Project Proposal n#2

Semi-Structured Text Document Clustering Tool

*Project Report*

M. Aziz and R. Bou Zerdan

**Abstract**— Clustering is an integral part of grouping similar data based on numerous factors which include but are not limited to similarity distance, and number of clusters among others. Clustering is implemented in order to develop a relation between different objects in a set of data and group them into clusters based on common features for the purpose of understanding and manipulating data based on the requirements. This process is identified as an unsupervised machine learning approach to extract trends and relationships between data objects to satisfy their purposes. Data has become of high variety and hence the development of a semi structured text document clustering tool was developed to ensure accurate findings.

**Index Terms**—Data, Document Indexing, Indexing Methods, Natural Language Processing, Searching, Semantics, Clustering, classification, and association rules, Machine learning

Adapted from the IEEE Computer Society template

—————————— ◆ ——————————

**Contents**

[1 Introduction 1](#_Toc153795661)

[2 Project Background 2](#_Toc153795662)

[3 Design 2](#_Toc153795663)

[4 Experimental Evaluations 4](#_Toc153795664)

[6 Equations 5](#_Toc153795665)

[7 Conclusion 5](#_Toc153795666)

[**References** 6](#_Toc153795667)

# 1 Introduction

W

ith the increase of data types and data sources and with the generation of a variety of data every second, it has become important to develop methodologies and algorithms to facilitate dealing with this data and ensure that best practices are applied to extract useful insights and manipulate data for purposes of data analysis among others. New and large data depend on new relations and new dependencies to conform with the issuance of different types of data at any instance of time thus unsupervised algorithms have become at the forefront of this development making clustering and its underlying dependencies crucial part of business development, analysis trends, pattern recognition, and information retrieval.

Thus, the objective of this project is to develop a clustering tool which facilitates grouping of non-numerical data based on semantic relations between the different semi structured documents that are inputted to the algorithm. Since no preassigned restrictions and features are assigned, clustering uses common trends between the input data to group data for adequate groupings. Semi structured data requires additional steps to develop a clustering tool that conforms with the minimal structure and variety of values.

This project would allow the user through the user interface to input the type of clustering that he would like to refer to and afterwards the respective parameters in order to ensure that the user achieves the required purpose. A visual representation is available as well to make the results more relevant to any type of user and easier to extract relations based on the clustering performed.

# 2 Project Background

## 2.1 Context and Problem Analysis

This project was developed for the purpose of applying acquired knowledge from the Intelligent Data Processing Course in terms of concepts needed for data clustering to solve the issue of grouping textual data which is semi structured data with the help of unsupervised clustering algorithms. This has been achieved by building upon the semi structured search engine which encompasses converting data from semi structured text documents into more structured format using the Bert Model and Sentence Transformers.

Document based data are found all over the internet and acts as a main reference for individuals in the work field, but even though the abundance of data renders more accurate results, finding similar objects to the target or finding some common objectives and topics between a set of documents is quite challenging. Thus, the use of such a tool is of severe importance to deduce relations and meaning out of the data. Semi structured and unstructured data is found in numerous formats but for the purposes of this project, text files were used to implement the clustering and test its functionality.

## 2.2 Data Needed

For the purposes of this project, we used 4 input text files which were acquired from articles on the web, references to these articles are listed below. We named these as input\_1, input\_2, input\_3, and input\_4 [2], [3], [4], [5] which were preprocessed and combined before being input into the next step. In the next step, this data is used to perform the search model that was designed considering all the steps needed. Moreover, the documents, represented by vectors, are then transformed into embeddings as part of the corpus which have been used afterwards to implement the clustering algorithms which are kmeans partitional clustering and agglomerative hierarchical clustering as part of our unsupervised clustering algorithms.

## 2.3 Existing Solutions

Based on research and course content, there exists numerous clustering solutions including partitional, hierarchical divisive, hierarchical agglomerative, spectral, and others. These types of clustering are easily implemented on numerical data through scaling numerical content and hence apply vectorization. But when it comes to textual data further steps should be taken in order to be able first to vectorize the data, and second embed the data into the corpus which is going to be later used to apply each of the mentioned clustering algorithms.

# 3 Design

To design our clustering tool, we tackled 3 main steps which include conceptual modeling, software application design, and implementation. The mentioned parts will be discussed thoroughly below.

