and the storage API should be able to manipulate such an amount of text.

However, it is essential to note that localStorage has certain limitations, such as being unsuitable for storing significant amounts of data and storing all data in plain text format, which can pose a security risk if confidential information is stored in the extension. In light of potential future developments, if CloudChoice is to be upgraded or redesigned to accommodate more intricate tasks and data-intensive operations, a storage methodology should be selected that supports larger datasets and ensures heightened security measures.

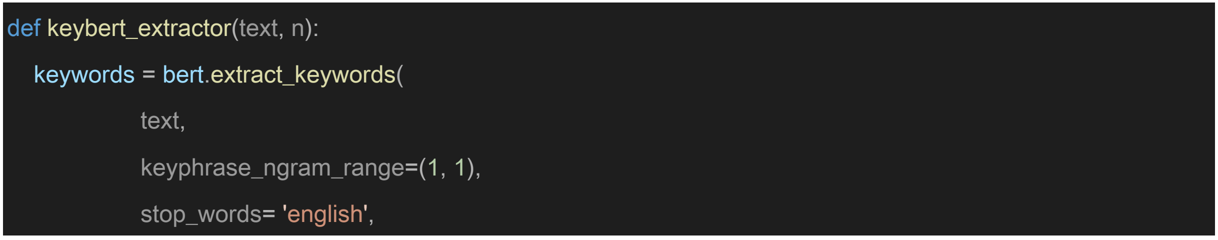
7.4 Keywords Extraction

Section 6.1.2 introduced six of the most popular keywords extraction techniques. The present study examines and compares the approaches, features, output, and performance of the six models applied to collected data. RAKE, YAKE, TopicRank, PositionRank, and MultipartiteRank are all unsupervised models that do not rely on pre-defined labelled data to extract keywords. In contrast, KeyBERT is a supervised model that uses pre-existing labelled data to extract keywords.

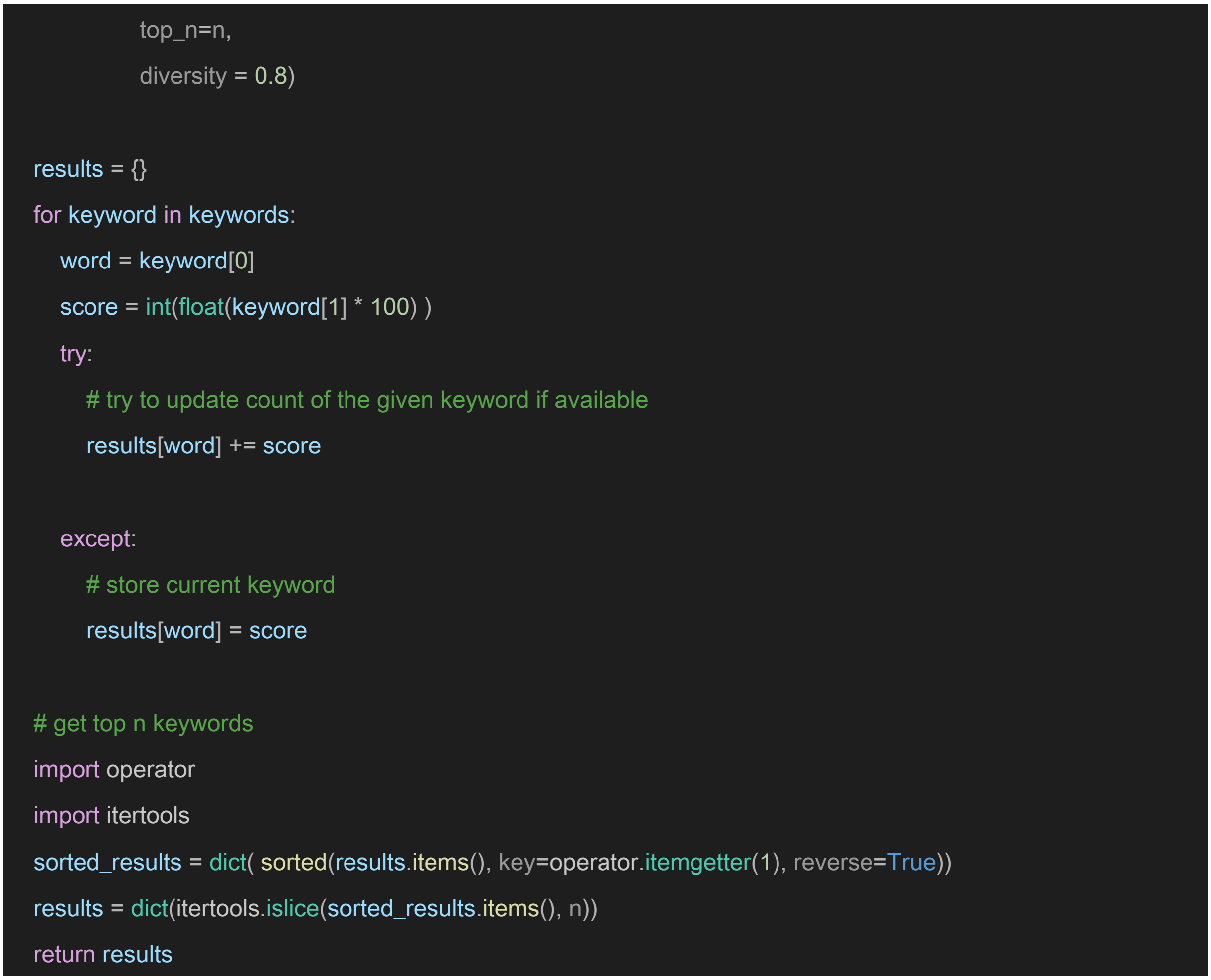
Of the features used in keyword extraction, the former five models consider frequency, co-occurrence, and position of words in a text. Conversely, KeyBERT employs deep learning-based models to extract keywords based on the semantic similarity between the text and existing labelled data.

