**NLP-based Course Recommender System**

**Report**

**Team name: essentials6**

**Names of team members:**

**Tian, Jianmeng; Chen, Sihan; Song, Hongfeng; Wang, Xiaoyi; Wei, Yinan; Wu, Boshi**

**IT usernames:**

**ml212jt; sc22sc; sc22hs; sc22xw; sc22yw2; sc22b2w**

**Background:**

The proposed project aims to develop a course recommendation system for the University of Leeds ACS (Advanced Computer Science) program, utilizing NLP (Natural Language Processing) technology to assist students in choosing the most suitable courses based on their personal background information. The system will analyze the student's background, including their research and work experience, and mastered technologies, to provide personalized recommendations for the most appropriate courses.

In recent years, NLP technology has gained significant popularity in various fields, including education, where it is being utilized for student performance analysis, plagiarism detection, and automated essay grading, among other things. Moreover, several studies have been conducted to investigate the potential of NLP technology in developing course recommendation systems for universities worldwide. (Korab, 2020) (Li, S., 2019)

The demand for computer science is increasing across higher education. BCS' analysis in 2022 indicated that the number of applications for computer science degree programs rose by 13%, which is the largest increase among all university subjects in the UK. (Education Today, 2022)

This trend highlights the need for personalized course recommendations that can assist students in choosing the most appropriate courses.

Therefore, the proposed project aligns with the current trend in the UK and abroad, where universities are incorporating NLP technology to enhance the student's learning experience. The project's objective is to develop a course recommendation system that will aid the University of Leeds ACS program's students in selecting courses that are best suited to their interests and background.

**Contribution to knowledge:**

The proposed research on developing a course recommendation system for the University of Leeds ACS program utilizing NLP technology has the potential to make a significant contribution to the fields of educational technology and personalized learning. The system's uniqueness lies in its ability to analyze a student's personal background information, including research experience, work experience, and mastered technologies, and provide tailored recommendations for the most suitable courses.

This research can benefit the global research community by demonstrating the potential of NLP technology in creating personalized course recommendation systems that can assist students in selecting the most appropriate courses. The findings of this study can be particularly useful for universities worldwide, as they can adopt similar systems to enhance their students' learning experience.

Furthermore, this research provides opportunities to collaborate with researchers from other disciplines, such as machine learning, data mining, and information retrieval, to advance the development of more sophisticated and accurate course recommendation systems.

Additionally, the NLP-based recommendation system developed for the University of Leeds ACS program has potential practical applications beyond the education sector. For instance, the system could be modified and applied to other domains, such as job search and career development, by analyzing an individual's skills, work experience, and career aspirations and recommending suitable job opportunities and career paths.

**Importance:**

The proposed project of developing a course recommendation system for the University of Leeds ACS program based on NLP technology has the potential to contribute to current and future economic success, address key societal challenges, and aid the development of key emerging industries.

Firstly, the course recommendation system would enhance the learning experience of the University of Leeds ACS program's students, improving their academic performance and employability prospects. By selecting courses that are best suited to their interests and background, students are more likely to succeed in their academic pursuits and develop the skills necessary to enter the workforce. This, in turn, would contribute to the economic success of the region and the country by producing highly skilled and qualified graduates who can meet the demands of the emerging tech industry.

Secondly, the proposed research could address key societal challenges by increasing access to higher education and reducing the skills gap. The personalized course recommendation system would ensure that students are enrolled in courses that align with their skills and interests, increasing their chances of completing their degree programs. Additionally, by identifying the skills and technologies that are in demand, the system could encourage students to pursue careers in these areas, reducing the skills gap and contributing to the development of key emerging industries.

Thirdly, the proposed project could aid the development of key emerging industries, such as artificial intelligence and machine learning, by producing highly skilled graduates who are equipped with the necessary skills and knowledge to contribute to the growth of these industries. By identifying the skills and technologies that are in demand, the course recommendation system could encourage students to pursue careers in these areas, providing the talent pool necessary for these industries to thrive.

In conclusion, the proposed project's potential contributions to economic success, addressing societal challenges, and aiding the development of key emerging industries demonstrate its importance and relevance to the broader society.

**Research hypothesis and objectives**

Research hypothesis: The development of a course recommendation system for the University of Leeds ACS program based on NLP technology will improve the learning experience of students by providing personalized course recommendations, and contribute to the development of a more skilled and qualified workforce in the emerging tech industry.

The proposed project is novel and timely as it seeks to address the need for personalized learning and career development tools that can assist students in selecting courses that align with their skills and interests. The application of NLP technology in developing the course recommendation system represents a significant scientific ambition and has the potential for transformative outcomes in the field of educational technology.

The aim of the project is to develop a course recommendation system for the University of Leeds ACS program based on NLP technology that can analyze a student's personal background information and provide personalized course recommendations.

The measurable objectives against which the outputs, outcomes, and impacts of the work will be assessed is to develop an NLP-based algorithm for analyzing student background information and generating personalized course recommendations.

To conclude, the proposed project's aims and measurable objectives align with the research hypothesis and demonstrate the project's significance and potential transformative outcomes in the field of educational technology and career development.

**Programming and methodology**

**1.web scraping**:

First, we use web scraping techniques to extract data from the University of Leeds course catalog for each of the specified courses.

To accomplish this, we start by initializing an empty list called course\_content and another list called url containing URLs for each specific course at the University of Leeds. We then loop through the courses list and create a new URL for each course using an f-string with the course code as a variable. Finally, we append the new URLs to the url list.

Then we perform web scraping on the URLs in the `url` list, extracts various pieces of course information, and stores the information in a list of dictionaries `course\_contet`.

For each URL in the `url` list, the code sends a GET request to the URL using the `requests` library, and creates a BeautifulSoup object from the response using the `html.parser` parser. It then uses various methods of the BeautifulSoup object to extract information from the HTML of the course catalog page.

The information extracted includes the course title, summary (if available), objectives, learning outcomes, and syllabus. This information is stored in a dictionary `page\_info`, and the dictionary is appended to the `course\_contet` list. Finally, the code prints 'success' to the console to indicate that the scraping for that URL was successful.

Finally, we iterate through each dictionary in the `course\_contet` list, and prints the values of each key-value pair in the dictionary to the console. The values are printed in a specific order: title, summary, objectives, learning outcomes, and syllabus. Each dictionary is printed on a separate line, with a blank line between them.

**2. data clean:**

We can put our course data into a pandas Dataframe and then export it to a CSV file. This way, we won't need to run web scraping again. Then we loading data from a CSV file into a pandas DataFrame and then cleaning it.

The `fillna()` method is used to replace any missing values in the DataFrame with empty strings.

Next, we are combining the `title`, `summary`, `objectives`, `Learning\_outcomes`, and `syllabus` columns into a single `text` column by concatenating them with the `+` operator.

Then, we are removing HTML tags from the `text` column using a regular expression pattern that matches any string starting with `<` and ending with `>`.

After that, we are removing punctuation from the `text` column using another regular expression pattern that matches any character that is not a letter or digit.

Finally, we are converting all the text in the `text` column to lowercase letters using the `lower()` method.

Overall, these cleaning steps are designed to prepare the data for analysis by removing unwanted characters and formatting inconsistencies that could interfere with the accuracy of the analysis.

**3. Preprocess**

In this section of, we are tokenizing the text data in the `text` column by using the `nltk.word\_tokenize()` method from the Natural Language Toolkit (NLTK) library. This splits each sentence in the `text` column into individual words or tokens.

Next, we are removing stop words using the `stopwords` module from the NLTK library, which contains a list of common words that do not contribute much to the meaning of the text.

After that, we are lemmatizing the tokens using the `WordNetLemmatizer` from the NLTK library. This reduces each token to its base form, or lemma, which helps to normalize the data and reduce the number of unique words.

Finally, we are updating the DataFrame with the tokenized and lemmatized text data by adding a new column called `tokens`.

Overall, these steps help to further clean and preprocess the text data by reducing noise and redundancy, and preparing it for further analysis, such as topic modeling or sentiment analysis.

**4. Cluster**

In this section, we are using the `sklearn` library to cluster the tokenized and vectorized text data.

First, we are vectorizing the text data using the `CountVectorizer()` method, which creates a bag-of-words model of the text data by converting each document into a matrix of word counts.

Next, we are using the `KMeans()` method from the `sklearn.cluster` module to cluster the data into a specified number of clusters. We are setting the number of clusters to range from 2 to 12 using a `for` loop, and storing the resulting silhouette scores in a list.

The silhouette score measures how well each data point fits into its assigned cluster, with higher scores indicating better cluster quality.

Finally, we are visualizing the results using a line plot of the silhouette scores for each number of clusters. This allows you to choose the optimal number of clusters based on the highest silhouette score.

Overall, this process helps to identify the optimal number of clusters for the data, which is important for further analysis such as topic modeling or sentiment analysis.

We perform KMeans clustering on a dataset after reducing the dimensionality of the data using PCA. It then generates a silhouette plot and a visualization of the clustered data for different numbers of clusters.

The `PCA` class from `sklearn.decomposition` is used to reduce the dimensionality of the data to 2 dimensions. The `fit\_transform` method is used to fit the PCA model on the data `X` and transform it to 2 dimensions.

The `silhouette\_samples` and `silhouette\_score` functions from `sklearn.metrics` are used to compute the silhouette scores for each sample and the average silhouette score for all the samples, respectively. The silhouette score measures the similarity of a sample to its own cluster compared to other clusters, with values ranging from -1 to 1. Higher silhouette scores indicate better clustering results.

We then loop over different numbers of clusters and creates a subplot for each number of clusters. The first subplot shows the silhouette plot, which displays the silhouette scores for each sample and the average silhouette score for each cluster. The second subplot shows the visualization of the clustered data.

In each iteration of the loop, the `KMeans` class from `sklearn.cluster` is used to cluster the data into `n\_clusters` clusters. The `fit\_predict` method is used to fit the KMeans model on the reduced data `X\_reduced` and predict the cluster labels.

The silhouette scores are computed and plotted in the first subplot using the `silhouette\_samples` and `fill\_betweenx` functions. The cluster labels and the corresponding silhouette scores are sorted and plotted in different colors to show the silhouette scores for each cluster. The average silhouette score is also plotted as a red dashed line.

The clustered data is plotted in the second subplot using the `scatter` function. The `colors` variable is used to set the colors of the markers based on the cluster labels. The cluster centers are also plotted as white circles with black edges and labeled with their corresponding cluster numbers.