## 3.1 Conceptual Modeling

Our data is extracted from text files which have minimal structure based on titles and subtitles. Hence, to be able to use this data, as an input to the clustering tool built, it should be transformed into another format which should be either Json, xml, or csv format where the semi structured data will be available for the next phase. We decided to parse our data into csv format and to do so we read the data from the files and delimited the text into different sections based on a the end of every section and start of a new one. In our case, we chose “.” as our delimiter, since the chosen data is only divided based on unnumbered titles and subtitles so in case the data was structured in a different way, the delimitation would have been performed somehow differently in order to work for the variations of inputted text files. The texts were extracted using technologies such as pandas and python and were divided into rows containing every section of the text. Then this data was divided into article\_title, titles (containing article\_title and subtitles in the article), date, and content sections. Afterwards, this data was preprocessed, by eliminating null values, replacing empty areas in the title section with the corresponding title, replacing empty areas in the article\_title section with the article. Then the preprocessing is applied to all the input text files in the directory and finally combined into one csv file to facilitate the process of applying the next steps.

The aim is to convert the available textual data into vector form where they can be embedded in a corpus and hence can be clustered accordingly.

Then, the vectorized column is to be used to embed the individual vectors into the corpus to be used by both the hierarchical agglomerative clustering and partitional clustering. The “Sentence Transformer has been used for this purpose to ensure that textual data can be embedded into the corpus to be used as references for similarity measures.

Two clustering algorithms were implemented namely agglomerative hierarchical and kmeans partitional clustering to deal with our semi structured data.

## 3.2 Software Application Design

A simple user interface is developed to display the results of the search along with the results of the clustering algorithm. The user will be able to choose the needed clustering algorithm out of the two mentioned above by specifying its name and parameters including number of clusters and linkage when applicable. The user interface consists of one page hosted locally encompassing a search bar, an area to return the search results, dropdown menus to choose the clustering algorithm along with all related features, and scatter plots and dendrograms to display the clustering results.

A screenshot of a graph

Description automatically generated

Figure : Preview of Stream Lit Frontend

## 3.3 Implementation

The following tools were used in this project:

For Part 1 and Part 2 of the Project:

1. Python
2. Pandas
3. Numpy
4. NLTK Library
5. Elastic Search
6. Sentence Transformer
7. Scikit-Learn which encompasses both types of clustering as libraries (Agglomerative and Kmeans).
8. SciPy for agglomerative clustering features (Dendrogram)
9. Matplotlib for plotting and visualizing clusters.

For User Interface:

1. Python
2. Stream Lit

Python is used as the main coding language due to its wide use in the field of data since it encompasses a lot of useful libraries that facilitate extracting, preprocessing, analyzing, and searching the data. In addition to what was developed in project 1, vectorization of the data has been performed using the NLTK library which are then inputted into the embedding using the sentence transformer library where they are added to the corpus to be used by the Kmeans and Agglomerative functions. Afterwards, the resulting clustering is plotted into a graph using the Matplotlib methods to represent the different clusters and their respective centroids after the iterations have been performed whenever applicable. Then, a user interface is created including drop down menus to choose the clustering method along with the needed parameters to perform the required clustering. As an output, a figure is displayed to show the result of each of the algorithms in terms of dots for components of a cluster having same color and crosses that represent the centroids when they apply.

# 4 Experimental Evaluations

In our testing, when inputting keywords or phrases and based on our search engine implemented in project 1, documents would be returned based on relevancy. Given that we have applied semantic search initially, inputting phrases is allowed and hence returning results based on meaning. When it comes to clustering, we can choose between 2 types agglomerative hierarchical and partitional clustering. In the case of agglomerative clustering linkage can be chosen, to be either single, complete, ward, or average and hence the clustering would be affected. By default, we kept the affinity to be cosine similarity since our search engine and calculations are based on this approach. However, in Kmeans which represents partitional clustering, the user can input the number of clusters and by default we assigned a value for maximum number of iterations equal to 100, tolerance value equal to 1e-4, and the algorithm to full to make sure that our data is clustered correctly given that it is mostly sparse. It has been assigned these values as default because we believe that the user is less concerned with choosing the number of iterations and the remaining default inputted values since he or she is more concerned with the number clusters and clustering algorithm rather than these values which are usually in implementations kept as default values.