In terms of output, RAKE, YAKE, TopicRank, PositionRank, and MultipartiteRank generate a list of keywords with scores or weights indicating their relative importance in the text. KeyBERT, on the other hand, produces a list of phrases from the text that are most similar to the labelled data.

Below is a set of implementation details of KeyBERT, which are then explained.



```
def keybert_extractor(text, n):
    keywords = bert.extract_keywords(
                text,
                keyphrase_ngram_range=(1, 1),
                stop_words= 'english',
```



```
            top_n=n,
            diversity = 0.8)

    results = {}
    for keyword in keywords:
        word = keyword[0]
        score = int(float(keyword[1] * 100) )
        try:
            # try to update count of the given keyword if available
            results[word] += score

        except:
            # store current keyword
            results[word] = score

    # get top n keywords
    import operator
    import itertools
    sorted_results = dict( sorted(results.items(), key=operator.itemgetter(1), reverse=True))
    results = dict(itertools.islice(sorted_results.items(), n))
    return results
```

Code snippet – KeyBERT Implementation

KeyBERT is a supervised model that combines deep learning-based models and pre-existing labelled data to extract keywords based on semantic similarity. The other five are unsupervised models that use statistical measures to extract keywords. KeyBERT's unique approach results in better performance. Section 8 contains an explanation of the availability of analytical performance evaluation, which serves as a critical aspect of the subject matter under discussion.

7.5 Attribute Ranking

In fulfillment of the aforementioned extension, I adhered to the initial plan and utilized d3.js, an open-source JavaScript library renowned for its ability to create interactive and customizable data visualizations. The latest version, i.e., version 4.7.4, was employed to develop a stacked bar chart that effectively represented the results. The data was normalised and scaled down , therefore the bar charts are organized in descending order based on the respective values.

The visual outcome pertaining to the selection of graduate schools, as previously exemplified, is depicted in Figure 7.

