International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I3P121

Volume 72 Issue 3, 230-248, March 2024

Original Article

Keyword Extraction and Pattern Model Identification on Online Learning Contents for Classification to Enhance Microlearning Concepts and Obtain Personalized eLearning Contents

T. B. Lalitha[1], P. S. Sreeja[2]

[1]*Hindustan Institute of Technology and Science, Chennai, India.*
[2]*Vellore Institute of Technology, Chennai, India.*

[1]*Corresponding Author : lalitha.srm@gmail.com*

Received: 12 December 2023 Revised: 08 February 2024 Accepted: 23 February 2024 Published: 17 March 2024

Abstract *- The realm of keyword extraction and pattern model identification within the context of online learning materials, specifically focusing on its application to enhance the microlearning concept, delves into challenges in developing intricate recommendation systems. The rapid evolution of digital education platforms has underscored the need for effective content classification techniques to optimize the microlearning experience. Drawing upon an extensive corpus of online learning materials, this research employs advanced computational methods to extract pertinent keywords that encapsulate the essence of the content. By leveraging natural language processing and machine learning techniques, the study aims to unveil the intrinsic keywords that play a pivotal role in elucidating the core themes and concepts embedded within the learning materials. Furthermore, the research delves into identifying pattern models that underlie the structure and organization of the online learning content. These pattern models are systematically categorized and characterized through meticulous analysis and serve as a foundation for the subsequent classification process. The classification process itself constitutes a key facet of the study, as it involves the systematic categorization of online learning materials based on the extracted keywords and identified pattern models. The utilization of K-means, DBSCAN, and Agglomerative algorithms enables the discernment of meaningful clusters, patterns, and relationships within the corpus of online learning contents. This classification process augments the microlearning concept by providing learners with tailored and concise modules that align with their specific learning objectives. By enhancing the granularity and precision of content delivery, learners are empowered to engage more effectively with the material, thereby fostering a more impactful and efficient learning experience. This paper contributes to the scholarly discourse by presenting a comprehensive framework for keyword extraction, pattern model identification, and subsequent classification of online learning materials. The proposed approach not only enhances the microlearning paradigm but also offers insights into the broader landscape of digital education content recommendations. As the realm of online learning continues to evolve, the findings from this study hold significant implications for educators, instructional designers, and researchers alike, providing a robust foundation for the advancement of tailored and effective pedagogical strategies.*

Keyword *- Keyword extraction, Pattern model, eLearning, Clustering, K-means, DBSCAN, Agglomerative algorithm, classification.*

1. Introduction

In the ever-evolving view of education, the emergence of online learning has revolutionized the way knowledge is acquired and disseminated. With the digital world becoming an integral part of one's life, the education paradigm has expanded beyond traditional classroom settings. Online learning, with its inherent flexibility and accessibility, has empowered learners of diverse backgrounds and geographical locations to embark on educational journeys tailored to their individual needs. Both online learning and traditional learning have their own advantages and limitations. The effectiveness of each can vary based on their learners' preferences, circumstances, and the nature of the subject being taught [2]. Table 1 shows the comparison between Online learning and traditional learning. However, as the digital repository of educational content burgeons exponentially, the challenge of effectively organizing, classifying, and optimizing these resources for enhanced learning experiences becomes increasingly paramount. This educational methodology recognizes the limitations of human attention spans and leverages cognitive principles to optimize the acquisition and retention of knowledge. In the context of this research interest, microlearning holds relevance, as it aligns with the overarching goal of enhancing the learning experience within the area of online education.

Table 1. Online learning vs Traditional learning

Aspect	Online Learning	Traditional Learning
Delivery Method	Delivered via digital platforms and devices	Conducted in physical classrooms
Accessibility	Can be accessed from anywhere with internet	Limited to physical location
Flexibility	Offers flexible schedules and self-paced	Follows fixed schedules and timelines
Interaction	Primarily through digital communication	In-person interactions with peers/teachers
Resources	Relies on digital resources and materials	Uses physical textbooks and materials
Personalization	Can be tailored to individual leaming	Generalized curriculum for all students
Learning Pace	Can be faster or slower based on individual	Standardized pace for all students
Engagement	Requires self-discipline and motivation	Face-to-face engagement and accountability
Cost	Often more affordable due to reduced costs	May involve higher costs for tuition, etc.
Feedback	Often automated with instant assessments	Immediate, personalized feedback
Social Interaction	Limited physical interaction, more focus on virtual communication	Rich face-to-face social interactions

The Microlearning Paradigm and its Evolution

Microlearning, an innovative pedagogical approach, is characterized by its emphasis on delivering small, focused, and easily digestible bursts of learning content [3]. It stands in stark contrast to traditional methods of education, which often entail prolonged sessions and exhaustive content consumption. Microlearning capitalizes on the cognitive principle of "chunking," where information is presented in small, digestible units that align with the cognitive capacity of learners. This approach acknowledges the limitations of attention spans and seeks to optimize knowledge absorption.

The Key Characteristics of Microlearning are [4]:

- *Brevity*: Microlearning modules are designed to be brief, typically lasting for a few minutes. This brevity ensures learners can engage with the content without feeling overwhelmed or fatigued.
- *Focused Learning Objectives:* Each microlearning unit addresses a specific learning objective or concept. This focused approach facilitates clarity and precision in content delivery.
- *Multi-Modal Content*: Microlearning leverages various formats such as videos, infographics, quizzes, animations, and interactive simulations. This diversity caters to different learning styles and enhances engagement.
- *Just-in-Time Learning*: Microlearning is often used to provide on-demand learning, allowing learners to access relevant information precisely when needed.
- *Repetition and Reinforcement*: Microlearning modules can be revisited multiple times, reinforcing learning and aiding memory retention.
- *Mobile Compatibility*: Microlearning is well-suited for mobile devices, aligning with modern learners' preferences for learning on-the-go.

The Advantages of Microlearning are [2]:

- *Efficiency*: Microlearning optimizes the use of learners' time by delivering concise content that directly addresses specific learning goals.
- *Engagement*: Using multimedia and interactive elements keeps learners engaged and motivated, promoting active participation in the learning process.
- *Flexibility*: Microlearning's bite-sized format enables learners to fit learning into their busy schedules, enhancing accessibility and accommodating diverse lifestyles.
- *Retention*: The repetition inherent in microlearning aids memory retention, ensuring that key concepts are more likely to be retained over time.
- *Accessibility*: Learners with varying levels of prior knowledge can benefit from microlearning modules tailored to their specific needs.

In the context of online learning, microlearning acquires new dimensions. The digital environment, characterized by rapid information dissemination and abbreviated interactions, is inherently conducive to microlearning. Learners seeking quick answers, targeted information, or succinct explanations can benefit greatly from microlearning modules. Learners navigating online platforms are primed for brief yet impactful learning experiences. Online learners often have diverse backgrounds and varying levels of familiarity with the subject matter, so microlearning allows for a customized and adaptable learning experience.

The Significance of Keyword Extraction and Pattern Model Identification

In the contemporary era dominated by digital advancements, the sheer volume of textual data available at

the fingertips is nothing short of staggering. From scholarly articles to social media posts, the internet is awash with an overwhelming amount of information. Amidst this sea of data, the challenge lies not only in accessing the information but also in distilling meaningful insights from it. Keyword extraction has remained a vibrant research domain for numerous years, encompassing a diverse array of applications within the realms of Text Mining, Information Retrieval, and Natural Language Processing [5]. This versatile field has evolved to address diverse demands and specifications. This is where the significance of keyword extraction and pattern model identification comes to the fore. These techniques serve as guiding beacons in the vast expanse of textual content, offering a roadmap to navigate the intricate network of words and ideas.

Keyword Extraction [5]: Unlocking the Essence

- Effective communication and comprehension hinge on the skill of identifying and highlighting core concepts in a text.
- Keyword extraction is pivotal in achieving this precision by isolating essential terms, phrases, and ideas.
- Keywords act as guiding signposts, directing readers toward the central content of a lengthy document.
- The technique enhances information retrieval efficiency and deepens understanding of the subject matter.
- In academic research, keyword extraction holds immense significance for researchers, scholars, and students.
- Swift identification of key themes expedites literature review processes and aids in creating concise summaries.
- Abstracts produced through keyword extraction encapsulate the fundamental essence of a text.
- In the realm of search engines and information retrieval, accurate keyword extraction ensures users receive relevant search results aligned with their queries.

Pattern Model Identification [6]: Unveiling Hidden Relationships

- Textual data holds intricate relationships and patterns beyond what keywords reveal.
- Pattern model identification utilizes advanced algorithms to uncover recurring structures, connections, and trends within the text.
- This technique is akin to a linguistic archaeologist peeling back text layers to unveil concealed insights.
- Pattern model identification's significance spans diverse domains, showcasing its versatile applications.
- In the business realm, it aids in sentiment analysis, exposing emotional tones in customer reviews.
- Within social sciences, it can map the evolution of ideas and ideologies over historical texts.
- In healthcare, it assists in identifying correlations between symptoms and diseases in medical records.
- These applications harness the power of pattern model identification to extract previously hidden knowledge.

The Symbiotic Relationship [7]: Amplifying Insights

- Keyword extraction and pattern model identification possess inherent individual capabilities.
- Their true potency, however, emerges from a synergistic alliance.
- Keyword extraction serves as anchor points, guiding pattern model identification towards salient concepts.
- Pattern model identification, in turn, enhances keyword context, infusing them with deeper significance.
- The combined effect amplifies the capacity to comprehend textual content.
- These techniques empower us to derive insightful conclusions and make informed decisions.
- Moreover, they become catalysts for driving innovation and sparking new ideas.