**5.Data Mining**

We generate a word cloud image using the `WordCloud` module from the `wordcloud` library. The `text` variable is a string that contains all the text data from the `data` DataFrame. The `stopwords` variable contains a set of common English words that are excluded from the word cloud.

The `WordCloud` object is created with the `stopwords`, `background\_color`, `width`, and `height` parameters. The `generate()` method is used to generate the word cloud image from the `text` data. Finally, the `imshow()` and `axis()` methods are used to display the generated image without showing the axis.

The resulting image displays the most frequently occurring words in the text data, with the size of each word proportional to its frequency.

We using bubble charts to showing the extract bigrams (pairs of two adjacent words) from the text data, the size of the bubble is the frequency of each bigram.

We are creating a bar chart of the top 10 most frequent words for each course in the data dataframe.

Finally, we calculate the Pearson correlation coefficient between all pairs of course variables in a dataset and visualizes it using a heatmap.

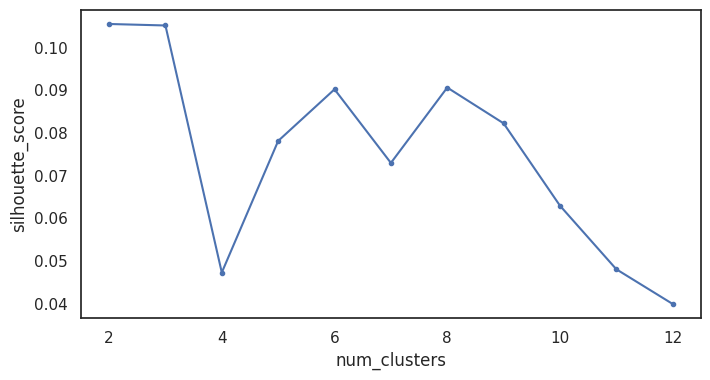
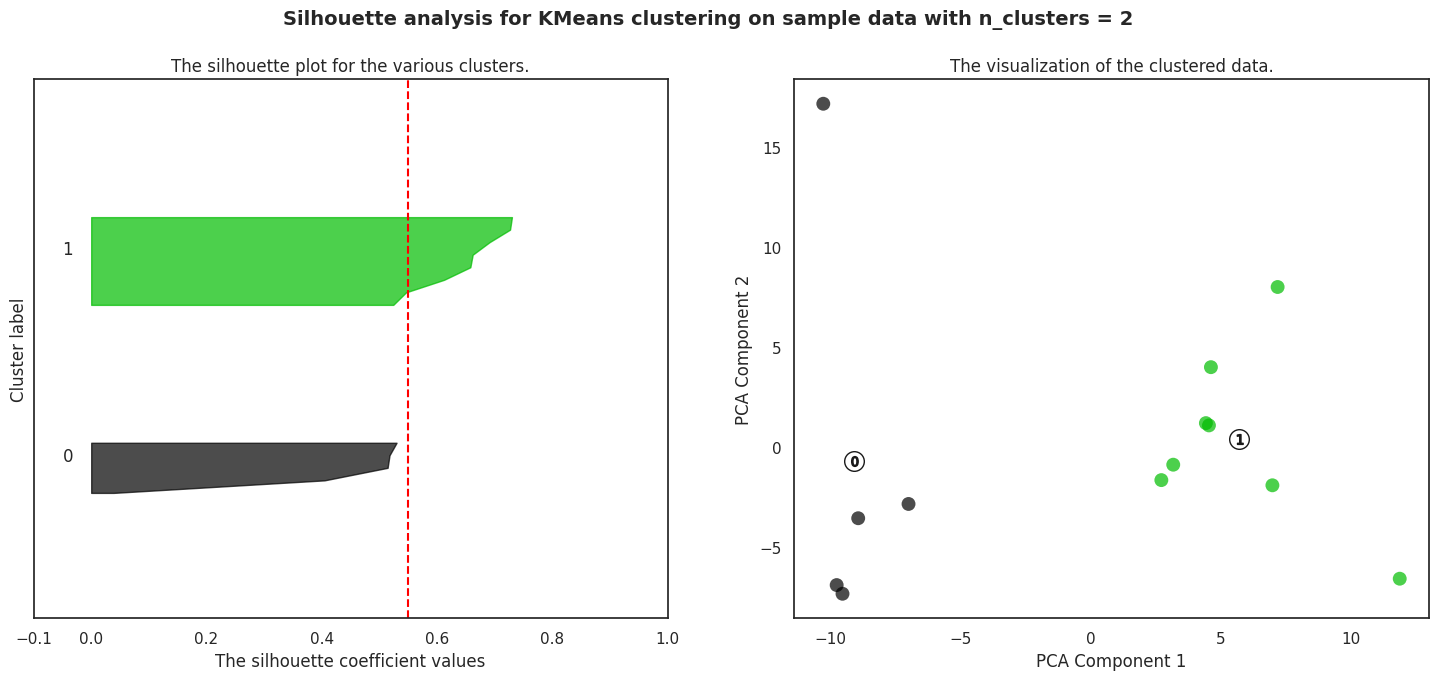
**6. Recommend**

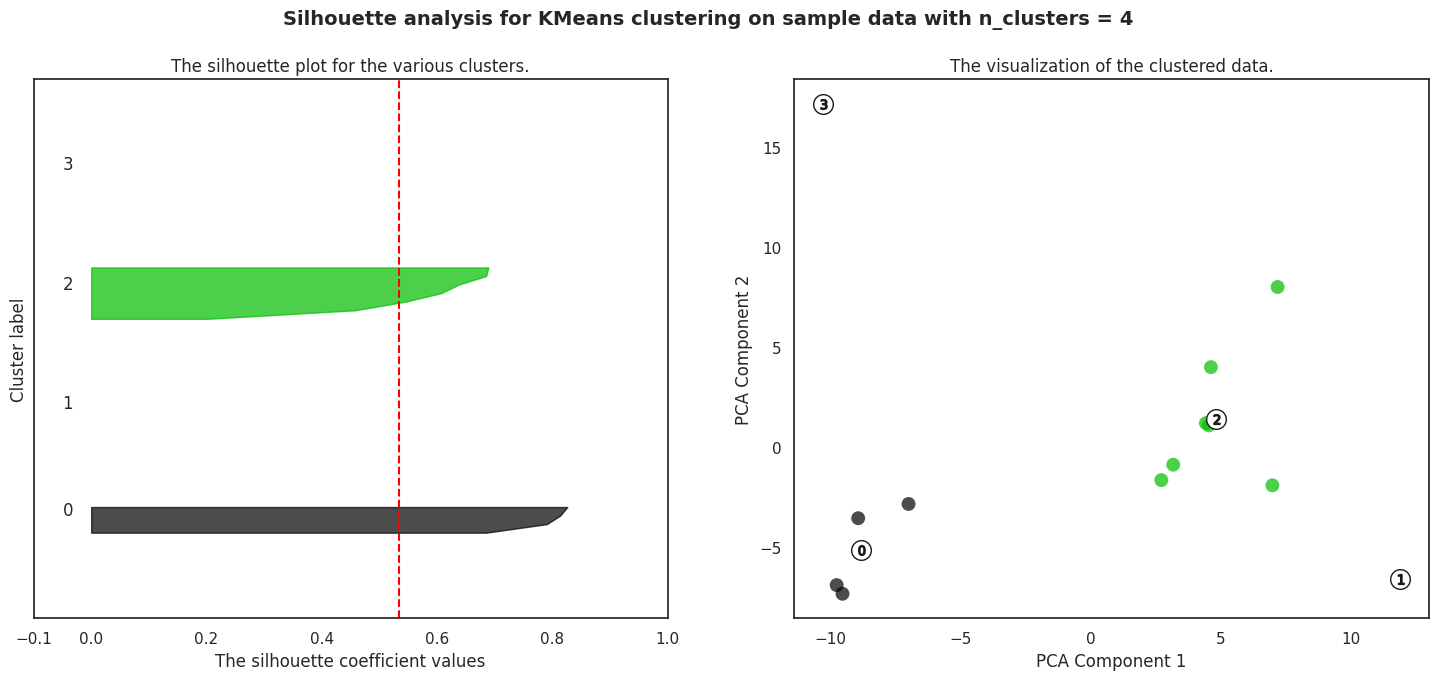
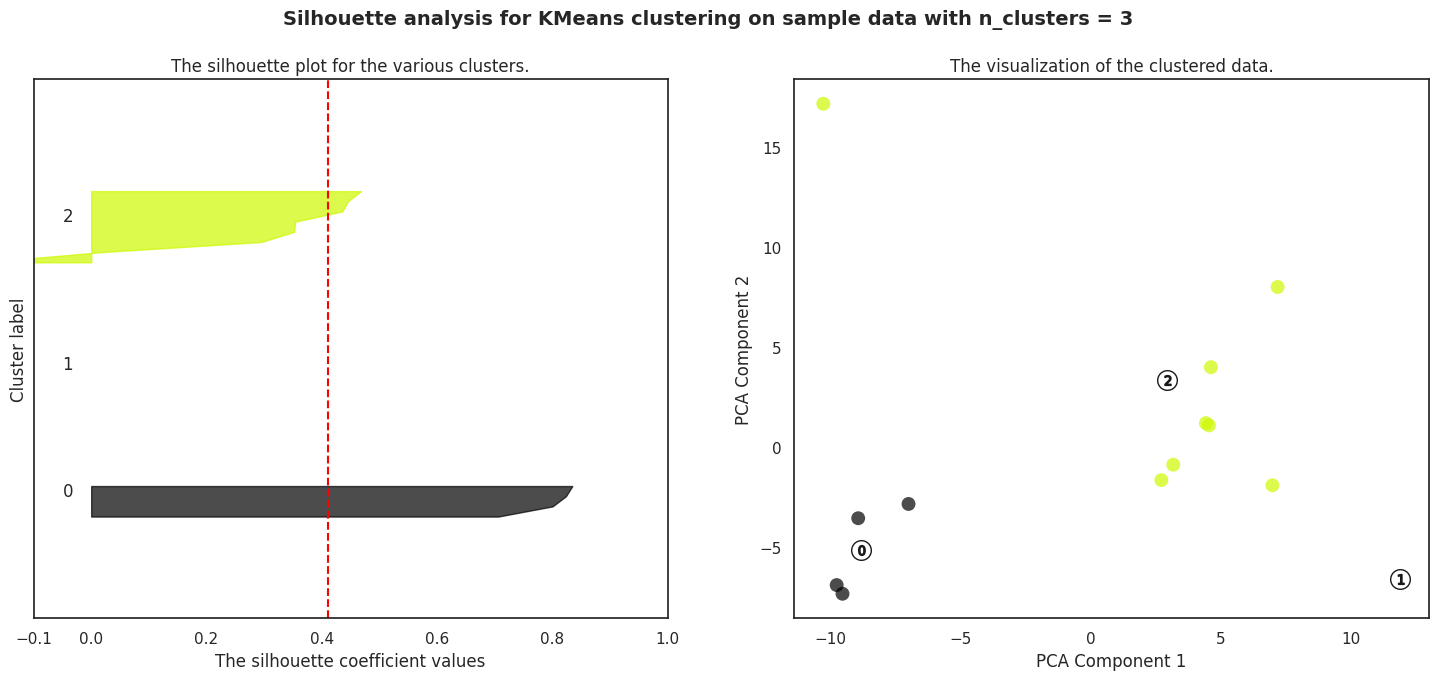
We now can take a student's input (a string) and a dataframe (with a 'title' column) as arguments. Then we use the vectorizer object (previously defined) to transform the input into a vector of word frequencies. We calculate the cosine similarity between the input vector and each course's vector in the dataframe. We add a 'similarity' column to the dataframe with the sum of cosine similarities for each course. We then select the top 3 courses based on the highest similarity scores, and returns their titles as an array. The assumption is that the courses with the highest similarity scores are the most relevant/recommended for the student.