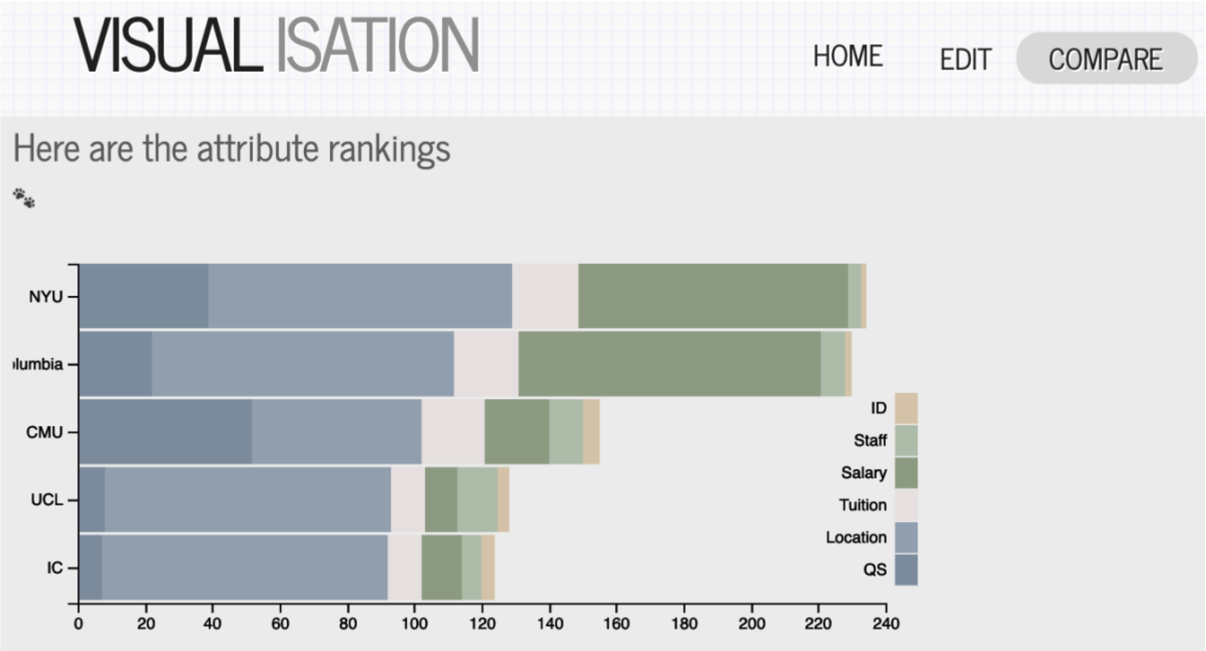
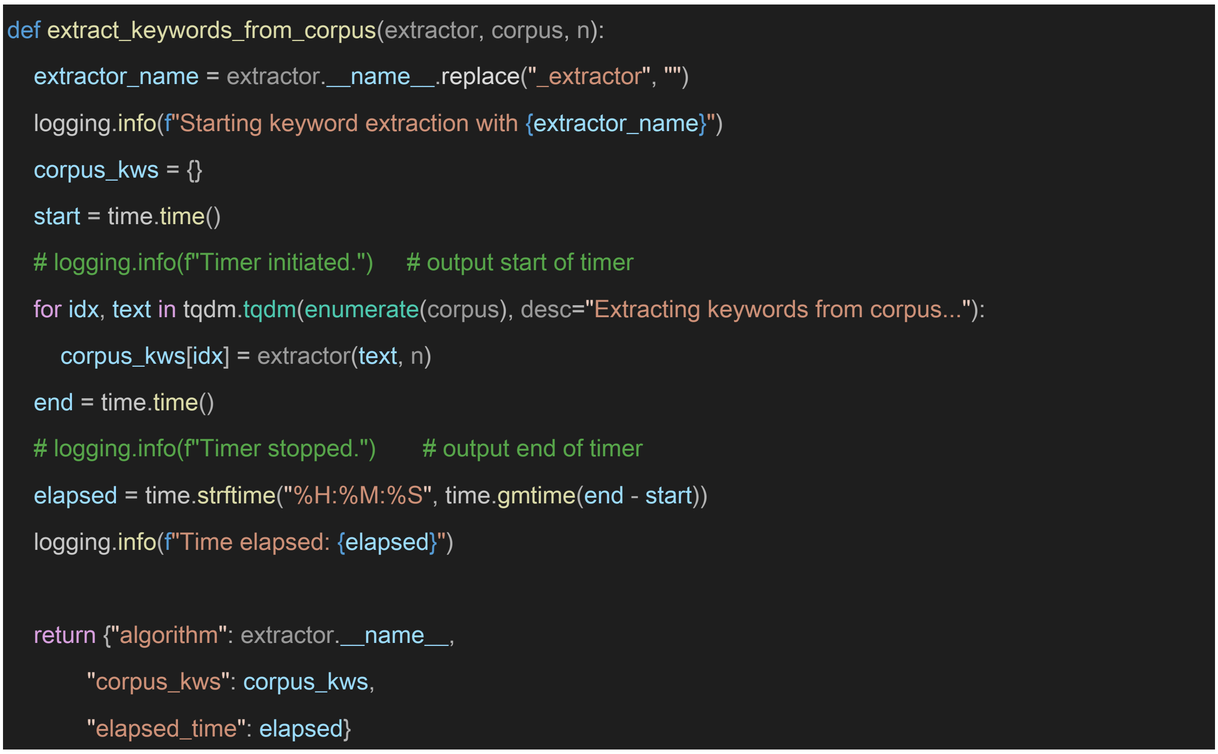


Figure 7 – Visualisation: Stacked Bar Chart

8. Evaluation

8.1 Keyword Extraction Method Evaluation

A benchmark function is used to assess keyword extraction algorithms' effectiveness. Specifically, a benchmark function is defined to accept two inputs: the corpus and a Boolean value to indicate whether the data should be shuffled.



```
def extract_keywords_from_corpus(extractor, corpus, n):
    extractor_name = extractor.__name__.replace("_extractor", "")
    logging.info(f"Starting keyword extraction with {extractor_name}")
    corpus_kws = {}
    start = time.time()
    # logging.info(f"Timer initiated.")      # output start of timer
    for idx, text in tqdm.tqdm(enumerate(corpus), desc="Extracting keywords from corpus..."):
        corpus_kws[idx] = extractor(text, n)
    end = time.time()
    # logging.info(f"Timer stopped.")      # output end of timer
    elapsed = time.strftime("%H:%M:%S", time.gmtime(end - start))
    logging.info(f"Time elapsed: {elapsed}")

    return {"algorithm": extractor.__name__,
            "corpus_kws": corpus_kws,
            "elapsed_time": elapsed}
```

Code snippet - extract_keywords_from_corpus

The extract_keywords_from_corpus function is then invoked for each extractor, resulting in a dictionary that contains the respective extractor's outcome. This value is subsequently stored in a list. For every algorithm within the list, the average number of extracted keywords and matched keywords are computed. (figure 8) All data is consolidated in a Pandas DataFrame and subsequently exported for further analysis.