Moreover, integrating the research findings on keyword extraction and pattern model identification can further optimize the design and delivery of microlearning content, ensuring its relevance and precision. Incorporating microlearning into online education can elevate the learning experience by catering to contemporary learners' preferences for bite-sized, engaging, and flexible learning materials. This research can potentially enhance the synergy between microlearning, algorithmic precision, and educational pedagogy, redefining the parameters of effective knowledge dissemination within the digital age.

This paper embarks on an eLearning online content exploration, delving into the aspects of keyword extraction, pattern model identification, and classification, with a specific focus on elevating the level of the microlearning content within the context of online education to enhance the efficacy and accuracy of recommendations of contents to learners.

The subsequent sections provide information: section 2. includes a background overview and literature review; Section 3. includes methodologies for the proposed work; Section 4. includes results and discussion; and finally, Section 5 as a conclusion. This facilitates a concise exposition of the issue at hand, the methodologies utilized, the outcomes, and their subsequent implications.

2. Background and Related Works

The fusion of technology and education has given rise to innovative methodologies aimed at optimizing the process of learning. The present literature survey delves into the interplay of keyword extraction, pattern model identification, and online learning content classification with the overarching goal of enhancing the microlearning concept for a profound recommendation system. This survey encompasses a wide spectrum of academic discourse, exploring the significance, techniques, and applications of these methodologies within the context of online education.

Keyword Extraction in eLearning web contents [8]:

- Keyword extraction has been recognized as a critical element in improving information retrieval and comprehension.
- Researchers have explored the integration of keyword extraction algorithms to enhance the organization and accessibility of educational content.
- The application of keyword extraction techniques in eLearning platforms has shown promise in facilitating personalized learning experiences.

Pattern Model Identification on Online Content:

- Pattern model identification has found traction in diverse fields, including natural language processing and text mining.
- In the realm of education, pattern model identification has been utilized to uncover hidden relationships within educational content, enabling insights into learning patterns and preferences.
- Studies have highlighted the potential of pattern model identification in identifying cognitive trends and mapping knowledge acquisition trajectories.

Integrating keyword extraction and pattern model identification contributes to a more personalized and efficient learning experience. Enhanced content organization resulting from these techniques promotes learner engagement and satisfaction. The forthcoming section comprehensively explores numerous pertinent studies that delve deeply into the realm of key extraction and pattern model identification. This meticulous examination will elucidate the intricate process of classifying eLearning web content, unraveling the nuances and intricacies inherent in these methodologies. Here are some works of literature related to keyword extraction and classification of eLearning web content.

Ao Xiog et al. [9] proposed a work that discusses the importance of keyword extraction in natural language processing and the challenges associated with it. It highlights the need for efficient and accurate keyword extraction algorithms to filter and disseminate information effectively in the network. It mentions that the test target of an experiment conducted in the document is a Chinese news library. The experiment involved extracting keywords from news articles obtained from a website. This mentions different methods and techniques used for keyword extraction, such as TF-IDF, TextRank, and semantic clustering. The experiments conducted compare the extraction effects of these algorithms based on precision, recall, and F1 value.

Achsan, H.T.Y., et al. [10] proposed a work to automatically extract stopwords from a large corpus of about seven million words in the Indonesian language. The researchers aim to reduce the computational costs of processing large and numerous documents by removing common words or stopwords. They use the Term Frequency - Inverse Document Frequency (TF-IDF) method to rank stopwords and develop a methodology that can be applied to different languages without prior linguistic knowledge. The research also aims to overcome the challenges of developing an automatic stopword extractor for the low-resource Indonesian language.

The research involved three stages: data gathering, preprocessing or data cleaning, and stopwords extraction. In the data gathering stage, the dataset was collected from the "Republika Daily" newspaper using the "Focused Web Crawling" method. Preprocessing steps included case folding, HTML tag removal, special character removal, tokenizing, dealing with missing data, data error handling, and stemming. The TF-IDF method was used for stopwords extraction, a combination of Term Frequency and Inverse Document Frequency methods. Chang, I. C., et al. [11]. The study's objective was to analyze and classify documents related to environmental education. The researchers used topic modeling and text-mining techniques to identify key topics and themes in the documents.

They aimed to provide insights into the field of environmental education and generate discussions about text mining in this context. The study also emphasized the importance of involving domain experts in environmental education research. It discusses various methods related to text mining and automatic document classification. It uses co-word analysis, machine learning processes, and applying a naïve Bayes algorithm for document classification. Additionally, the document highlights the use of text mining to analyze longitudinal trends in research and the analysis of titles and abstracts for identifying research trends. The methods also include word segmentation, feature word decision, and the involvement of domain experts in environmental education.

Arai, K. [12] work is to determine the importance of knowledge to be used to extract search keywords from documents. The proposed method aims to automatically extract keywords from paper media documents, such as drawings and forms, and construct a database for retrieval. The importance of the keywords is evaluated based on factors such as font size, position, and appearance frequency.

The goal is to improve the efficiency and accuracy of keyword extraction by minimizing the intervention of operators and considering the subjectivity of the knowledge. The document discusses various methods for extracting keywords from paper documents and drawings.

The proposed method utilizes the Analytic Hierarchy Process (AHP) to determine the importance of knowledge in selecting appropriate keywords for retrieval. The method involves converting paper documents into image files, classifying them into letters, forms, and drawings, and performing character recognition. Through experiments, the proposed system achieved a 98% success rate in extracting keywords with likelihood or certainty factors. Using AHP in the production system improved the success rate by 50% compared to the existing system without AHP.

Obaid, A. J. et al. [13] work is to develop or modify clustering algorithms in order to improve the search results for users accessing and retrieving information from web pages. The aim is to make the data structured and machine-readable, supporting easier data discovery, integration, navigation, and automation of tasks. By clustering web pages based on their contents, the goal is to improve the results of web search engines and other applications such as information retrieval systems. Various clustering algorithms are used, including density-based clustering methods (such as ExCC, MR-Stream, Denstream, and FlockStream), grid-based clustering methods, and hybrid methods that modify the DBSCAN algorithm. These methods are used for analyzing document clusters and web content. It mentions the Linked Data format, which contains 1,255 datasets with 16,174 links. It also discusses the characteristics and methods of clustering algorithms, including density-based clustering methods and graph construction methods.

Jayaram, K. et al. [14] work to explore text mining by machine learning techniques to facilitate the process of solving research problems and structuring thesis documents to help researchers access knowledge easily. The work also aims to extract the main thesis fields using decision trees and to extract code segments and their descriptions from research articles. Additionally, the work aims to predict characterization techniques and organization names using classification algorithms and evaluate the algorithms using LDA, NBS, and LIBLINEAR. Finally, the work discusses the importance of scientific research papers for experts and the challenges faced in viewing and retrieving scientific literature. Devi, S. A. et al. [15] explore various techniques and approaches for text classification and feature extraction. It focuses on methods such as word embedding, feature selection, and clustering algorithms to enhance the classification of text documents. The document also discusses using different models and algorithms, such as Naïve Bayes, SVM, and genetic algorithms, to handle uncertainty and optimize document prediction. Additionally, it highlights the importance of domain-specific knowledge and the use of commonsense information derived from the experiences of average people over the internet.

Najadat, H. M. et al. [16] work to present a new algorithm called Automatic Key phrases Extraction from Arabic (AKEA) that extracts key phrases from Arabic documents. The algorithm uses various attributes such as phrase frequency, term frequency, title threshold, TF-IDF, phrase position, and phrase distribution to identify key phrases. The document also discusses related works in the field of key phrase extraction and compares the performance of different attribute combinations. It involves a dataset of 100 Arabic documents from Arabic Wikipedia that was used to test the AKEA algorithm. Additionally, another dataset of 56 agricultural documents was downloaded from the Food and Agriculture Organization of the United Nations (FAO) website to evaluate the algorithm's performance further. The evaluation results showed that the AKEA system achieved 83% precision in identifying 2-word and 3-word key phrases. The document concludes by stating that the AKEA algorithm effectively extracts key phrases from Arabic documents and suggests future work to improve the algorithm further. Vashishta S. et al. [17] proposed works aimed at data mining that uncovers patterns and relationships already present in a target dataset. This requires assembling a large enough dataset that contains these patterns. Preprocessing is essential to analyze the dataset before clustering or data mining. The data is then cleaned to remove noise and missing data. The data is organized into clusters based on keywords extracted from biomedical text using the fuzzy C-means algorithm. The goal is to discover similar groups and structures in the data without using known structures in the data. One method is named entity recognition, which identifies biological entities such as protein and gene names in free text. Another method is the association of gene clusters obtained from microarray experiments with the corresponding literature. Additionally, text mining techniques are used to automatically extract protein interactions, associations of proteins to functional concepts, and even the extraction of kinetic parameters and subcellular locations of proteins. These methods rely on information extraction and text mining technology to analyze and extract relevant information from biomedical texts. Al-Maghasbeh, M. K. A., et al. [18] proposed an automatic domain extraction method to improve the retrieval of Arabic documents in information retrieval systems. The document discusses the problem of Arabic information retrieval and the importance of text classification in enhancing the performance and accuracy of retrieval systems.