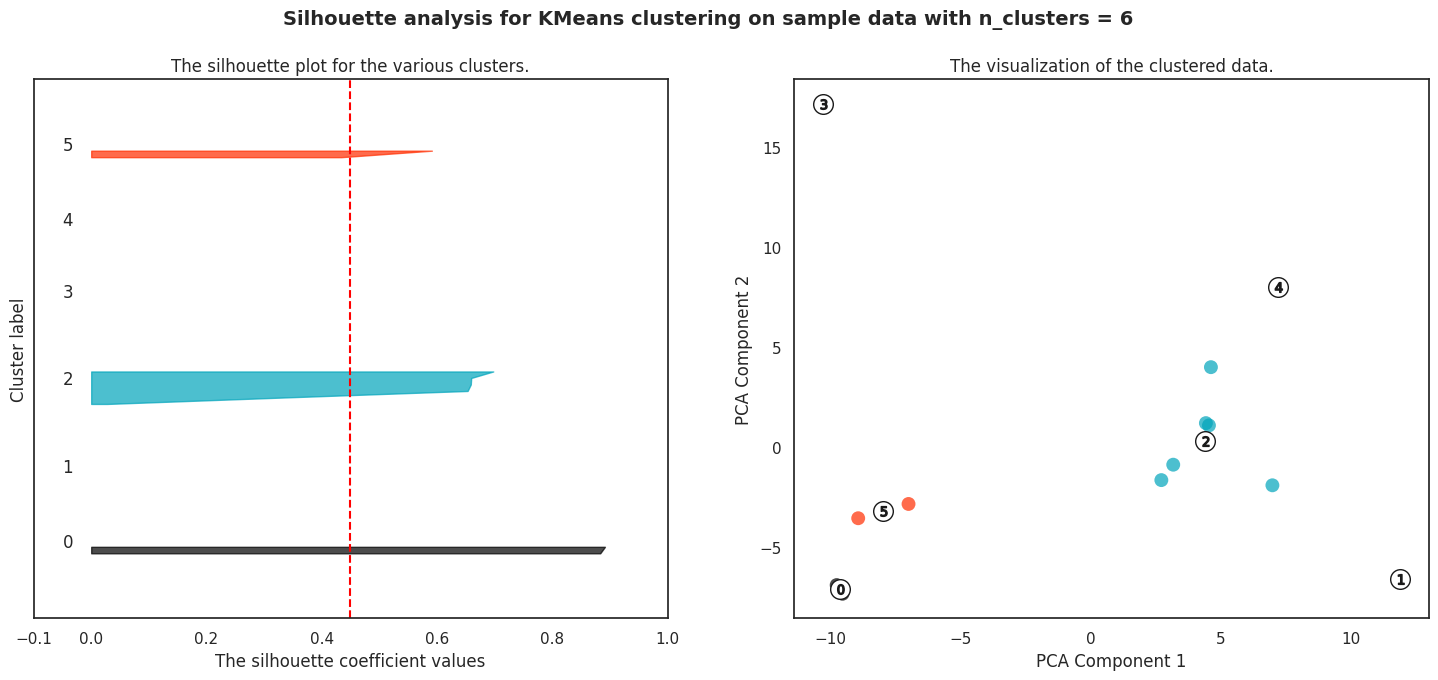
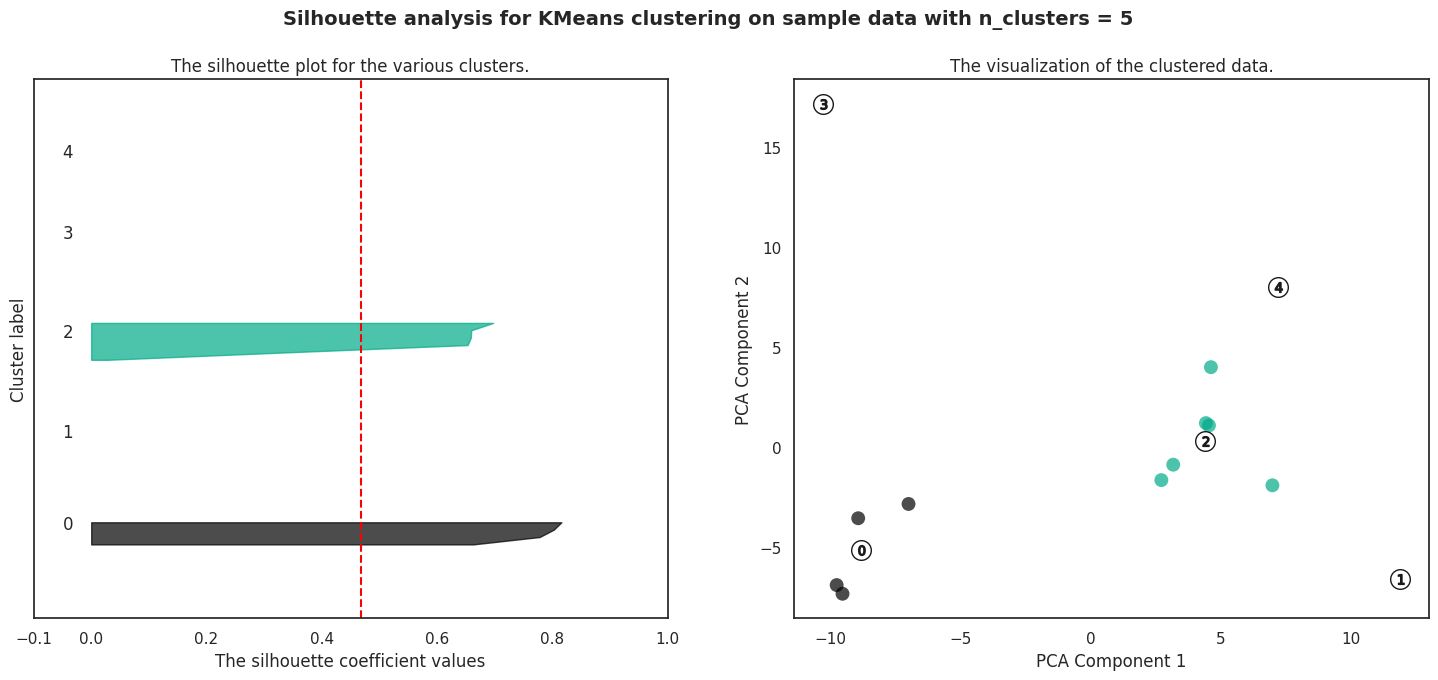
**Appendix**

1. Program outcomes and explanations (all codes on Main.ipynb and WebScraping.ipynb)

Clustering

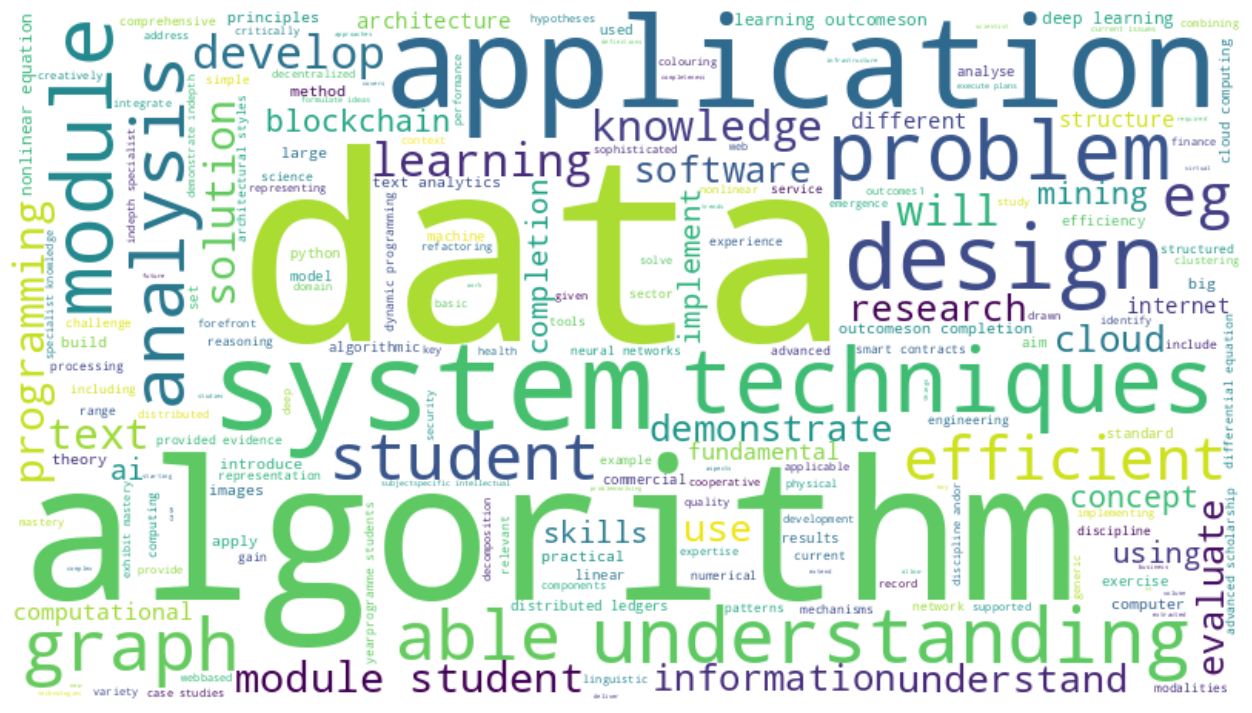
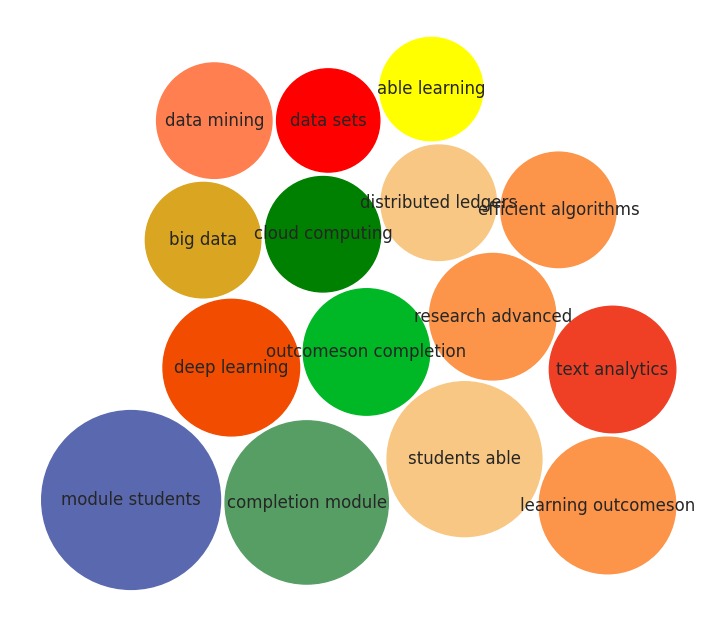
 