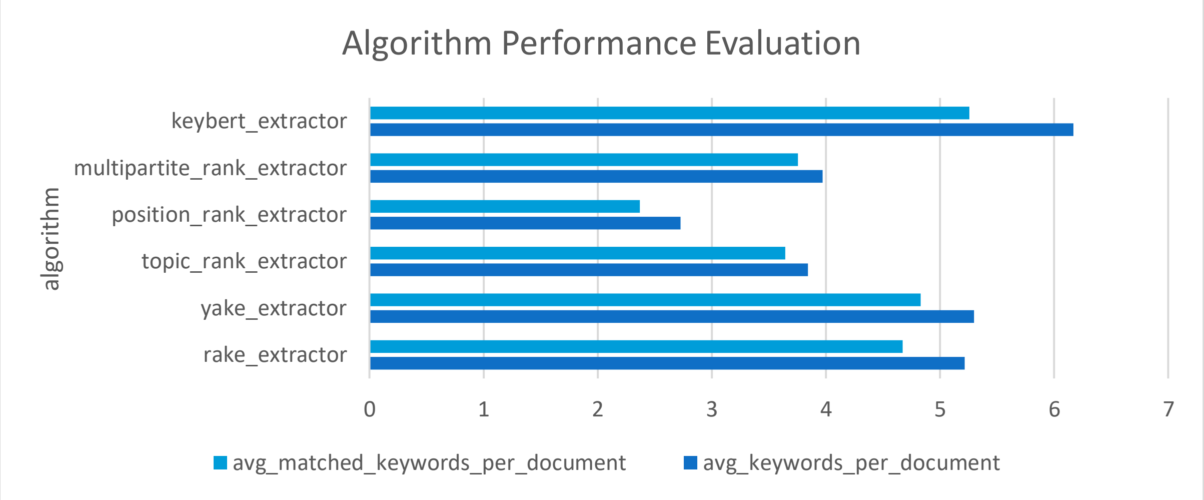


Figure 8 – Keywords Extraction Algorithm Performance Evaluation

According to benchmark studies, KeyBERT outperforms other models such as RAKE, YAKE, TopicRank, PositionRank, and MultipartiteRank. This is attributed to KeyBERT's use of pre-existing labelled data to learn the underlying semantic structure of the text and identify the most relevant keywords based on that information. Consequently, KeyBERT achieves better accuracy and relevance to identify identifying the most important keywords in a given text. Therefore, the keyword list with rank and scores visualised in Word Cloud is generated by KeyBERT.

8.2 User Experience

This evaluation section aims to analyse the results of a user evaluation questionnaire distributed to 46 participants as a way of gaining insights into their experience with the CloudChoice extension.

The results of the user evaluation questionnaire were largely positive, with the majority of participants finding the system easy to understand (31 participants) and navigate (29 participants). These findings suggest that the CloudChoice interface is intuitive and user-friendly, an essential aspect of any successful software application.

Furthermore, 25 participants indicated that they would use the extension in daily tasks if it was released. This response is significant, as it suggests that the CloudChoice extension has the potential to become a popular tool for individuals who require web-scraping and keyword visualisation in their daily work.

However, the user evaluation questionnaire also revealed some areas for improvement. A few participants reported technical difficulties while using the extension, and some suggested that the interface could be further refined to enhance the user experience. The current approach’s principal limitation relates to the lack of real-time web-scraping, which necessitates manual operations in the terminal while conducting the user experience survey. Addressing this limitation is crucial and should be prioritized in future work as a primary area for improvement.

The sample size of this study is relatively small, which may limit the findings’ generalisability . Moreover, the questionnaire only focused on user experience and did not examine the extension's effectiveness in achieving its intended purpose. Further research is necessary to validate these findings and examine other aspects of the CloudChoice extension's functionality.

9. Summary and Reflections

9.1 Project Management

Over the course of the project it became clear that the work load proposed was ambitious, with many long nights required to meet milestones at the times set out in the plan. That said, the final chrome extension application produced as a result of this has met the initial expectations, providing the number of features planned. It has also already found application in part of a personal project , which in turn was one of the original motivations to create the visualisation tool.

9.1.1 Motivation and Drive

At times, it was difficult to maintain motivation, especially when faced with tasks that did not take into account variations in the time needed to complete them. Previous experience has shown that losing motivation can lead to project failure and so it is important to have a solution in place to deal with a loss of motivation. In Data Collection process, it is mentioned that this stage took longer than anticipated to return to the project in January 2022 due to the workload and other coursework deadlines, which made it seem that the project could not be completed. However, my development methodology helped to break down the workload into manageable tasks and lessened the impression of impossibility, helping to restore motivation. More work was required to make up for lost time, which was taken into account when breaking down the remaining workload.