The proposed approach involves document processing and user query processing phases, where text normalization, tokenization, and stop word removal are applied to the documents. Keywords are extracted using an ontology and patterns from other documents, and general topics or domains are determined by computing the vector space between the document keywords. Documents are classified based on cosine similarity, and user queries are processed to determine the query domain. The system matches the documents related to the user query and ranks them. The document also mentions related works in the area of Arabic text classification, such as the use of latent semantic analysis model, Naïve Bayesian method, and support vector machines.

Challenges and Future Directions

- While keyword extraction and pattern model identification offer immense potential, challenges such as noise reduction, algorithm selection, and evaluation metrics warrant further exploration.
- Microlearning is gaining popularity due to its flexibility, but algorithms tailored for such content are still under-explored
- Future research could delve into refining the algorithms, optimizing their integration into existing online learning platforms, and evaluating their impact on learner outcomes.

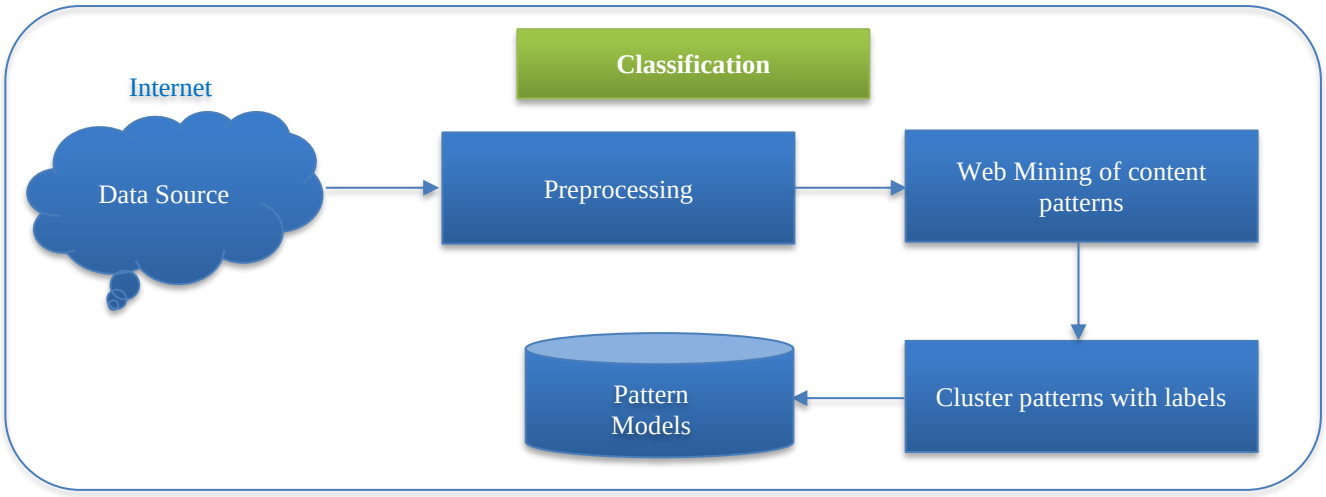

Fig. 1 Architecture of classification of web contents

The literature survey underscores the pivotal role of keyword extraction and pattern model identification in enriching online learning environments through enhanced content classification and delivery. Integrating these techniques holds the promise of transforming the microlearning concept into a dynamic and personalized educational experience, paving the way for a more effective and engaging digital pedagogy.

3. Proposed Work for the Classification of Web Contents

Web content classification is the process of categorizing web pages, documents, or other online content into predefined categories or topics based on their content, structure, or metadata [21]. This classification is often performed using various techniques, including machine learning algorithms [23], natural language processing, and text analysis. The goal of web content classification is to organize and make sense of the vast amount of information available on the internet, enabling efficient information retrieval and personalized content recommendations [22].

The proposed work aims at, depending on their profession, with varying qualifications and levels of experience, users to seek knowledge and expertise scattered across the vast expanse of the internet. However, the challenge lies in providing a personalized and efficient means of accessing this wealth of information to each user's unique needs. It is imperative to tailor the information to meet the specific needs of each user. In the context of eLearning, this personalization is both a convenience and a fundamental factor in effective knowledge acquisition. The aspiration is to transform this wealth of data into a personalized and enriching eLearning experience tailored to the user's query. To achieve this level of personalization, a pivotal step is to cluster the eLearning content users explore.

Clustering allows us to group similar content together based on quantitative content analysis rather than qualitative level of content, making it easier to offer relevant levels of materials such as low, medium, and high to users based on their queries and preferences. Thus, the approach focuses on and enhances the concept of microlearning that emphasizes the user's needs. This process hinges on extracting keywords from the eLearning content, as keywords serve as the anchor points for clustering. The high-level approach to classifying web content modules is given below in Figure 1.

3.1. Data Collection

The process of collecting the dataset involves a sophisticated technique known as web scraping. This technique employs web content mining to gather the necessary web content documents from search engine websites [24]. Web scraping essentially acts as an automated tool capable of extracting substantial volumes of precise data from web pages. It accomplishes this by deploying specialized bots that access websites directly, retrieving their contents in an organized manner. The web scraping process unfolds through a series of well-defined steps. Initially, input keywords are dispatched to a search engine, typically Google. Subsequently, URLs and page content are systematically scraped and collected.

The objective is to extract specific and relevant data from the web pages. This data extraction includes features of particular interest for further analysis and classification. Once the data has been harvested, it undergoes a refinement process to structure it into a format suitable for in-depth analysis. The structured dataset is then meticulously organized and stored within an Excel spreadsheet. This dataset contains various details, such as the quantity of data collected, the Google rank list, URLs of the web pages, keywords used, keyword density, and the total number of words within each web page.

The dataset is subsequently transferred and stored in Google Drive to ensure ease of access and convenience. Google Drive serves as a cloud-based storage service, offering a user-friendly environment for storing and retrieving the dataset. This cloud-based approach [25] ensures accessibility from various locations and devices, facilitating a seamless workflow for subsequent analysis and research activities.

	Page listing number on google	Website	keywords	word densities	word count
0	3	https://www.informit.com/articles/article.aspx...	Coordinationbetweenthreads-wait"	0.00	3353
1	4	https://www.programmerall.com/article/5083760149/	Coordinationbetweenthreads-wait"	1.02	1917
2	5	https://www.programmerall.com/article/5618303223/	Coordinationbetweenthreads-wait"	2.32	1606
3	6	https://www.tutorialspoint.com/importance-of-w...	Coordinationbetweenthreads-wait"	4.48	585
4	7	https://stackoverflow.com/questions/370265/java...	Coordinationbetweenthreads-wait"	2.83	9494
...	...	...	...	...	...
1560	4	http://coddingbuddy.com/article/55934002/read-...	writingtoafileusingvariousAPIsinJava	8.15	2332
1561	5	https://www.c-sharpcorner.com/article/file-cla...	writingtoafileusingvariousAPIsinJava	6.82	2131
1562	6	https://www.informit.com/articles/article.aspx...	writingtoafileusingvariousAPIsinJava	0.00	3248
1563	8	https://www.computerworld.com/article/2076076/...	writingtoafileusingvariousAPIsinJava	0.00	4069
1564	9	https://www.webucator.com/article/how-to-handl...	writingtoafileusingvariousAPIsinJava	7.98	915

1565 rows × 5 columns

Fig. 2 Exploratory sample dataset

	Website	Page listing number on google	word count	word densities	score
1496	http://pedemeiras.com.br/oxbmt0a/article.php?...	13	6398	10.24	7.319960
1520	http://goldenpharos.com/css/f7dd9i/article.php...	11	6055	6.49	5.030795
269	http://fabulous-imports.com/qmc65tw/lwxaf/arti...	9	4721	6.56	4.868754
1139	https://java.database-info.com/article/1209341...	7	6470	6.02	4.819300
150	https://www.vogella.com/tutorials/JavaIntroduc...	5	8582	5.52	4.683731
972	https://programmerall.com/article/8393390783/	6	9079	4.89	4.289178
1143	https://www.programmerall.com/article/4856627002/	2	4753	4.39	3.743445
433	https://www.vogella.com/tutorials/JavaIntroduc...	4	8582	4.07	3.722545
59	https://www.vogella.com/tutorials/JavaIntroduc...	3	8582	4.05	3.709288
1367	https://programmerall.com/article/88401115851/	10	4745	4.32	3.706415

Fig. 3 Popular websites in descending order of score values

In response to this data structure, a strategic decision was made to initiate web scraping procedures, targeting the website URLs. The objective was to acquire textual content from these web pages, thereby enabling preprocessing for subsequent analysis. The dataset comprises 1565 rows and 5 columns, encompassing information such as 'Page listing number on Google,' 'Website,' 'keywords,' 'word densities,' and 'word count.' Upon meticulous examination, it was discerned that the 'Website' and 'keywords' columns are of string data type. Figure 2 shows the exploratory sample dataset. The 'Website' column contains URLs and https links, while the 'keywords' column comprises words strung together without providing meaningful insights. Furthermore, an endeavor was undertaken to enhance data presentation and accessibility. This involved the creation of a new column labeled 'score,' which was calculated based on a weighted average of both 'word count' and 'word densities.' Here, the score value, taken based on the IMDB formula, which is given as

$$Score = (v / (v+m) * R) + (m / (m+v) * C) \quad (1)$$

Where v is the array of wordcount, R is the array of word densities. Subsequently, efforts were directed towards showcasing the list of websites in descending order, considering values for 'word count,' 'word densities,' and 'score.' The outcomes of these data manipulation and analysis endeavors are presented visually in Figure 3.