Based on the silhouette score plot, it seems that the number 1 has the highest silhouette score. However, upon examining the PCA cluster plot, it appears that a cluster number of 4 would be a more appropriate choice for KMeans clustering.

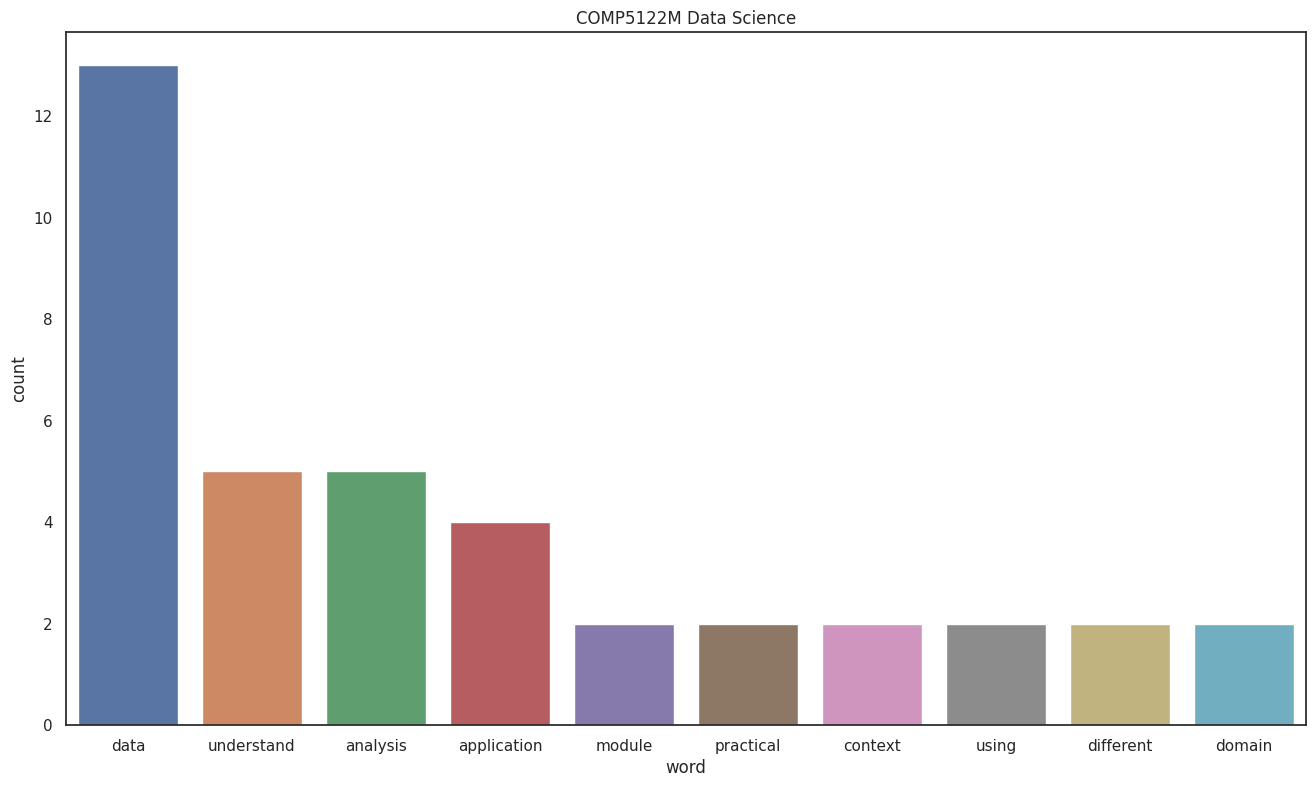
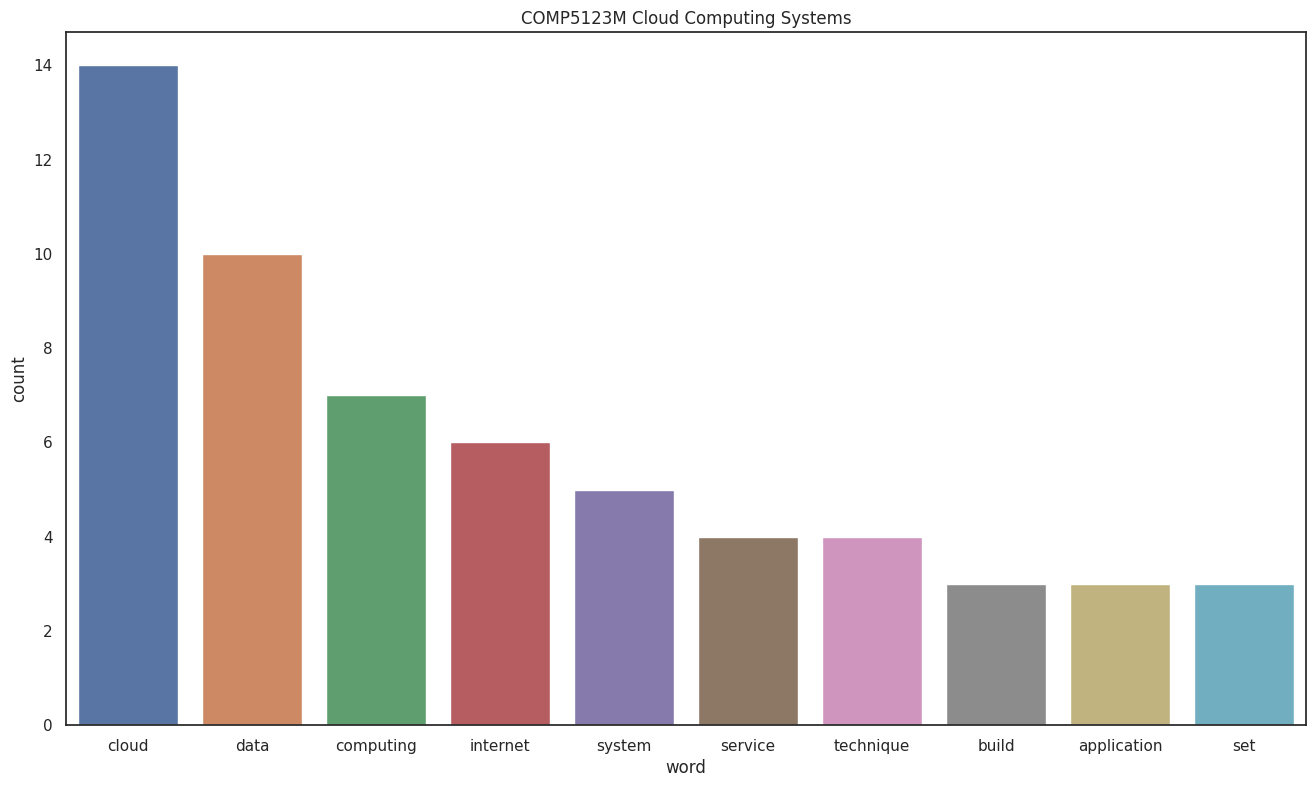
Most frequently terms