Apart from depending on the development methodology, I stick to the deadlines set by the University. Racing against the clock and working under pressure were stimulating factors for them, and the deadlines acted as a great motivator. If not for these deadlines, the project would have probably lagged behind schedule. I used a similar strategy during supervisor meetings. They planned smaller demonstrations for almost completed features to spur them on towards completion, as a preemptive measure to tackle potential roadblocks.

9.1.2 Time Management

To meet deadlines as set out for this project, both by the plan and the University, the ability to self-manage was critical to success. To achieve this, a balance needed to be struck between external commitments and tasks to complete to allow elective time management, along with the need to maintain the motivation and drive required to complete work over the duration of the project. Of these, the latter was perhaps the most difficult to manage.

At the start of the semester in October, a project time management plan was proposed with Literature Search and Review period and three main iterative work periods. Reaching the end of the project, a final recorded timeline is produced. See the attached table for updated progress.

Task Name	**Start Date**	**Duration**	**Hours**
1. Literature Search and Review	12/10/2022	3 weeks	20 hours
Literature Search	12/10/2022	1 week	5 hours
Literature Review	19/10/2022	2 weeks	15 hours
2. Design visualisation user interface	01/12/2022	2 weeks	8 hours
3. Building user interface	15/12/2022	18.5 weeks	75 hours
Build front-end structure	15/12/2022	4 weeks	15 hours
Build word-cloud	01/02/2023	10 days	15 hours
Build data table interface	14/02/2023	10 days	15 hours
Build attrib ranking interface	01/03/2023	24 days	30 hours
4. NLP			
Investigate NLP models	19/11/2022	2 weeks	8 hours
Design NLP model	10/12/2022	2 weeks	15 hours
Apply NLP model	30/12/2022	2 weeks	10 hours
5. Data management			
Data Collection	10/01/2023	2 weeks	10 hours
Data Pre-processing	30/01/2023	2 weeks	10 hours
6. Interim report	17/01/2023	2 weeks	15 hours
Integrate NLP model to the interface	20/02/2023	2 weeks	15 hours
7. Correction and optimizing	27/03/2023	10 days	25 hours
8. User Evaluation	06/04/2023	10 days	
9. Final Report	10/04/2023	14 days	25 hours

According to the work plan, literature search and review has been successfully conducted under the instructions of my supervisor Kai Xu.

During the first semester, I acquired knowledge about the fundamental constituents of a machine learning model, which was a completely new domain. The current project involves the development of a machine learning model along with NLP libraries in Python, to generate the Word Cloud section. This section will allow users to select relevant keywords.

The development of the front-end structure commenced in November, and the integration of the machine learning model can only be carried out after completing the interface structuring. Hence, both the processes can be performed simultaneously.

To ensure the smooth progress of the project, supervisor meetings have been scheduled bi-weekly and the minutes of these meetings recorded in Apple Notes for reference. Frequent informal discussions have been held amongst peer students to avoid duplication of effort and to promote better understanding among different research directions.

Front-end structure was completed in February, including any substantive content and the actual word cloud. NLP models were designed and integrated into CloudChoice in March.

Generally, the project went as planned with small changes and a postponed finish date for building the front-end interface. This delay did not significantly impede the progression of the overall process, as there remained ample time to make up for lost time during subsequent stages.

9.1.3 Gantt Chart

The endeavor progressed largely as intended, with a Gantt chart serving as a tool to monitor and record advancements made throughout the course of the year. (figure 9)