3.1.1. Extraction of Keywords

Following the web scraping process from individual URLs, proceeded to analyze the scraped data using various Python libraries, including spacy, rake, and yake. These libraries were instrumental in extracting keywords from the respective URLs.

The outcome of this analysis was then exported to a new CSV file named 'train_keyword_content.' This newly created file serves as the updated baseline training data, enriched with keyword information obtained through the application of these Python libraries. This refined dataset provides a more nuanced and detailed foundation for further analysis and model training.

3.1.2. Data Preprocessing

As a fundamental aspect of preprocessing the training data, focused on the 'spacy_keywords' [26] column feature, which essentially consists of a list of lists of keywords, forming the core of this analytical approach. Furthermore, each row has been meticulously examined to ascertain the number of keywords it contains. Subsequently, a meticulous cleansing process was executed. This process entailed the removal of punctuation, special characters, and common stop words, all contributing to an enhancement in the quality of the textual data to refine the contents.The culmination of these preprocessing efforts is represented in the input feature matrix, denoted as 'X.' To render this text data

comprehensible to machine learning algorithms, harnessed the power of the CountVectorizer method [27]. This method plays a pivotal role in converting this textual data into a format that machine learning algorithms can process. Moreover, transformed the data into a sparse matrix format, which is not only memory-efficient but also sets the stage for the forthcoming application of clustering algorithms.

This transformation is an essential precursor to efficient and effective data clustering, a crucial step in this analytical journey.

3.1.3. Applying the Clustering Algorithm

The overarching objective is to leverage diverse clustering algorithms to analyze extracted keywords. These algorithms, encompassing K-Means, DBSCAN, Agglomerative, and potentially others, are poised to unveil patterns, relationships, and thematic clusters within the realm of eLearning materials [28]. A critical aspect of this process involves meticulously evaluating algorithmic performance to identify the one that most aptly aligns with the specific requirements.

This rigorous assessment is pivotal in selecting the optimal algorithm for the subsequent classification and organization of eLearning content. The advantages of proficiently clustering and classifying eLearning materials are indeed multifaceted. This process not only simplifies users' access to pertinent content but also elevates the quality of the overall learning experience. Learners can navigate the vast landscape of online resources more effectively, pinpointing precisely the materials required to progress in their respective fields. Furthermore, this approach holds the promise of conserving valuable time and effort for both educators and learners, ultimately rendering the learning journey more efficient and enjoyable. The grouping of relevant data from this dataset falls under the unsupervised learning domain, making clustering algorithms the preferred choice for this task. Clustering algorithms come in various types, each with its unique approach:

- Centroid-Based clustering [29]: In centroid-based clustering, data points are grouped around the centroid of a cluster. K-Means is a prominent example of a centroid-based algorithm. It aims to minimize the distance between data points and the cluster center.
- Density-Based clustering [30]: Density-based clustering identifies clusters based on the density of data points in a particular area. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a well-known algorithm in this category. It identifies dense regions as clusters.
- Hierarchical-Based clustering [31]: Hierarchical clustering builds a hierarchy of clusters, forming a tree-like structure (dendrogram). It can be either divisive (top-down) or agglomerative (bottom-up). Agglomerative hierarchical clustering is commonly used, combining data points into successively larger clusters.
- Distribution-Based clustering [32]: Distribution-based clustering assumes that the data in each cluster follows a specific statistical distribution. Gaussian Mixture Models (GMM) is a popular distribution-based clustering algorithm. It models clusters as Gaussian distributions.

Each of these clustering methods offers a distinct approach to uncovering patterns and relationships within data. The choice of the most suitable clustering algorithm depends on the nature of the data and the specific objectives of the analysis. The application of these clustering techniques to this dataset holds the potential to unveil valuable insights and organize eLearning content effectively. The application of clustering algorithms [33] follows a systematic process, as outlined below:

1. Determine the Optimal Number of Clusters ('k' Values): The first step is determining the optimal number of clusters for the dataset. This involves assessing different 'k' values, representing the number of clusters into which the data will be divided. Various methods, such as the elbow method or silhouette analysis, can be employed to identify the most suitable 'k' value.
2. Visualize the Clusters (Centroid of Clusters) for Different 'k' Values: Once the optimal 'k' value is determined, the next step is to visualize the clusters. This visualization often involves plotting the centroids of the clusters. It provides a visual representation of how data points are distributed within each cluster for different 'k' values.
3. Evaluate the Performance of the Clustering Algorithm in Terms of 'Silhouette Score': The Silhouette score is a common metric used to evaluate the performance of clustering algorithms. It measures the quality of clusters by assessing how similar data points are to their own cluster (cohesion) compared to other clusters (separation). If the Silhouette score is high, then it indicates better-defined clusters.
4. Comparison of Clustering Algorithm Performances: After applying the above steps to each clustering algorithm, a thorough comparison of their performances is conducted. This comparison involves assessing the quality of the clusters and how well they align with the specific requirements of the analysis.
5. Selection of the Most Suitable Clustering Algorithm: Based on the performance evaluations, the clustering algorithm that best meets the analysis requirements is selected. The chosen algorithm will be the one that yields the most coherent and relevant clusters within the eLearning materials dataset.

The following sections will provide a detailed description of the application of clustering algorithms and their outcomes, ultimately leading to selecting the most appropriate algorithm for the specific use case.

The current work focuses on three distinct clustering algorithms, each bringing a unique approach to the analysis:

K-Means Algorithm [34]

- Type: Centroid-based algorithm.
- Operation: It forms clusters around centroids, with data points assigned to the cluster whose centroid is closest to them.
- Characteristic: Effective for well-defined and compact clusters, minimizing the intra-cluster variance.

The K-Means algorithm is arguably one of the most renowned clustering techniques. Its primary objective is to assign data examples to clusters in a manner that minimizes the variance within each cluster. This process is implemented through the "K-Means" class in machine learning libraries.

The central configuration to fine-tune when working with K-Means is the "n_clusters" hyperparameter. This parameter should be set to the estimated number of clusters that best fit the data. This tuning is critical in achieving meaningful cluster assignments. Upon running the K-Means algorithm, the model is fitted to the training dataset, and each example in the dataset is assigned to a cluster based on its characteristics. A scatter plot is often created to provide a visual representation of the clusters, where data points are colored according to their assigned clusters. This visualization aids in understanding the grouping of data within the dataset and can reveal patterns and structures that exist within the data.

The pseudo-code gives outlines of the K-Means Clustering algorithm:
Input:

- Data points D
- Number of clusters k

Pseudo Algorithm:

1. Initialize k means with random values.
2. For a given number of iterations (this is a tuning parameter):
- Traverse each data point in the dataset.
- For each data point:
- Find the mean (centroid) closest to the data point by calculating the Euclidean distance between the data point and each cluster means.
- Assign the data point to the cluster with the nearest mean.
- Update the mean of that cluster by shifting it to the average of all the data points in that cluster.

Output:

- Data points with their cluster memberships

This algorithm iteratively refines the cluster assignments and updates the cluster means until convergence. The result is data points assigned to clusters, and each cluster has its own mean. K-Means is an effective method for partitioning data into clusters based on similarity, making it a valuable tool for various applications, including image segmentation, customer segmentation, and more.

DBSCAN Algorithm [35]:

- Type: Density-based algorithm.
- Operation: Identifies clusters based on the density of data points. It can find clusters of arbitrary shapes and is robust to outliers.
- Characteristic: Particularly useful for datasets with varying density and irregularly shaped clusters.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that identifies high-density regions in the feature space and expands those regions to form clusters. Here's an explanation of how DBSCAN works and its main hyperparameters:

*Implementation: DBSCAN is implemented through the DBSCAN class in machine learning libraries.

*Hyperparameters to Tune:

- eps (epsilon): Epsilon is the maximum distance between two data points for one to be considered as in the neighborhood of the other. It defines the radius around each data point within which other data points are considered neighbors. This parameter is essential for specifying how close data points should be to one another to be considered part of the same cluster. It's a crucial parameter and should be chosen carefully according to the dataset and distance function. The default value for epsilon is 0.5.
- min_samples: Min_samples is the number of data points (or total weight) in a neighborhood for a point to be considered a core point. Core points are the central data points around which clusters are formed. The min_samples parameter also includes the data point itself. The default value for min_samples is taken as 5.

*Running the Algorithm:

- To apply DBSCAN, fit the model to the training dataset, which involves determining the clusters based on the specified epsilon and min_samples values.
- After fitting the model, it can predict the cluster assignment for each data point in the dataset.
- A scatter plot is often created to visualize the clustering results, with data points colored according to their assigned cluster.

DBSCAN is particularly useful for finding complex-shaped clusters and handling noisy data. Choosing appropriate values for epsilon and min_samples is important to achieve meaningful and reliable cluster assignments. Finding the best values of epsilon (eps) [1] and min_samples for DBSCAN can be a crucial step in ensuring the effectiveness of the clustering. The Silhouette score is a valuable metric for this purpose. Here's an explanation of how it works:

- The Silhouette score measures the clustering quality based on the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample.
- The Silhouette score is calculated for each sample as (b - a) / max(a, b). In this formula:

- ‘a’ represents the average distance from the sample to the other data points within the same cluster.
- ‘b’ represents the average distance from the sample to the data points in the nearest cluster that the sample is not part of.
- The Silhouette score is defined only when the number of labels (clusters) falls within the range of 2 <= n_labels <= n_samples - 1.
- The best Silhouette score is 1, indicating well-separated clusters with samples much closer to their own cluster’s data points than other clusters. The worst score is -1, suggesting that samples have been assigned to the wrong clusters. Values near 0 indicate overlapping clusters.