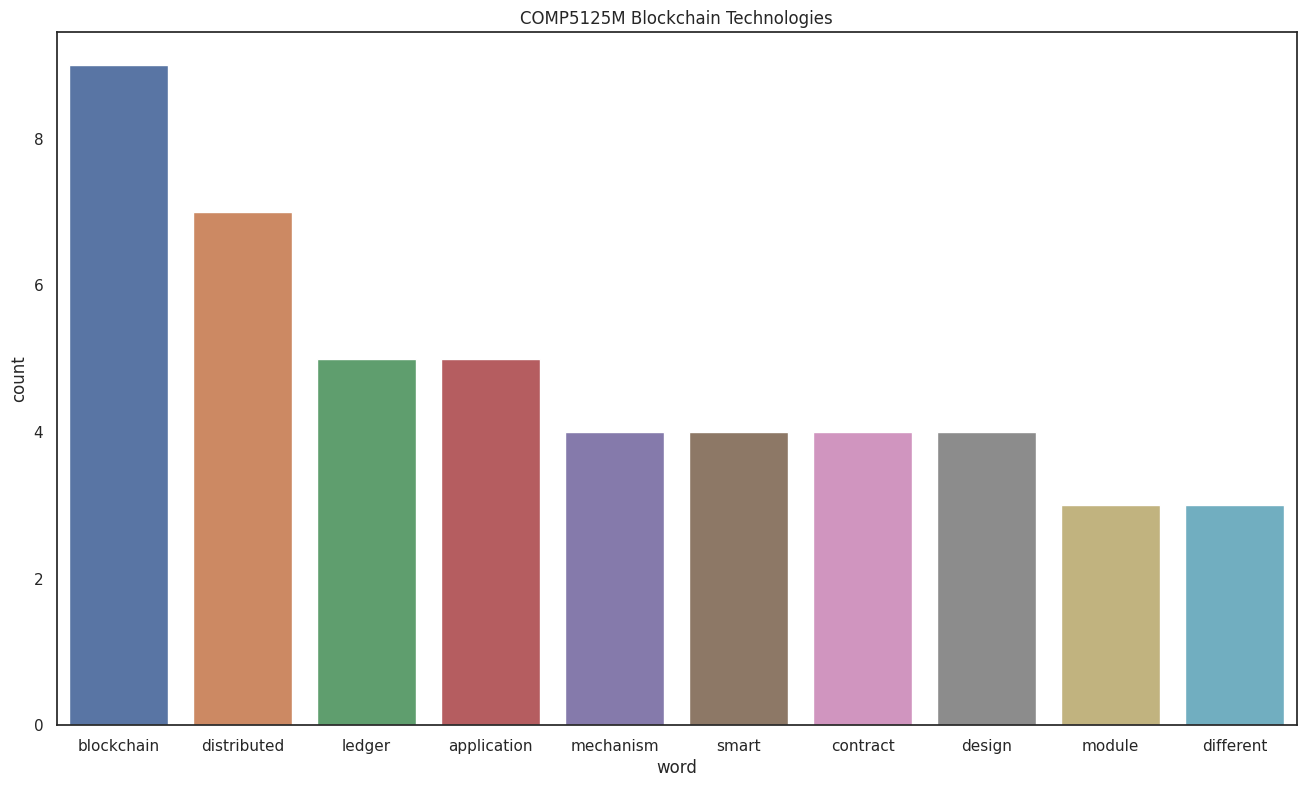
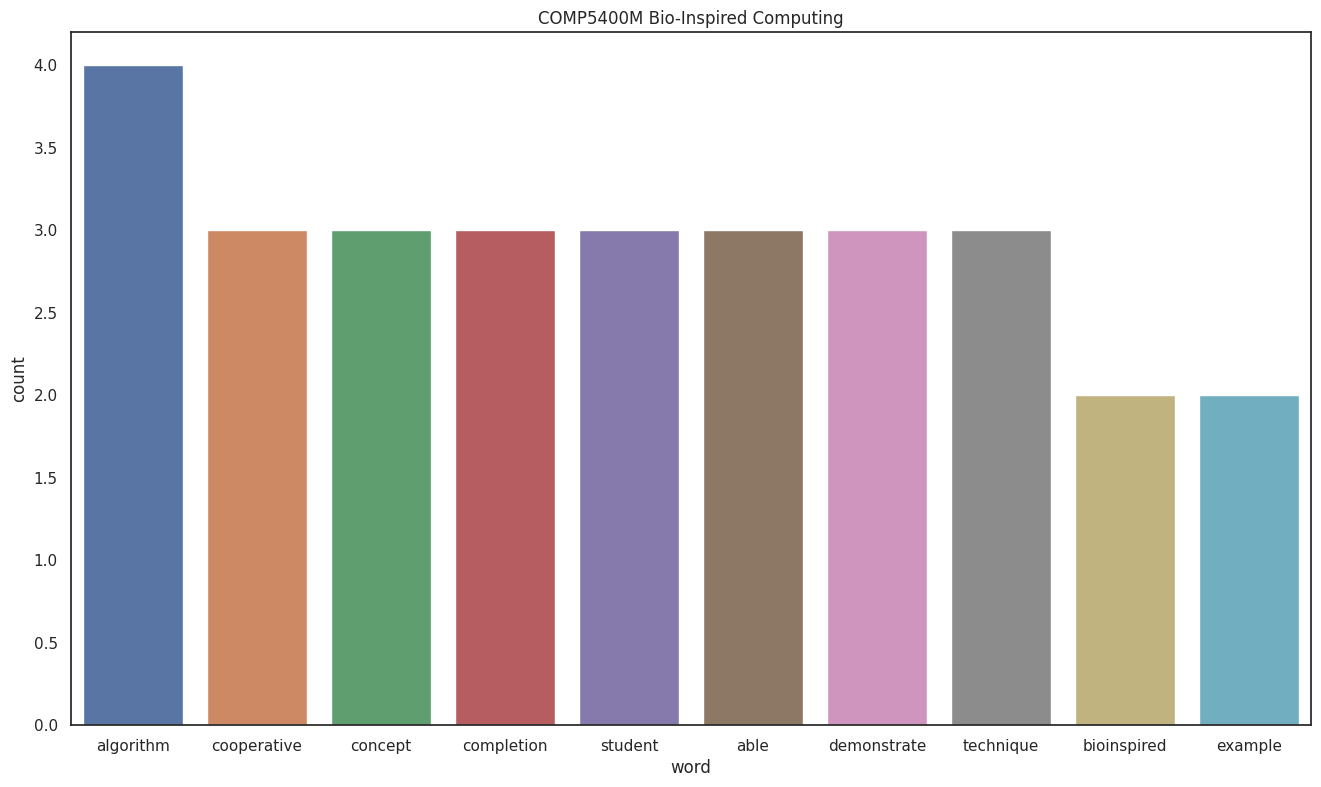
 

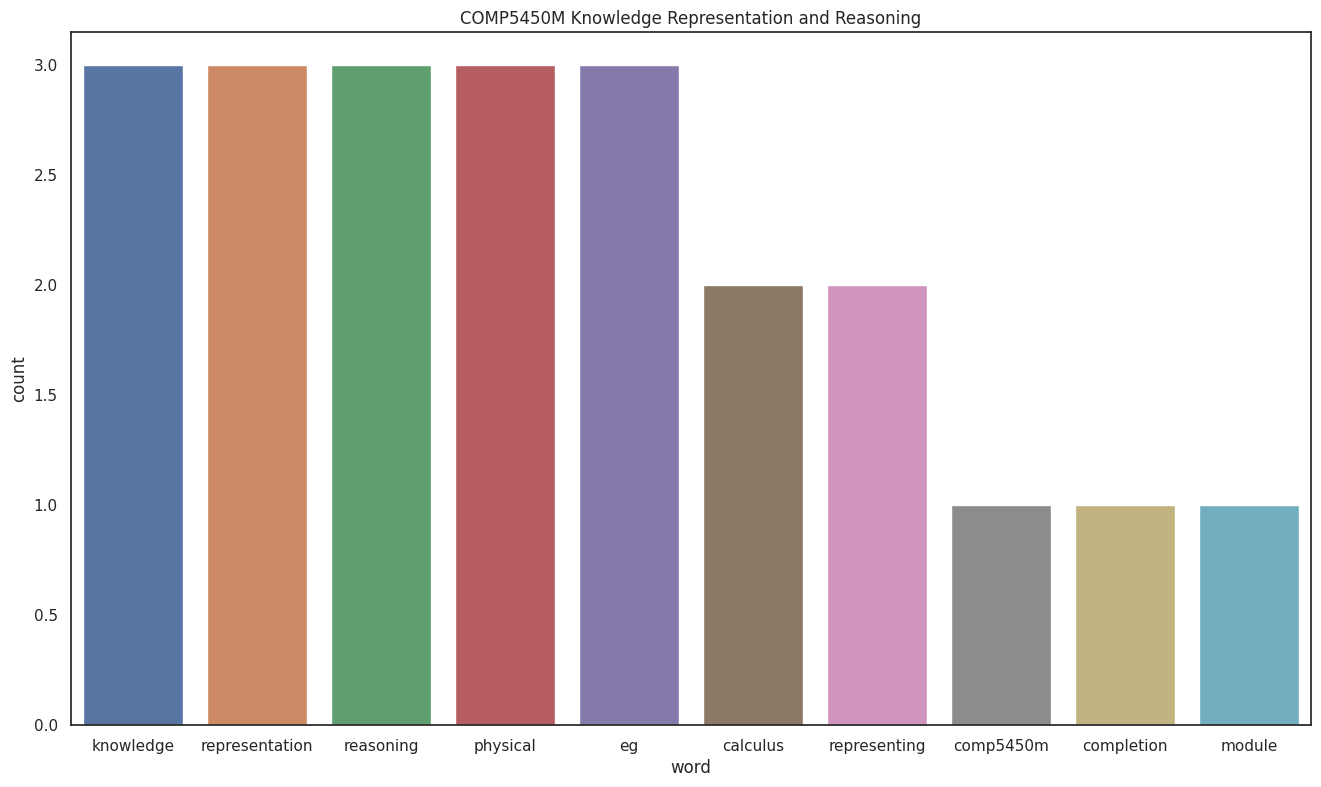
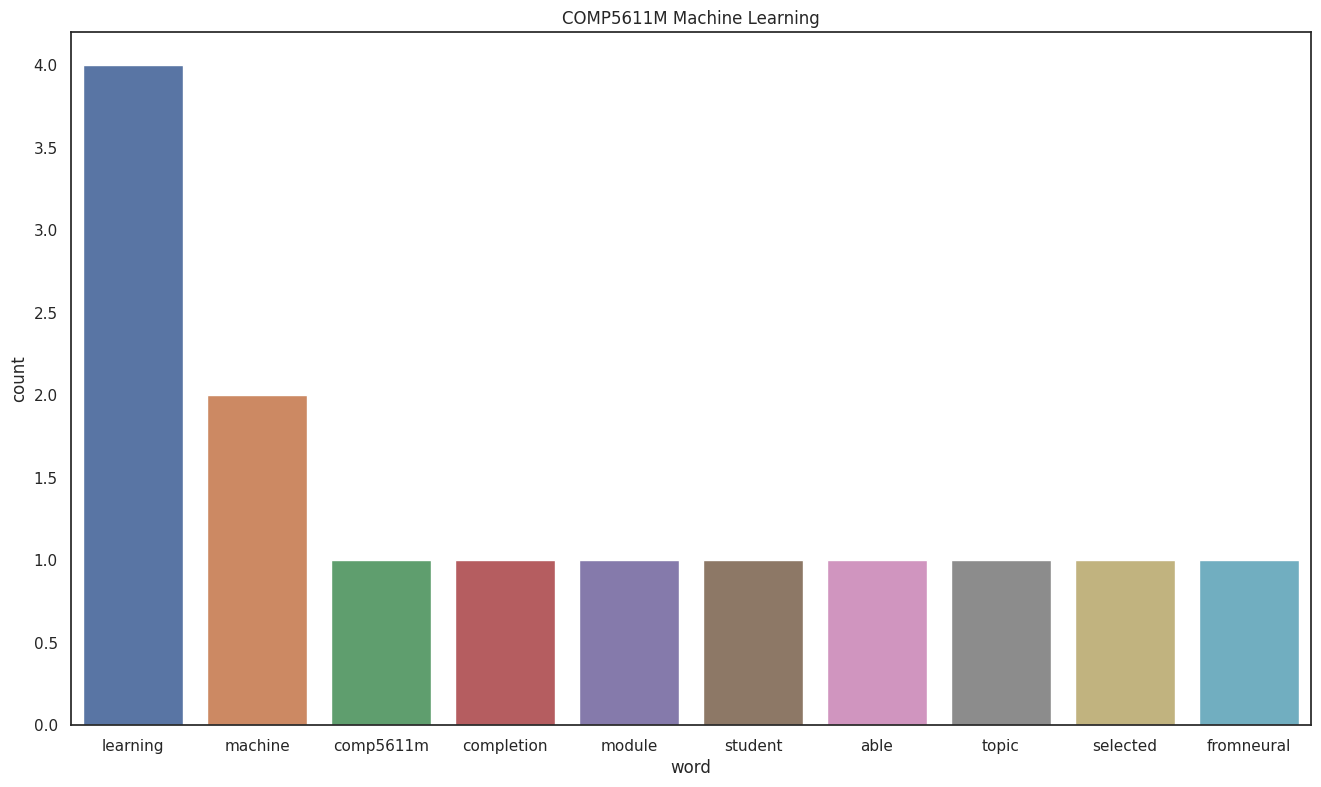
As we can see in the word cloud plot, the most frequently appearing keywords in the course are "data", "algorithm", and "application". This indicates the importance of these concepts in computer science.