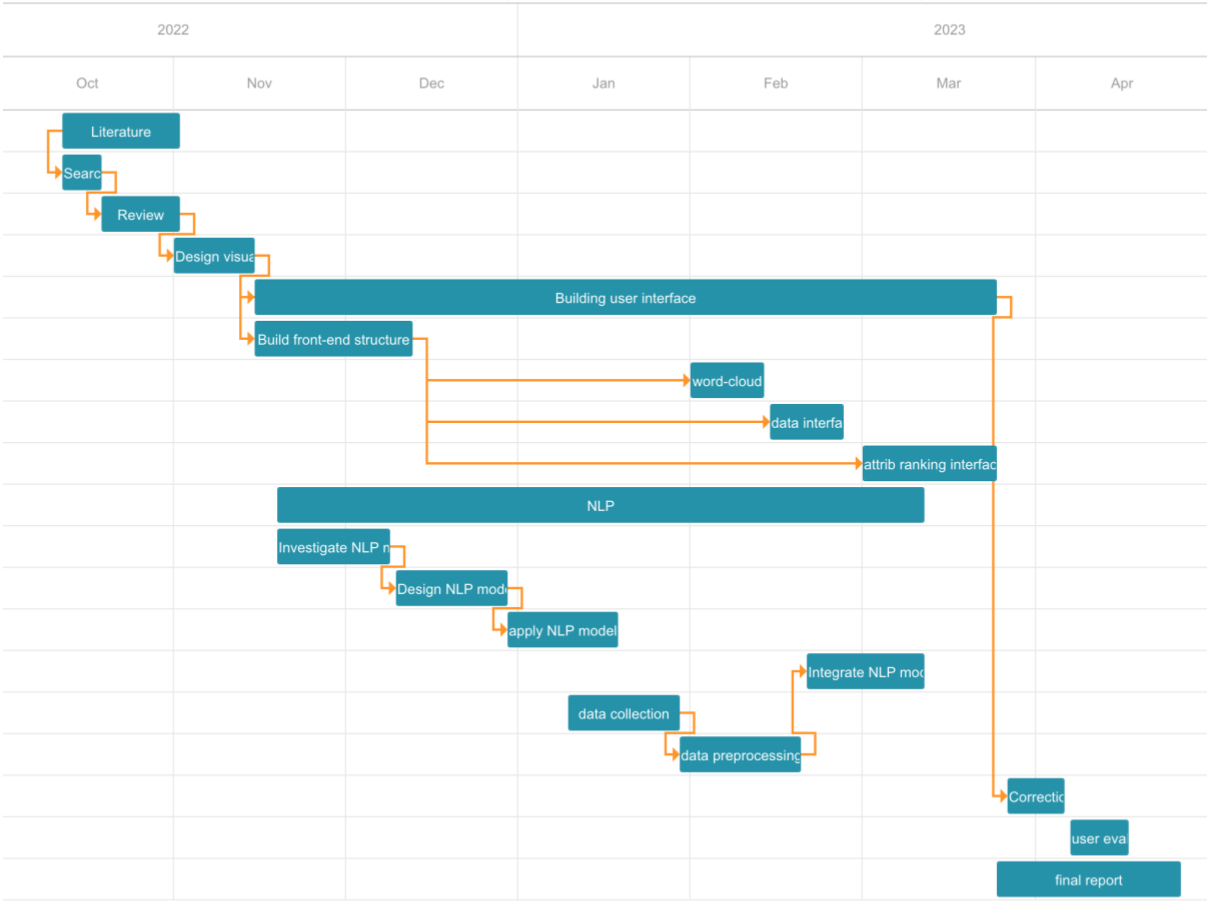
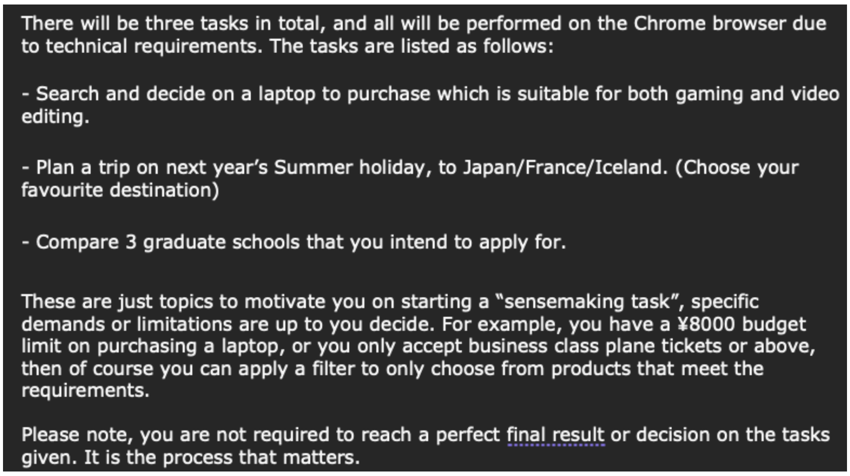


Figure 9 - Gantt Chart

9.1.4 User Data Management

Upon finishing ethics clearance, data collection was initiated to facilitate the deployment of NLP methods in this research project. To this end, three distinct use case scenarios were designed, with participants were required to perform designated tasks and their browsing history recorded under informed consent. The three use case scenarios are detailed in figure 10.

Given that this database is shared with other members, the first two tasks will be conducted in a similar manner while my current focus is primarily on the third task. To recruit participants, an email has been drafted as shown in Figure 10.



There will be three tasks in total, and all will be performed on the Chrome browser due to technical requirements. The tasks are listed as follows:

- Search and decide on a laptop to purchase which is suitable for both gaming and video editing.

- Plan a trip on next year's Summer holiday, to Japan/France/Iceland. (Choose your favourite destination)

- Compare 3 graduate schools that you intend to apply for.

These are just topics to motivate you on starting a "sensemaking task", specific demands or limitations are up to you decide. For example, you have a ¥8000 budget limit on purchasing a laptop, or you only accept business class plane tickets or above, then of course you can apply a filter to only choose from products that meet the requirements.

Please note, you are not required to reach a perfect final result or decision on the tasks given. It is the process that matters.

Figure 10 – excerpt from the email to partcipants

The userlog data is stored in OneDrive hosted by the University of Nottingham. If time and funding allow, and CloudChoice is to be developed and maintained in the future, more collected user data could be stored for further training, enabling data enhancement and performance optimisation.