When optimizing the parameters epsilon and min_samples for DBSCAN, you can perform the following steps:

1. Define a range of possible values for epsilon and min_samples to test.
2. For each combination of epsilon and min_samples, apply DBSCAN to the data.
3. Calculate the Silhouette score for the resulting clusters.
4. Choose the combination of epsilon and min_samples that gives the highest Silhouette score. This indicates the best parameter values for the dataset, resulting in meaningful and well-separated clusters.

This process helps to fine-tune DBSCAN for the specific data and cluster structure. The pseudo-code outlines a systematic approach to finding the best values of epsilon (eps) and min_samples for DBSCAN using the Silhouette score as an evaluation metric. Here’s a breakdown of the steps:

1. Initialize a loop for each epsilon value (eps), starting from 0.1.
2. Within this loop, initialize another loop for each value of min_samples, starting from 2.
3. Calculate the Silhouette score using DBSCAN clustering with the current values of eps and min_samples.
4. Increment the value of min_samples by 1 and continue calculating the Silhouette score until it reaches 5.
5. After the inner loop, increment the value of eps by 0.01 and repeat the entire process until it reaches 0.9.
6. End the second (inner) loop.
7. End the first (outer) loop.
8. Display the Silhouette score values in descending order, along with the corresponding values of eps and min_samples.
9. Identify the best or optimum values of eps and min_samples for which the Silhouette score is maximized.

This systematic exploration of different combinations of epsilon and min_samples allows us to find the parameter values that lead to the best clustering results. By sorting the Silhouette scores in descending order can easily identify the combination that yields the highest score, which indicates the most suitable parameters for the dataset.

The steps and principles of the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm are as follows:

- Find Neighbor Points: Begin by finding all the neighbor points within a distance of epsilon (eps) for each data point. Identify core points and data points with more than MinPts neighbors within this radius.
- Create Clusters: For each core point that has not already been assigned to a cluster, create a new cluster.
- Density-Connected Points: Recursively find all the density-connected points to each core point and assign them to the same cluster as the core point. Two points, ‘a’ and ‘b,’ are considered density-connected if there exists another point ‘c’, with a sufficient number of neighbors within the eps distance, and both ‘a’ and ‘b’ are within the eps distance from ‘c.’ This process forms a chain of connected points. In other words, if ‘b’ is a neighbor of ‘c,’ ‘c’ is a neighbor of ‘d,’ ‘d’ is a neighbor of ‘e,’ and ‘e’ is a neighbor of ‘a,’ it implies that ‘b’ is also a neighbor of ‘a.’
- Identify Noise: Iterate through the remaining unvisited points in the dataset. Those points lacking association with any cluster are deemed as noise.

DBSCAN effectively finds clusters of arbitrary shapes and handles noisy data, making it a valuable tool in density-based clustering scenarios. The algorithm’s ability to discover clusters based on data point density rather than assuming a fixed number of clusters sets it apart from other clustering algorithms. The pseudo-code effectively outlines the steps of the DBSCAN clustering algorithm. Here’s a summary of the main parts of the code:

- Input: The algorithm takes the dataset’ D,’ the desired number of clusters’ k,’ the distance threshold’ eps,’ and the minimum number of points in a cluster ‘MinPts’ as input.
- Main Function: The main function, named ‘DBSCAN,’ operates on the dataset with the specified parameters.
- Initialization: To begin by initializing a cluster index ‘C’ to 1. This variable will be used to assign cluster labels to data points.
- Iterating Through Data Points: For each unvisited data point’ p’ in the dataset, the algorithm performs the following steps:
  1. Mark the point ‘p’ as visited to ensure it’s not processed again.
  2. Find the neighboring points of ‘p’ within a distance of ‘eps’ and store them in the ‘Neighbors’ set ‘N’.
  3. Check if the number of neighbors, denoted by ‘|N|,’ is greater than or equal to ‘MinPts.’
  4. If there are enough neighbors, expand the set of neighbors’ N’ by finding the neighbors of neighbors (‘N’ union ‘N”).
  5. For each point ‘p” in the expanded set of neighbors’ N,’ if ‘p” has not already been assigned to any cluster, add it to cluster ‘C.’
- Output: The result of the DBSCAN algorithm is data points with cluster memberships, indicating which cluster each point belongs to.

The pseudo-code provides a clear and concise representation of the DBSCAN clustering process, making it easier to understand how the algorithm identifies clusters in a dataset based on density.

Agglomerative Algorithm [36]:

- Type: Hierarchical-based algorithm.
- Operation: Performs connectivity-based clustering, where data points close to each other on the similarity distance measure are clustered together. It creates a hierarchy of clusters.
- Characteristic: Hierarchical structure allows for the exploration of different levels of granularity in cluster organization.

Agglomerative clustering is a hierarchical clustering method that involves merging examples until the desired number of clusters is achieved. This method is available in scikit-learn via the Agglomerative Clustering class. The primary configuration parameter to adjust is the "n_clusters," which estimates the number of clusters in the data.

The general process of agglomerative clustering involves starting with individual data points as separate clusters and then iteratively merging clusters based on a specified linkage criterion, such as single linkage, complete linkage, or average linkage until the desired number of clusters is obtained. After fitting the agglomerative clustering model on the training dataset, it can be used to predict a cluster label for each example in the dataset. A scatter plot can then be created to visualize the clusters, with data points colored according to their assigned cluster. Agglomerative clustering is a versatile method that can be useful for exploring the hierarchical structure of data and identifying clusters at different levels of granularity. It provides a valuable tool for cluster analysis and visualization.

The pseudo-code effectively outlines the key steps of the Agglomerative Clustering algorithm. Here's a summary of the main parts of the code:

1. Input: The algorithm takes the dataset 'D' and the desired number of clusters' k' as input.
2. Initialization: The algorithm begins by initializing 'n' clusters, with each cluster containing one object. These clusters are numbered from 1 to 'n.'
3. Distance Computation: It then calculates the between-cluster distances' D(r, s)' for all pairs of clusters' r' and 's.' This involves computing the distance between the objects within each pair of clusters. If the objects are represented as quantitative vectors, Euclidean distance is commonly used.
4. Cluster Merging: The algorithm identifies the most similar pair of clusters' r' and 's' by finding the minimum distance 'D(r, s)' among all pairwise distances. It then merges these clusters into a new cluster, 't.' For each existing cluster 'k' that is not 'r' or 's,' it computes the between-cluster distance 'D(t, k).' After these distances are computed, the algorithm updates the distance matrix 'D' by removing the rows and columns corresponding to the old clusters' r' and 's' and adding new rows and columns corresponding to the newly formed cluster 't.'
5. Iterative Process: Steps 3 and 4 are repeated a total of 'n - 1' times, gradually merging clusters until there is only one cluster left.
6. Output: The result of the Agglomerative Clustering algorithm is data points with cluster memberships, indicating which cluster each point belongs to after the merging process.

The pseudo-code provides a clear and structured representation of the Agglomerative Clustering process, making it easier to understand how clusters are formed through a hierarchical merging approach. The selection of these algorithms showcases a comprehensive approach, considering both centroid-based, density-based, and hierarchical-based strategies. This diversity is beneficial for capturing different types of structures and patterns within the dataset. Each algorithm brings its strengths and is suited to different scenarios, contributing to a robust and nuanced analysis of the eLearning content.

4. Results and Discussion

4.1. K-MEANS Algorithm

4.1.1. Elbow method [37]

The Elbow Method is a graphical approach to finding the optimal number of clusters (K) in a K-Means clustering. It works by assessing the Within-Cluster Sum of Squares (WCSS), the sum of the squared distances between data points within a cluster and the cluster's centroid. Here's a pseudo-code to implement the Elbow Method for K-Means Clustering:

1. Run the K-Means algorithm for a range of values of K, for example, from K=1 to K=10.
2. For each value of K, calculate the sum of squared distances for each data point to its closest cluster centroid. This sum is referred to as the Sum of Squared Errors (SSE). SSE represents the error, as in an ideal scenario, every data point should be exactly on the centroid of its cluster, which is not practically achievable.
3. Plot the SSE values against the number of clusters, K. The curve generated typically resembles an "elbow."
4. The value of K, where the SSE starts to exhibit a significant decrease and levels off, is considered the "elbow point." This elbow point is often taken as an indicator of the appropriate number of clusters for the dataset.

The Elbow Method is a useful tool for making an informed decision about the number of clusters to use in K-Means clustering. It helps strike a balance between minimizing intra-cluster distance and avoiding an excessive number of clusters, which may lead to overfitting.

Figure 4 illustrates the Elbow Method, displaying the variation of the Within-Cluster Sum of Square (WCSS) distance between data points and their respective cluster

centroids as a function of the number of clusters. In this specific case, the optimal number of clusters is observed at k = 3. This point marks the "elbow" in the plot, indicating that beyond this value of k, the reduction in WCSS becomes less significant. Therefore, k = 3 is identified as the most appropriate number of clusters for the dataset, striking a balance between partitioning the data effectively and preventing an excessive number of clusters.