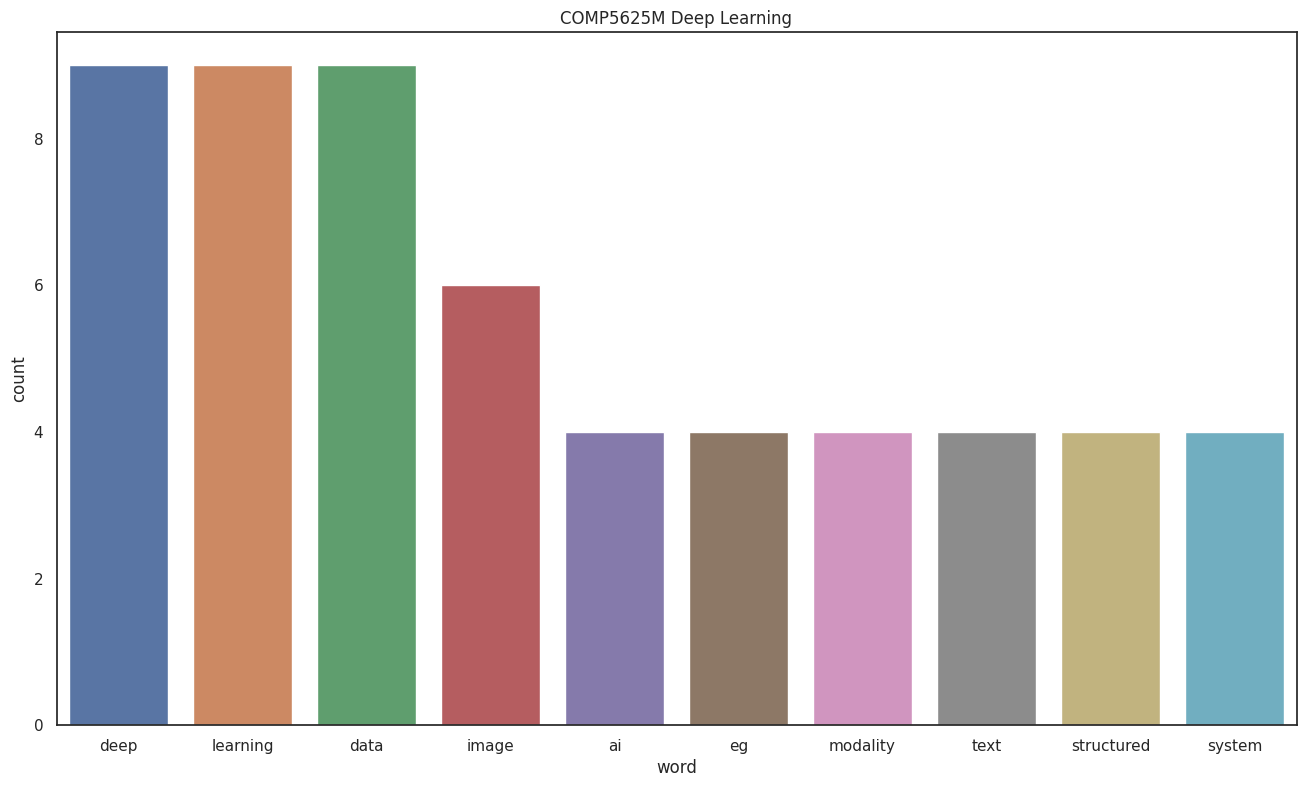
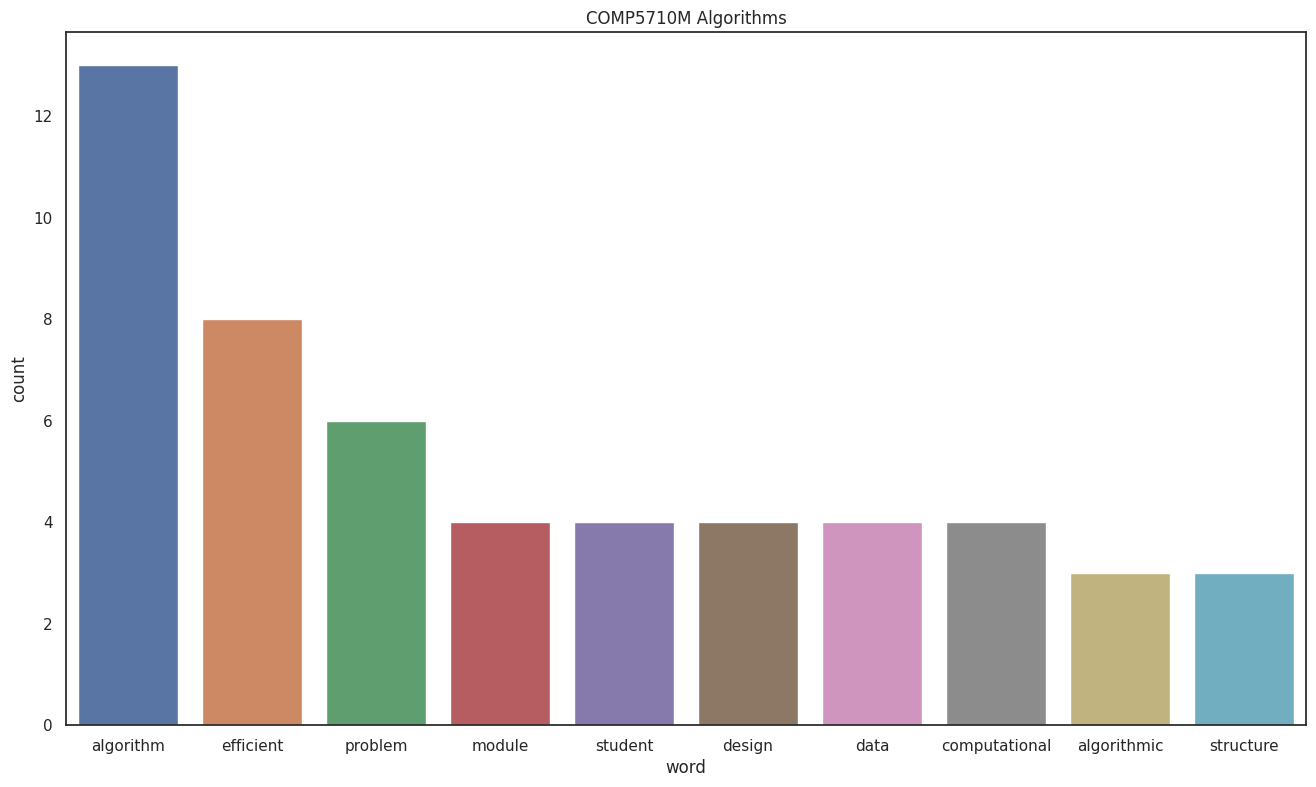
The bubble chart shows the most frequent 2-gram words. As we can see, the phrases "module students" and "completion module" appear most frequently. This suggests that the course is organized into modules.