Agglomerative algorithm has a time complexity of O (N2) whereas partitional clustering has a time complexity of

O (n × k × i) where n represents the number of objects which are the different divided paragraphs in our case based on our input data and preprocessing steps, k represents the number of clusters which is chosen by the user in the user interface, and i represents the number of iterations where in our case we assigned it a default value. Hence it can be deduced the Kmeans partitional clustering algorithm is always faster that the hierarchical agglomerative algorithm since it has a lower time complexity equaling to O (N). How fast the processing is executed depends on the inputted factors n, k, I, but in general Kmeans is well suited for large datasets. Both algorithms are good approaches for dealing with small datasets but may render it less accurate than when having large data where there is a wide variety of data and similar documents, thus the clustering would be more visible.

# Equations

* **Term Frequency (TF)** refers to the number of times a particular term appears in a document.
* **Inverse Document Frequency (IDF)** measures the importance of a term in the entire corpus. Higher weights are assigned to terms that are rare in the corpus, whereas lower weights are assigned to terms that are more common. , where N is the total number of documents and n is the number of documents containing the term.
* **BM25** incorporates **document length normalization** to counteract the document size bias by dividing the term frequency by the document’s length and applying a normalization factor.

Where |D| represents the length of document D, and avgdl is the average document length in the corpus. Parameters k1 and b are tunable constants that control the impact of term frequency saturation and document length normalization, respectively.

* The below approach is followed by the partitional clustering built in algorithm:

A white background with black text

Description automatically generated

Figure : Partitional Clustering Algorithm Calculations

* When it comes to the different available linkage types in the hierarchical clustering based on similarity, there exists 4 common ones:
* Complete where it is taken as the min value.
* Single where it is taken as the max value.
* Average where it is taken as an average value.
* Ward where distance between items within a cluster is evaluated.

# Conclusion

In conclusion, the clustering approach used targeted the issue of semi structured data to ensure that known methodologies such as agglomerative and partitional clustering are implemented correctly on textual data rather than numerical data. Our approach involved the implementation of scikit-learn libraries where built functionalities were used. Other approaches may be used as well to acquire specific results. The proposed solution is thoroughly discussed throughout the report with applicable evidence and can be expanded as needed based on the targeted data and results desired by businesses. This project provided us with the opportunity to expand our knowledge in clustering algorithms by building upon acquired knowledge based on different available types. This project revolved around Python mainly, but other languages can be used as well to attain similar results. The process can be further expanded to target other types of clustering algorithms to accommodate the analysis requirements in case of specific input data and expected results. Additionally, outliers could be dealt with to obtain clearer results and well-defined clusters.

**References**

[1] “3   processing raw text,” ch03.rst2, https://www.nltk.org/book/ch03.html (accessed Nov. 12, 2023).

[2] Academic005Academic005                    5511 silver badge66 bronze badges and frankfrank                    10.6k33 gold badges2020 silver badges2929 bronze badges, “Intuitive explanation of Ward’s method,” Cross Validated, https://stats.stackexchange.com/questions/572409/intuitive-explanation-of-wards-method#:~:text=Ward%27s%20procedure%20is%20a%20variance,summed%20for%20all%20the%20objects. (accessed Dec. 18, 2023).

[3] “Keep your brain young with music,” Johns Hopkins Medicine, https://www.hopkinsmedicine.org/health/wellness-and-prevention/keep-your-brain-young-with-music (accessed Nov. 12, 2023).

[4] “Music and health: What you need to know,” National Center for Complementary and Integrative Health, https://www.nccih.nih.gov/health/music-and-health-what-you-need-to-know (accessed Nov. 12, 2023).

[5] “Music and health,” Harvard Health, https://www.health.harvard.edu/newsletter\_article/music-and-health (accessed Nov. 12, 2023).

[6] P. Everton Gomede, “Understanding the BM25 ranking algorithm,” Medium, https://medium.com/@evertongomede/understanding-the-bm25-ranking-algorithm-19f6d45c6ce (accessed Nov. 13, 2023).

[7] R. J. Stanborough, “Benefits of music on body, mind, relationships & more,” Healthline, https://www.healthline.com/health/benefits-of-music (accessed Nov. 12, 2023).