4.1.2. Inertia of cosine [20]

The term "inertia" in the context of K-Means clustering refers to the sum of squared distances of samples to their closest cluster center. To find the optimal number of clusters (k), an iterative approach is typically used:

1. Iterate through a range of k values, often from 1 to a specified upper limit, such as 10.
2. For each value of k, apply the K-Means algorithm to partition the data into k clusters.
3. Calculate the inertia for each k, the sum of squared distances from data points to their closest cluster centers.
4. By comparing the inertia values for different values of k, can identify the point at which the reduction in inertia starts to level off. This is typically the optimal value of k, representing the appropriate number of clusters for the dataset.

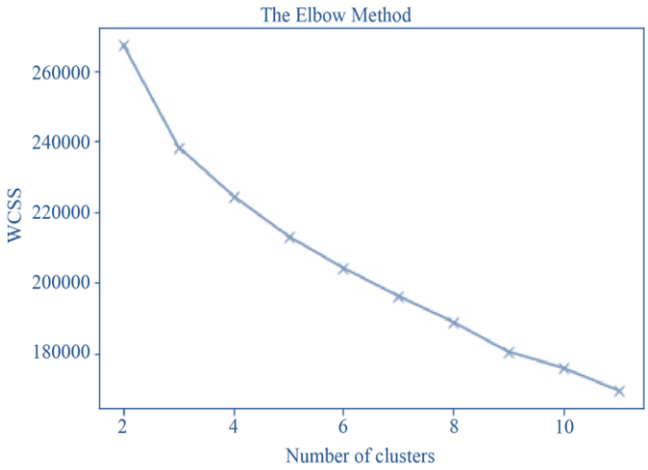

Fig. 4 Elbow method - WCSS vs. number of clusters

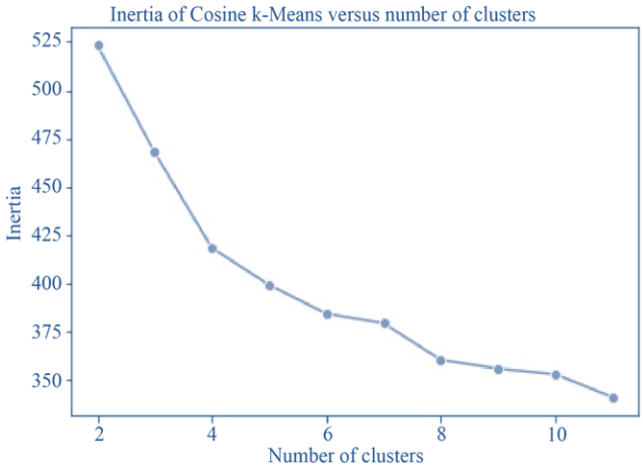

Fig. 5 Inertia of cosine vs. number of clusters

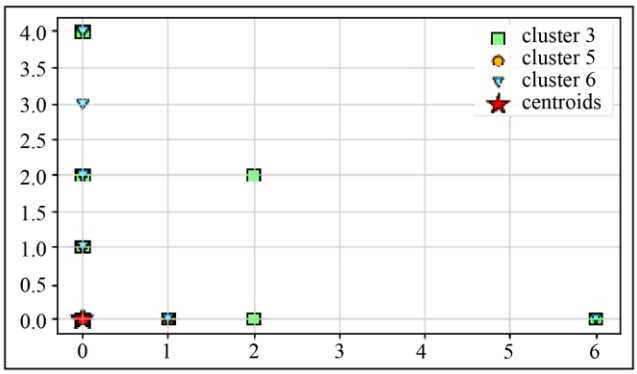

Fig. 6 Plot of clusters with centroid

This process allows us to systematically explore different cluster numbers and select the one that balances cluster quality and complexity, aiming for a meaningful data partition.

Figure 5 presents a visualization of the inertia of the K-Means clustering algorithm with respect to the number of clusters. In this context, the inertia is computed using the cosine distance metric. The plot shows how the inertia changes as vary the number of clusters increases. It's common to observe a pattern where the inertia decreases as the number of clusters increases. However, the rate of decrease may slow down at a certain point. The optimal number of clusters is often found at the "elbow" of the plot, where the reduction in inertia starts to level off. Figure 4 provides a graphical representation of the trade-off between the number of clusters and the clustering quality. It helps identify an appropriate number of clusters for the specific dataset and analysis.

4.1.3. Visualizing Clusters

Figure 6 displays a plot of 3 clusters, each along with their respective centroids. This visualization illustrates the data points grouped into three distinct clusters, with the centroids representing the center points of these clusters. This depiction offers insight into how the data is partitioned and how each cluster is characterized by its centroid.

Figure 7, on the other hand, presents a separate plot that specifically focuses on the centroids for the 3 clusters. It provides a clear view of the centroids' locations in relation to the data points, showcasing the central tendencies of each cluster. These visualizations are essential for understanding the results of a K-Means clustering analysis, as they offer a visual representation of how the data has been grouped and the central positions of these groups, represented by the centroids.

Figures 8, 9, 10, 11 depict clustered data points for varying numbers of clusters like 3, 5, 6, and 8 clusters. These visual representations provide an overview of how each scenario's data is segmented into different clusters. Examining the data in this manner helps assess the impact of different cluster numbers on the dataset's structure. It can guide the selection of an appropriate number of clusters for the specific analysis and objectives.

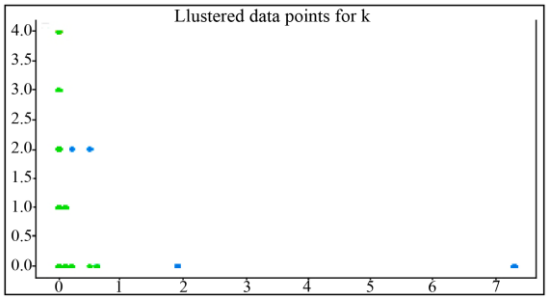

Fig. 7 Plot of centroid for 3 clusters

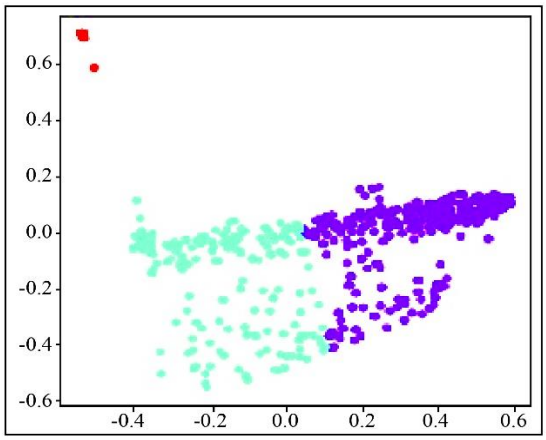

Fig. 8 Plot of data points for 3 Clusters

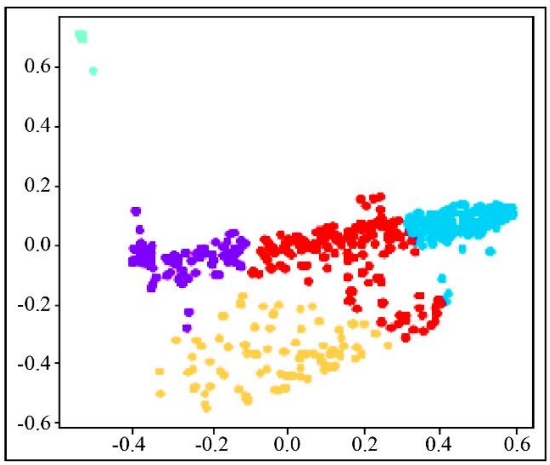

Fig. 9 Plot of data points for 5 Clusters

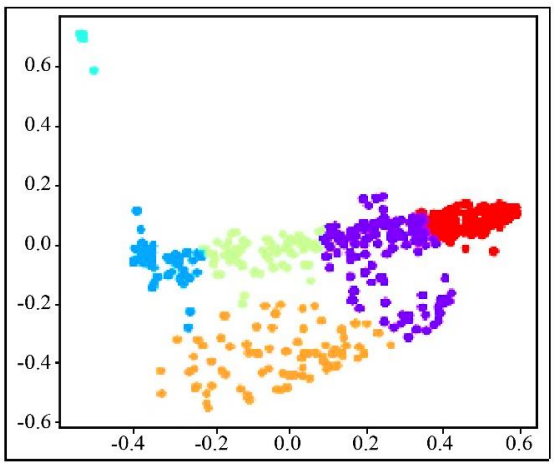

Fig. 10 Plot of data points for 6 Clusters

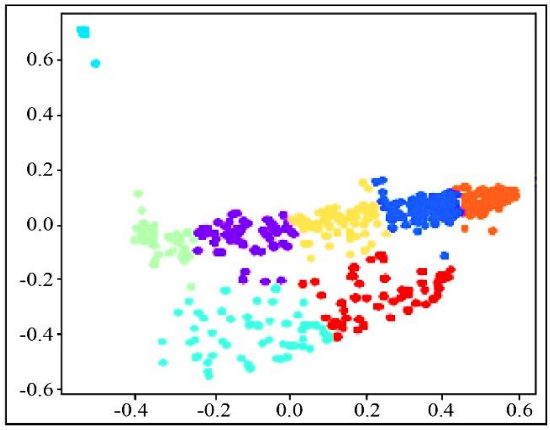

Fig. 11 Plot of data points for 8 Clusters

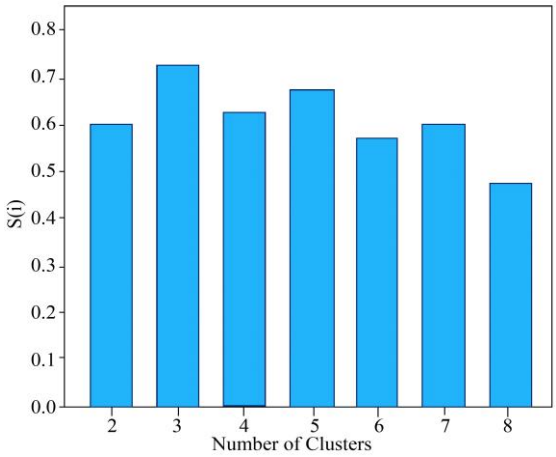

Fig. 12 Silhouette score vs the number of K clusters

4.1.4. Performance Evaluation

The Silhouette Score [19], also known as the Silhouette Coefficient, is an evaluation metric that provides a measure of how well data points are clustered. It yields values within the range of -1 to 1, with different interpretations:

- A score near 1 suggests that data points are tightly grouped within their respective clusters and well-separated from other clusters. This indicates a highly appropriate clustering.
- A score near -1 implies that data points may have been assigned to the wrong clusters, as they are closer to other clusters' centroids. This signifies a poor or inappropriate clustering.
- A score near 0 indicates that clusters may be overlapping, making it challenging to distinguish between them.

Here, the Silhouette Score is used to evaluate the performance of K-Means clustering, and the metric values are provided to assess the quality of the clustering. This score helps in gauging how well the clustering algorithm has organized the data into meaningful groups, with higher scores indicating better clustering quality.

The above Figure 12 presents a bar graph illustrating the variation of the Silhouette Score with the number of clusters for the K-Means clustering algorithm. This type of graph is particularly useful for visually identifying the optimal number of clusters based on the Silhouette Score.

The bar graph typically shows the Silhouette Score for different values of k, and the highest score or the "elbow" point in the graph is considered indicative of the optimal number of clusters. This is the value of k that results in the best balance between compactness within clusters and separation between clusters. Analyzing the Silhouette Score graph can assist in making an informed decision about the number of clusters that optimally represent the underlying structure of the data.

The performance metrics for K-Means clustering have been assessed, and the results are as follows:

- Silhouette Coefficient for K-Means: 0.7234283
- Silhouette Score for Cosine Similarity: 0.6699167

These scores provide a quantitative measure of the quality of the clustering achieved by the K-Means algorithm. A higher Silhouette Score generally indicates better-defined and well-separated clusters. In this case, both the Silhouette Coefficient for K-Means and the Silhouette Score for Cosine Similarity are relatively high, suggesting that the clustering has been effective in organizing the data points into distinct and cohesive clusters. These metrics are valuable for assessing the success of the clustering and ensuring that it aligns with the specific goals of the analysis.

DBSCAN Clustering Algorithm

Creating Clusters using the best hyperparameters, Fig. 13 illustrates the cluster centers created using the optimal hyperparameters, where eps is set to 0.10, and min_samples is set to 2. These parameters have been chosen to maximize the clustering effectiveness based on the analysis, resulting in well-defined clusters within the dataset. The cluster centers represent the central points within each cluster, and their positions provide insights into the structure and distribution of data points within these clusters. This visualization is a valuable representation of the results achieved with the DBSCAN algorithm and the selected parameter values. Visualizing clustered data points for different values of epsilon (eps) is essential for understanding how the choice of this parameter affects the clustering results.

Figure 14 shows the clustered data points for eps values of 0.10, 0.15, 0.30, and 0.60, with a consistent min_samples value of 2. These visualizations reveal how the number of clusters can vary based on the epsilon value, as indicated by the colored data points in the plots. The visual representation of different clusterings allows us to assess the impact of varying the epsilon parameter on the number and shape of clusters. It's an important step in understanding the sensitivity of the DBSCAN algorithm to this critical hyperparameter and fine-tuning it to suit the specific dataset.

The Silhouette score is a valuable metric for evaluating the performance of clustering algorithms like DBSCAN. It provides an indication of how well-defined and separate the clusters are within the data. Based on the evaluation, the Silhouette scores for DBSCAN clustering and cosine-based DBSCAN are as follows:

- Silhouette Score for DBSCAN: 0.6108615
- Silhouette Score for Cosine DBSCAN: 0.5775843

A heightened Silhouette score signifies more clearly defined and distinctly separated clusters. The results suggest that the DBSCAN algorithm, with its parameters tuned for the dataset, has produced clusters with a Silhouette score of 0.6108615, indicating a reasonable degree of cluster separation. The Silhouette score for cosine-based DBSCAN is 0.5775843, which is also useful information for assessing clustering performance when considering the cosine similarity metric.

Agglomerative Clustering Algorithm

Visualizing dendrograms is a common method for determining the optimal number of clusters in hierarchical clustering, such as Agglomerative Clustering. The process involves plotting the data in a way that resembles a tree structure, with horizontal lines representing clusters at different levels of merging. To find the optimal number of clusters, imagine all the horizontal lines being entirely horizontal and then calculate the maximum distance between any two horizontal lines. The horizontal line corresponding to this maximum distance is drawn, indicating the optimal number of clusters.

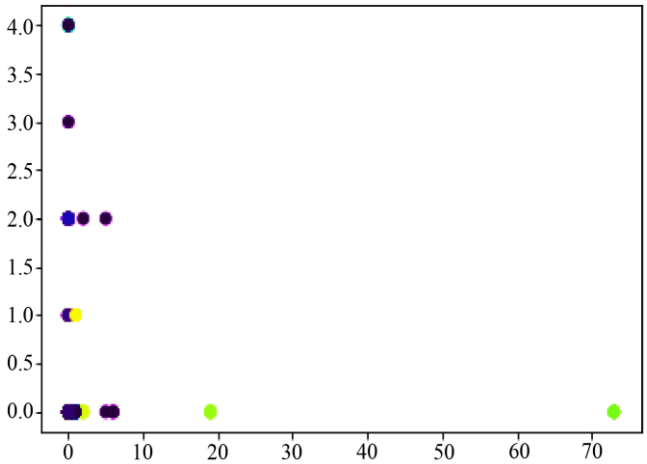

Fig. 13 Plot of cluster centers with eps = 0.10 and min_sample = 2

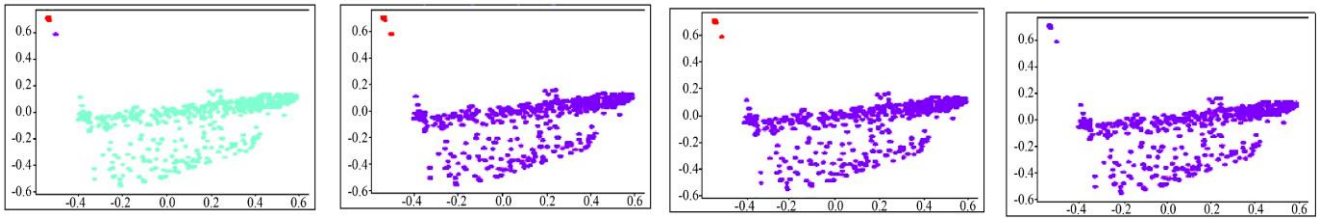

Fig. 14 Plot of data points for eps=0.10, 0.15, 0.30, 0.60

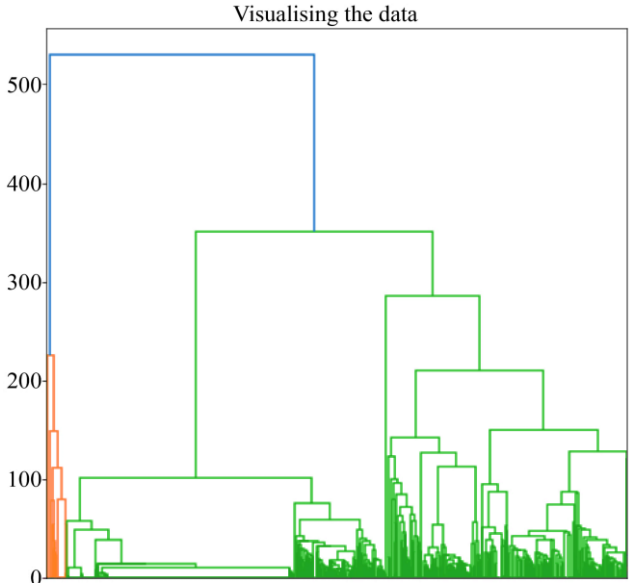

Fig. 15. Dendograms

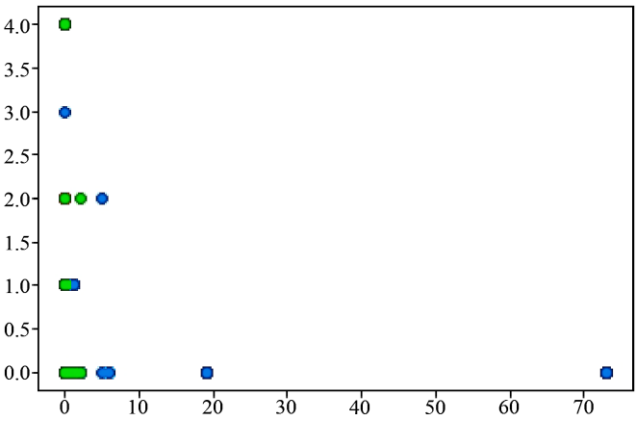

Fig. 16 Plot of cluster centers for 3 clusters

Figure 15 shows that the optimal number of clusters for the given data should be 3. This is based on the dendrogram's structure and the maximum distance between horizontal lines.