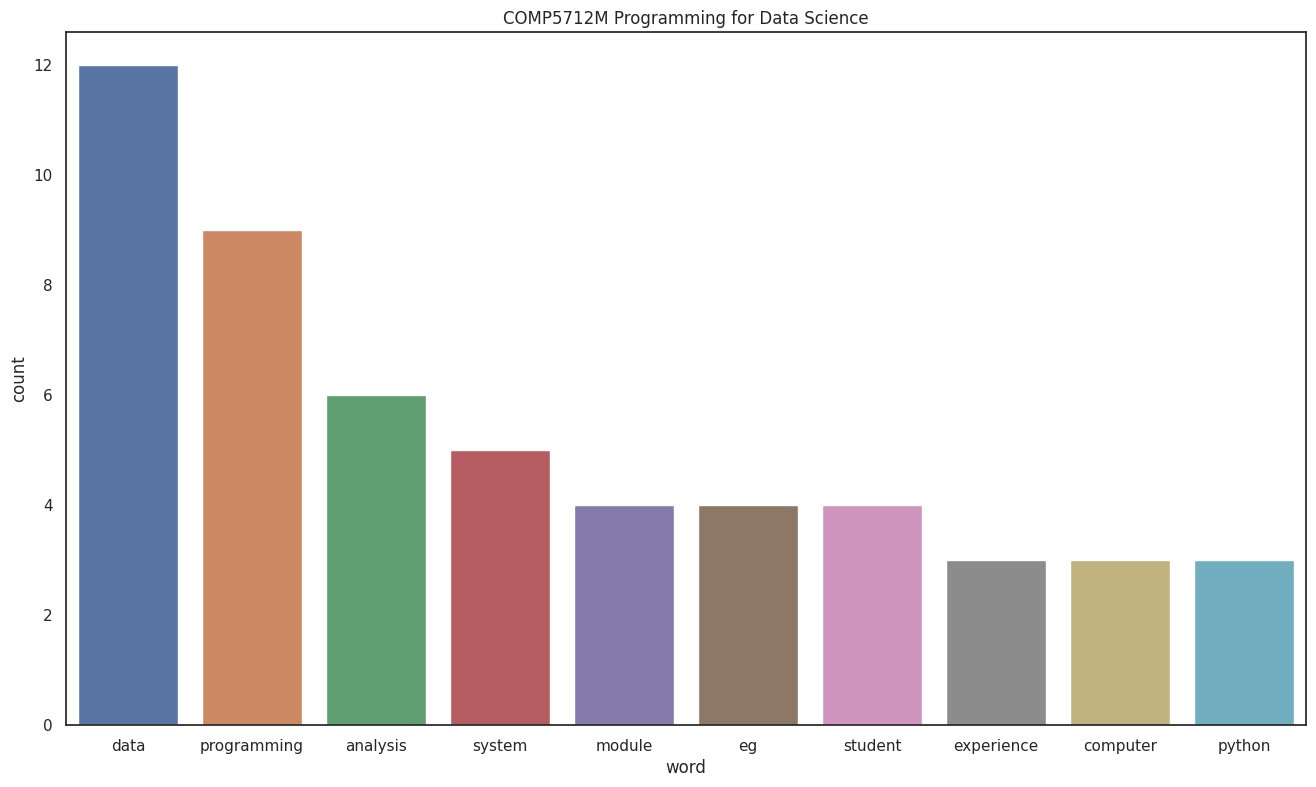
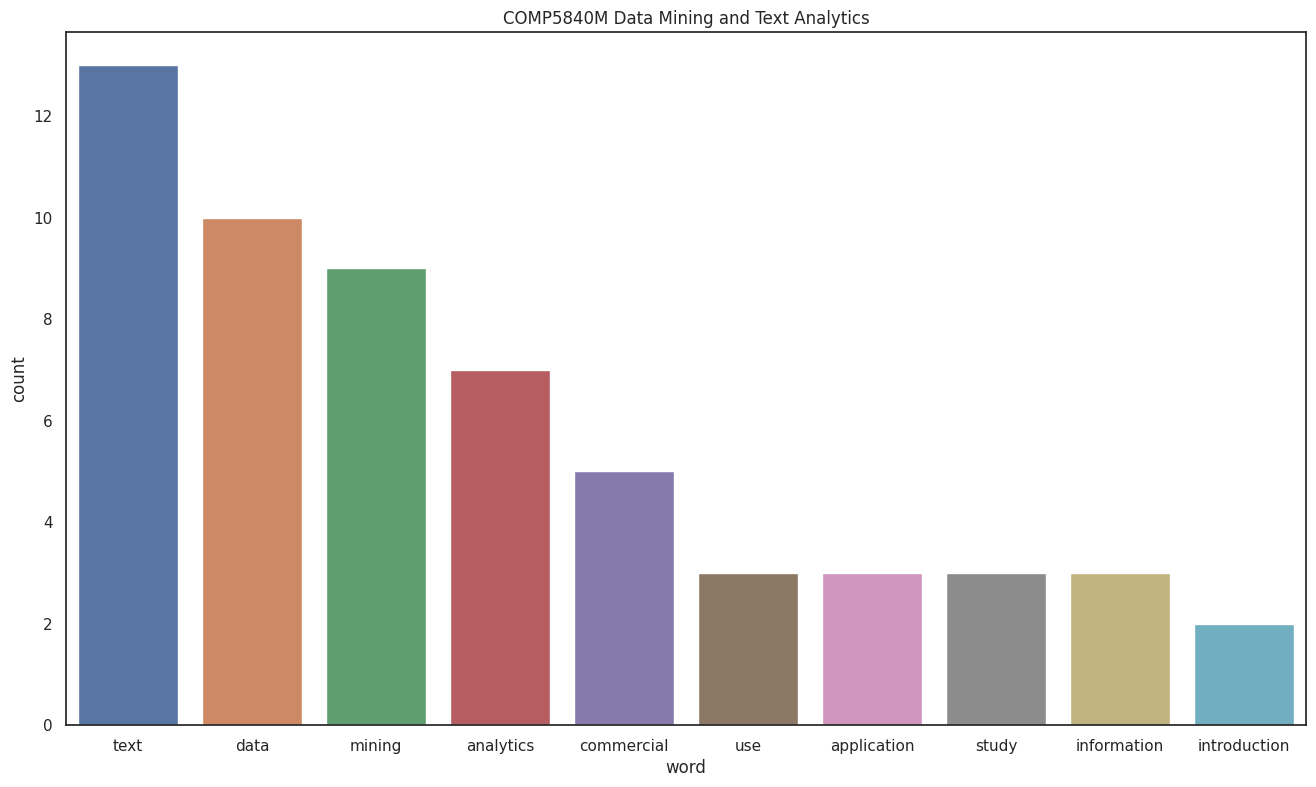
Course Relevance

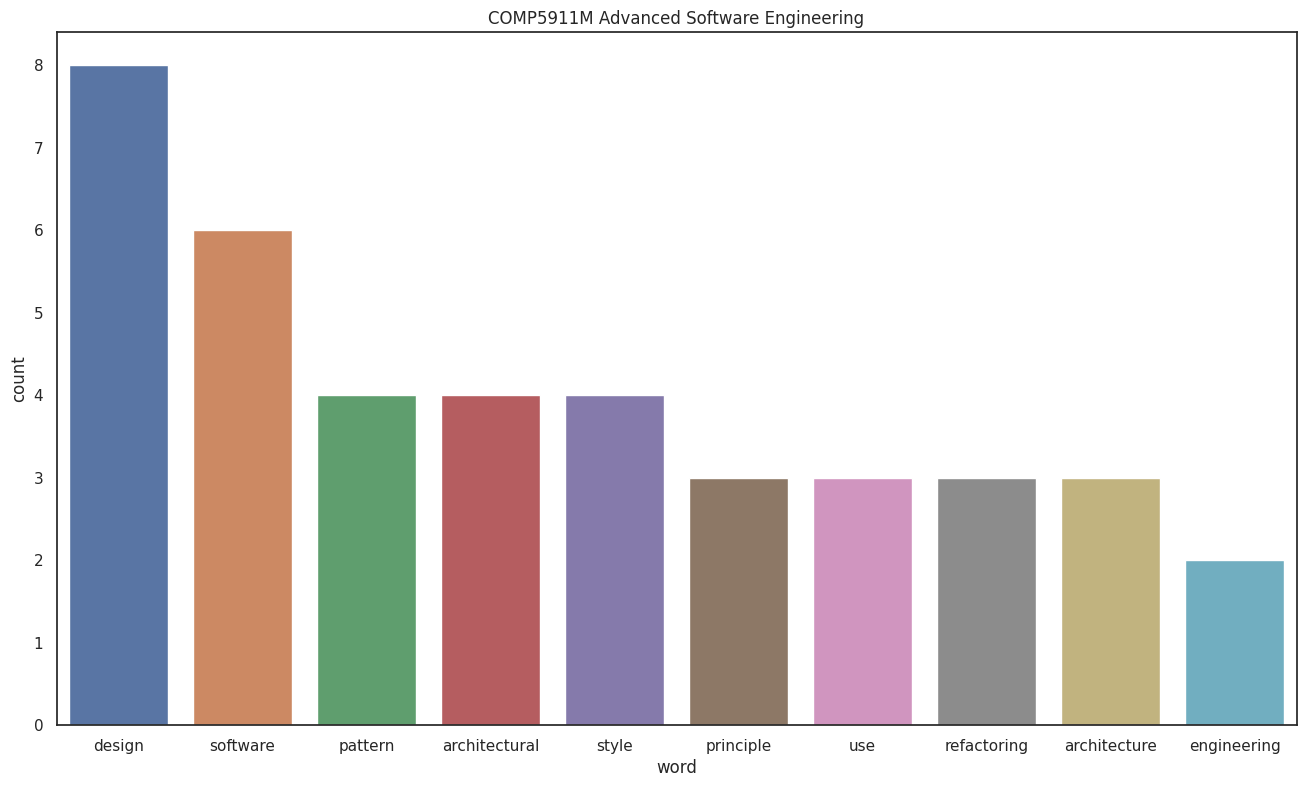
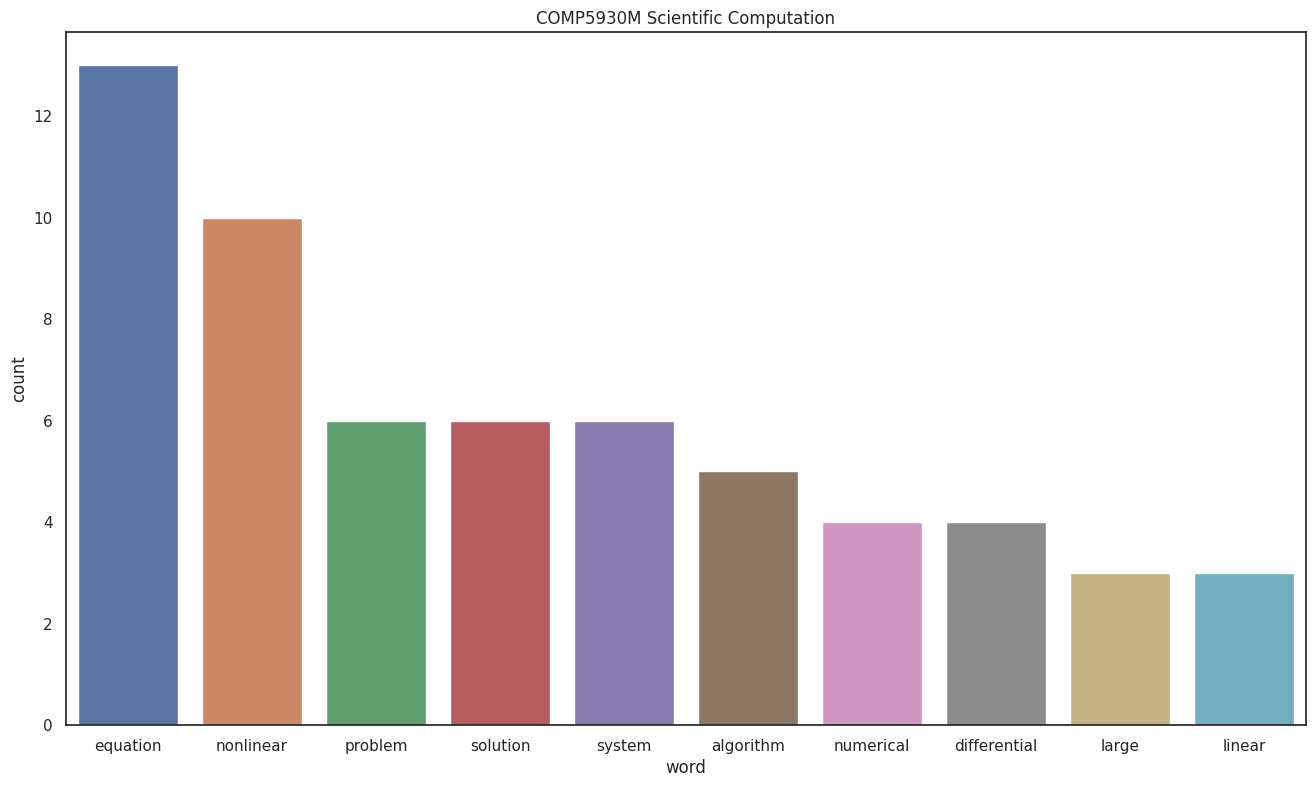
 