9.2 Contributions and reflections

9.2.1 Innovation

As shown in Section 3, previously developed applications or extensions that have been deployed for the purpose of information retrieval and data analysis have exhibited limitations in both scope and functionality. These applications have tended to focus on use cases that are either too specific or have been designed to operate solely in the context of attribute ranking without incorporating machine learning or NLP techniques that can expedite the process of extracting meaningful insights from the collected data.

By emphasizing the importance of related attributes in sensemaking tasks, *CloudChoice* can help users make more informed decisions and avoid overloading their cognitive capacities during comparison tasks. By visualizing the relevant attributes of different search results, users are able to quickly and easily identify the most relevant options and make informed choices. This approach has the potential to revolutionize the way in which we interact with and evaluate complex data sets, particularly in domains such as business, scientific research, and information technology.

9.2.2 Computer Laws, Social, Ethical and Professional issues

The research is undertaken to facilitate users' sensemaking tasks under the context of comparing different items potentially revolutionizing the way people make decisions about commercial products, schools, accommodations, and other day-to-day choices. In designing CloudChoice, the research has taken into account not only the technological aspects but also the legal, social, ethical, and professional considerations surrounding the use of such a tool. The development of this product must be guided by the principles of data protection, non-discrimination, and respect for privacy.

From a legal perspective, CloudChoice ensures that the collection and processing of data are compliant with the data protection regulations of the UK. In addition, the research has consider the legal implications of using machine learning models in decision-making processes. In contexts such as hiring, lending, and credit scoring, where algorithmic bias and discrimination may arise, CloudChoice has avoided them from happening.

Looking into the social impacts, CloudChoice has been evaluated for its potential impact on different user groups, including those potentially vulnerable or marginalized. The research has considered how the design may affect different groups' decision-making, including those who may be less familiar with technology or who may have different cultural or linguistic backgrounds. In addition, the research has ensured that the Word Cloud design does not perpetuate or reinforce existing biases or stereotypes.

From an ethical perspective, CloudChoice has taken into consideration the Word Cloud design's broader societal implications. The design has been evaluated for its potential impact on issues such as privacy, autonomy, and transparency. The research has ensured that users are fully informed about the data collection and processing practices during data collection and that they have the ability to control their data and make informed choices about how their data is used.

Professionally, the research ensured that the Word Cloud design is developed in accordance with the ethical standards and best practices of the relevant professional community. This project has taken issues like data accuracy, validity, and reliability into consideration, as well as issues related to intellectual property and data ownership.

As an analytical tool, this design is intended for those with expertise in the field or a general interest in the subject matter, but even those without analytical experience can benefit from its usability and user-friendly interface, since it is designed with good guessability and learnability. In the meantime, there is a small chance that CloudChoice may have negative social impacts and further investigations must be carried out to avoid discrimination or personal information leakage. These issue may arise when NLP models are exposed to and used by the general public.

In terms of data management, the data is stored and deployed in the University of Nottingham Onedrive account with permitted access only. The conducted data collection process did not start until ethical approval was granted and participants are informed of consent.

9.2.3 Reflection and Future Considerations

The findings of this study indicate that, while the CloudChoice extension is a useful tool for conducting user experience surveys, there is still significant room for improvement.

Looking to the future, improvements can be made. To accommodate more complex tasks and data-intensive operations, it is critical to adopt a storage methodology that supports larger datasets and ensures heightened security measures. A principal limitation of the current approach is the lack of real-time web-scraping. Addressing this limitation is crucial and should be prioritized as a primary area for improvement in future work.

Regarding the *Compare* page visualization, it is worth noting that some of the functionalities specified were not attained as intended. For instance, users are currently unable to freely adjust weights assigned to different attributes, which renders it challenging to account for individual user tendencies. If additional time is allocated, a pop-up window should be designed to enable users to input weight adjustments for each attribute.

In addition, the sample size of this study is relatively small, which may limit the findings' generalizability. Further research with a larger sample size is necessary to validate these findings and examine other aspects of the CloudChoice extension's functionality. This could involve exploring the effectiveness of the extension in achieving its intended purpose, beyond just assessing user experience. Future research could explore the potential of the CloudChoice extension for other types of research, such as user testing, A/B testing, or other forms of data collection. This would involve testing the extension's functionality for a broader range of research scenarios.

In conclusion, while the current study provides useful insights into the use of the CloudChoice extension for conducting user experience surveys, further research is necessary to explore its potential for other types of research and to address its limitations. Upgrading or redesigning the extension to accommodate more intricate tasks and data-intensive operations should be a priority, and the adoption of a storage methodology that supports larger datasets and heightened security measures is crucial.

Bibliography

[1] K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, and J. Wenskovitch, ‘Survey on the Analysis of User Interactions and Visualization Provenance’, *Comput. Graph. Forum*, vol. 39, no. 3, pp. 757–783, 2020, doi: 10.1111/cgf.14035.