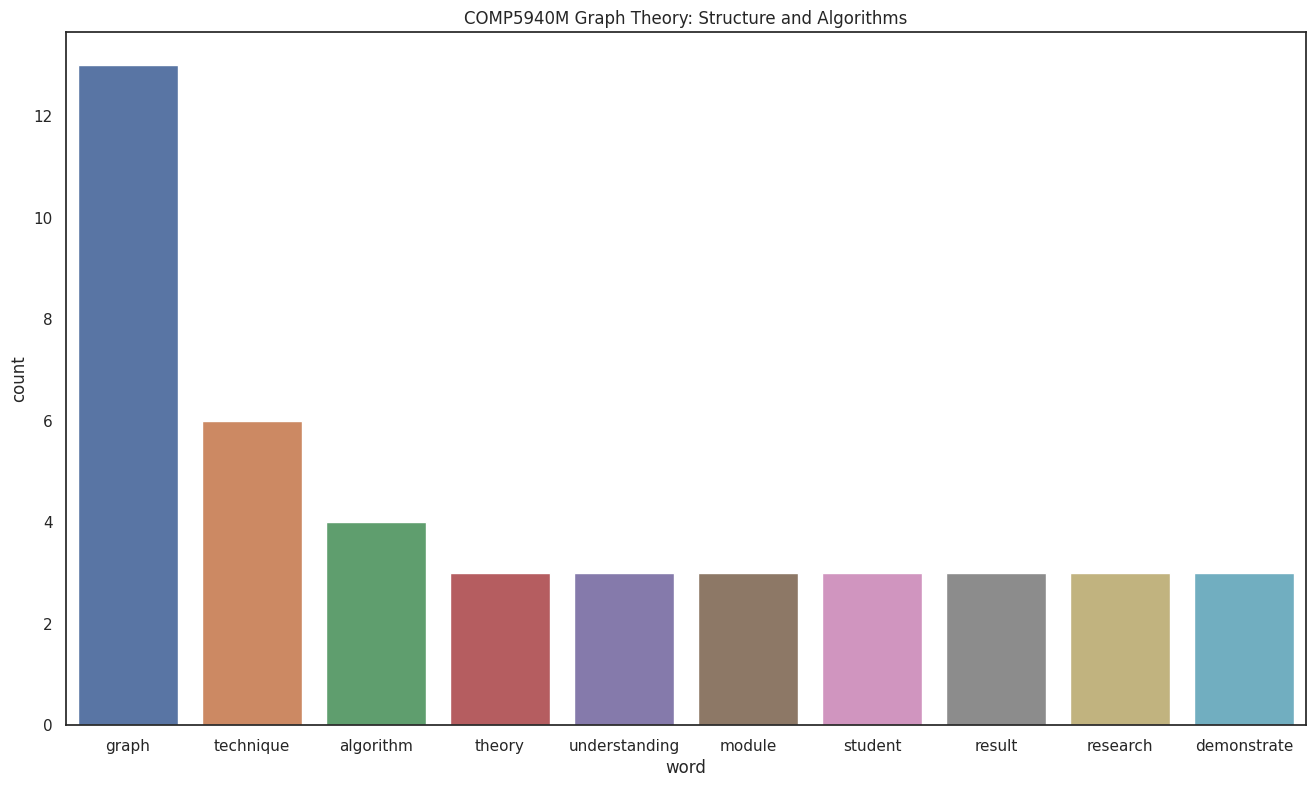
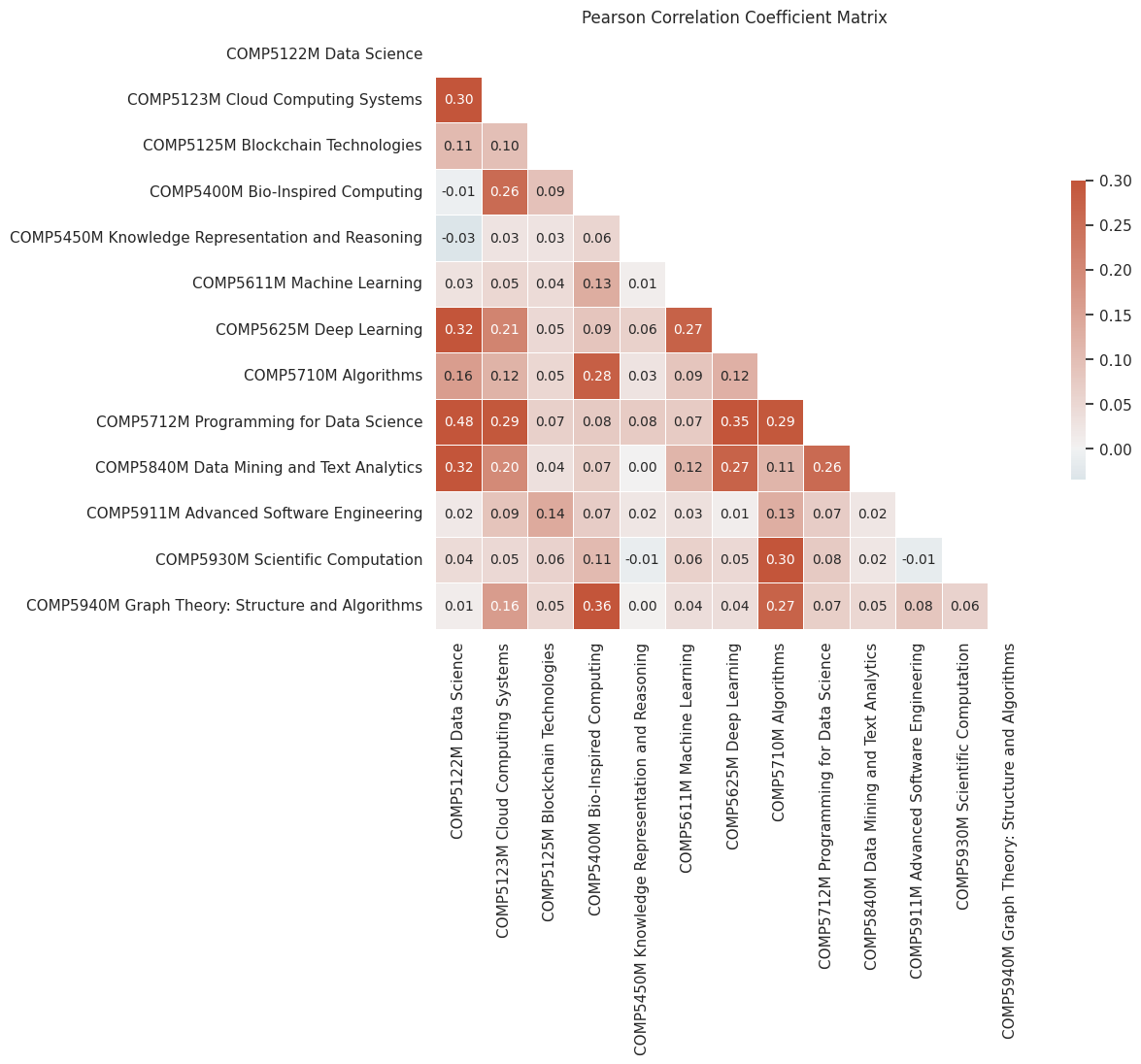
 

We can see each course most frequent words are highly possibly coming from its course title. The correlation heatmap shows that the COMP5712M Programming for Data Science are highly relative to the COMP5122M Data Science. Which is reasonable that we can just look at the course title and knows they are relative.

1. Text Analytic tools

ChatGPT

We asked him play the role of a university professor who helped correct our essays to make them more fluent, conform to academic writing style and check for grammatical errors.

When we finished the Contribution to knowledge chapter, we were not satisfied with what we wrote at the beginning. So, when we finished the code, we fed the code bit by bit to ChatGPT and let it list our project's contribution to knowledge. We then rewrote the chapter based on his results.

NewBing

NewBing is an AI assistant developed by Microsoft based on ChatGPT. Different from ChatGPT, it is connected to the Internet. So, we fed it all of our references, let him help us search it online for information on these referenced articles and URLs, and generate Leeds Harvard style references.

Reference

Korab, P. (2020) Advanced visualisations for text data analysis. Towards Data Science. Available at: https://towardsdatascience.com/advanced-visualisations-for-text-data-analysis-fc8add8796e2 (Accessed: 1 May 2023).

Li, S. (2019) A complete exploratory data analysis and visualization for text data. Towards Data Science. Available at: https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a (Accessed: 1 May 2023).

Education Today (2022) Record numbers of students choose computer science A-level in 2022. Available at: https://education-today.co.uk/record-numbers-of-students-choose-computer-science-a-level-in-2022/ (Accessed: 1 May 2023).

Li, Q. and Kim, J. 2021. A Deep Learning-Based Course Recommender System for Sustainable Development in Education. Applied Sciences. 11(19), p.8993.

Elahi, M., Starke, A., El Ioini, N., Lambrix, A.A. and Trattner, C. (2022) Developing and evaluating a university recommender system. Frontiers in Artificial Intelligence, 4, 796268. doi: 10.3389/frai.2021.796268