[2] D. Gotz and M. X. Zhou, ‘Characterizing Users’ Visual Analytic Activity for Insight Provenance’, p. 8.

[3] P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields, ‘SensePath: Understanding the Sensemaking Process Through Analytic Provenance’, *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 41–50, Jan. 2016, doi: 10.1109/TVCG.2015.2467611.

[4] M. Dörk, N. Henry Riche, G. Ramos, and S. Dumais, ‘PivotPaths: Strolling through Faceted Information Spaces’, *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2709–2718, Dec. 2012, doi: 10.1109/TVCG.2012.252.

[5] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, and B. L. W. Wong, ‘SenseMap: Supporting browser-based online sensemaking through analytic provenance’, in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2016, pp. 91–100. doi: 10.1109/VAST.2016.7883515.

[6] J. K. Li, S. Xu, Y. (Chris) Ye, and K.-L. Ma, ‘Resolving Conflicting Insights in Asynchronous Collaborative Visual Analysis’, *Comput. Graph. Forum*, vol. 39, no. 3, pp. 497–509, 2020, doi: 10.1111/cgf.13997.

[7] I. M. Yazici and M. S. Aktas, ‘A novel visualization approach for data provenance’, *Concurr. Comput. Pract. Exp.*, vol. 34, no. 9, p. e6523, 2022, doi: 10.1002/cpe.6523.

[8] A. Dattolo and M. Corbatto, ‘Assisting researchers in bibliographic tasks: A new usable, real-time tool for analyzing bibliographies’, *J. Assoc. Inf. Sci. Technol.*, vol. 73, no. 6, pp. 757–776, 2022, doi: 10.1002/asi.24578.

[9] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, ‘LineUp: Visual Analysis of Multi-Attribute Rankings’, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2277–2286, Dec. 2013, doi: 10.1109/TVCG.2013.173.

[10] S. Bird, E. Klein, E. Loper, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol, UNITED STATES: O’Reilly Media, Incorporated, 2009. Accessed: Dec. 07, 2022. [Online]. Available: http://ebookcentral.proquest.com/lib/nottingham/detail.action?docID=443090

[11] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, ‘A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate’, in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 338–343. doi: 10.1109/SNAMS.2019.8931850.

[12] C. Hu, this link will open in a new window Link to external site, H. Gong, this link will open in a new window Link to external site, and Y. He, ‘Data driven identification of international cutting edge science and technologies using SpaCy’, *PLoS One*, vol. 17, no. 10, p. e0275872, Oct. 2022, doi: 10.1371/journal.pone.0275872.

[13] M. Q. Khan *et al.*, ‘Impact analysis of keyword extraction using contextual word embedding’, *PeerJ Comput. Sci.*, vol. 8, p. e967, May 2022, doi: 10.7717/peerj-cs.967.

[14] ‘Puppeteer | Puppeteer’. https://pptr.dev/ (accessed Apr. 18, 2023).

[15] ‘Browserify’. https://browserify.org/ (accessed Apr. 18, 2023).

[16] B. Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, SpaCy, and Keras*. Birmingham, UNITED KINGDOM: Packt Publishing, Limited, 2018. Accessed: Dec. 08, 2022. [Online].

Available: http://ebookcentral.proquest.com/lib/nottingham/detail.action?docID=5446034

[17] S. Rose, D. Engel, N. Cramer, and W. Cowley, ‘Automatic Keyword Extraction from Individual Documents’, in *Text Mining*, John Wiley & Sons, Ltd, 2010, pp. 1–20. doi: 10.1002/9780470689646.ch1.

[18] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, ‘YAKE! Collection-Independent Automatic Keyword Extractor’, in *Advances in Information Retrieval*, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 806–810. doi: 10.1007/978-3-319-76941-7_80.

[19] A. Bougouin, F. Boudin, and B. Daille, ‘TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction’, presented at the International Joint Conference on Natural Language Processing (IJCNLP), Oct. 2013, p. 543. Accessed: Apr. 20, 2023. [Online]. Available: https://hal.science/hal-00917969

[20] C. Florescu and C. Caragea, ‘PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents’, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1105–1115. doi: 10.18653/v1/P17-1102.

[21] P. A, N. Sukiennik, and P. Hui, ‘Inflo: News Categorization and Keyphrase Extraction for Implementation in an Aggregation System’. arXiv, Dec. 10, 2018. doi: 10.48550/arXiv.1812.03781.

[22] M. A. Taher, ‘entrptaher/puppeteer-web’. Mar. 08, 2023. Accessed: Apr. 20, 2023. [Online]. Available: https://github.com/entrptaher/puppeteer-web

[23] ‘chore: remove puppeteer-web by jackfranklin · Pull Request #5750 · puppeteer/puppeteer’, *GitHub*. https://github.com/puppeteer/puppeteer/pull/5750 (accessed Apr. 20, 2023).