Visualizing dendrograms is a helpful technique for understanding the hierarchical clustering process and determining the appropriate number of clusters for the data. Visualizing the clusters for different values of 'k' is a crucial step in understanding how the data is segmented when using clustering algorithms and determining the optimal number of clusters.

In this case, Fig. 16 shows the cluster centers for 'k' equal to 3, indicating that it has been chosen to partition the data into three clusters. This visualization provides insights into how the data points are grouped within these clusters, helping to assess the quality and appropriateness of the chosen 'k' value for clustering. It's a valuable tool for exploring the results of the clustering analysis.

Figure 17 provides a comprehensive view of the data's clustering patterns for 'k' values ranging from 2 to 7. Observing these plots allows us to make informed decisions about the appropriate number of clusters for the specific dataset.

The variations in the number of clusters and the distribution of colored data points in the plots help to understand how different values of 'k' impact data grouping. This visual exploration is critical for refining the clustering analysis and choosing a value of 'k' that best captures the underlying structure of the data. It's a valuable step in ensuring the meaningful segmentation of the dataset into distinct clusters.

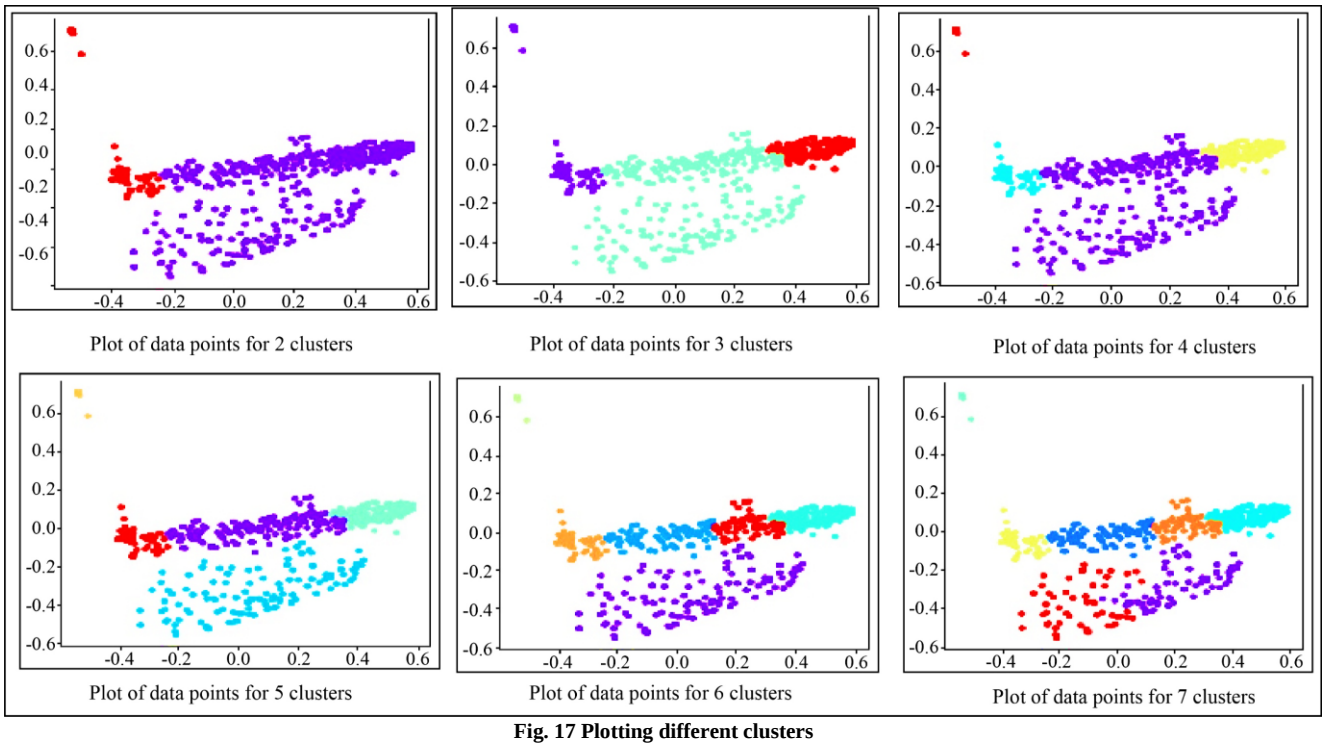

Fig. 17 Plotting different clusters

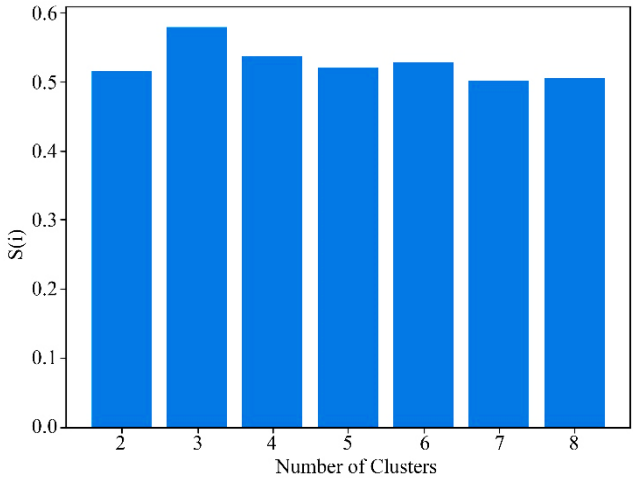

Fig. 18 Plot of Silhouette score vs # of K clusters

Figure 18, depicting the bar graph of the Silhouette score variation with the number of clusters, provides a quantitative measure of the quality and appropriateness of different 'k' values for clustering analysis. The Silhouette score is a valuable metric for assessing the cohesion and separation of clusters, with higher scores indicating better-defined clusters. Visualizing how the Silhouette score changes as the number of clusters ('k') varies can identify the 'k' value that results in the most meaningful and well-separated clusters. This graph is a helpful tool for objectively determining the optimal number of clusters for the dataset, as it allows us to make data-driven decisions about data segmentation.

Using Silhouette scores to determine the optimal number of clusters is a sound approach. In this analysis, it's concluded that the optimal number of clusters for the given data and clustering technique is 3. The Silhouette score helps ensure that the clustering results in well-defined and separated clusters, leading to more meaningful and interpretable results. The performance evaluation, as indicated by the Silhouette scores for Agglomerative Clustering and Cosine Agglomerative Clustering, provides a quantitative measure of the quality of the clusters. These scores can be used to compare the performance of different clustering techniques and guide the selection of the most appropriate approach for the specific data.

- Silhouette Score for Agglomerative Clustering: 0.5793522
- Silhouette Score for Cosine Agglomerative Clustering: 0.5059546

A Silhouette score of 0.5793522 for Agglomerative Clustering and 0.5059546 for Cosine Agglomerative Clustering indicates the quality of the clusters in terms of cohesion and separation. Higher scores are generally preferred, as they indicate well-separated and distinct clusters. This objective performance evaluation is valuable for making data-driven decisions in clustering analysis, ensuring that the chosen clustering technique and the number of clusters are suitable for the dataset.

Table 2. Comparative analysis of performance evaluation

Algorithm	Silhouette Coefficient	Silhouette score for cosine value	Processing Time
K-Means clustering	0.7234283	0.6699167	(9.375) + (10.025) = 29.4 s
DBSCAN	0.6108615	0.5775843	(321) + (5.667) = 326.67 s
Agglomerative	0.5793522	0.5059546	(32.26) + (3.596) = 35.83 s

Creation of Pattern Model Repository

Table 2 provides specific values for different algorithms used in this work, enabling a clear comparison and assessment of their performance in the context of this proposed research.

This table is a valuable reference for understanding how each algorithm performs and assists in selecting the most suitable approach for this work. It's an essential tool for making informed decisions in the research. "K-Means" has been selected as the clustering algorithm based on the Silhouette score for the given dataset, which is a well-informed decision. The Silhouette score helps ensure that the chosen algorithm and number of clusters result in meaningful and well-separated clusters. In this case, it has been determined that the optimal number of clusters is 3. This choice will guide the creation of the Pattern Model Repository (PMR), and it reflects a data-driven approach to clustering that is likely to yield valuable insights and patterns within the dataset. The total processing time in this context is the sum of two components: computational time and time for visualization.

$$\text{Total processing time} = \text{Computational time} + \text{time for visualization}$$

- Computational Time: This includes the time taken for the algorithm to process the data, perform clustering, and any other analytical tasks. It involves all the calculations and operations carried out on the dataset to obtain the clustering results.
- Time for Visualization: This component accounts for the time spent creating visualizations, such as plots, graphs, and charts, to represent and interpret the clustering results. Visualization is essential for gaining insights from the data and presenting these insights more understandably.

Adding these two components can calculate the total processing time, which provides a comprehensive view of the time required to complete the data analysis and clustering tasks, including computation and visualization. This thorough evaluation of clustering techniques and using objective metrics like the Silhouette score demonstrate a commitment to making informed decisions in the analysis.

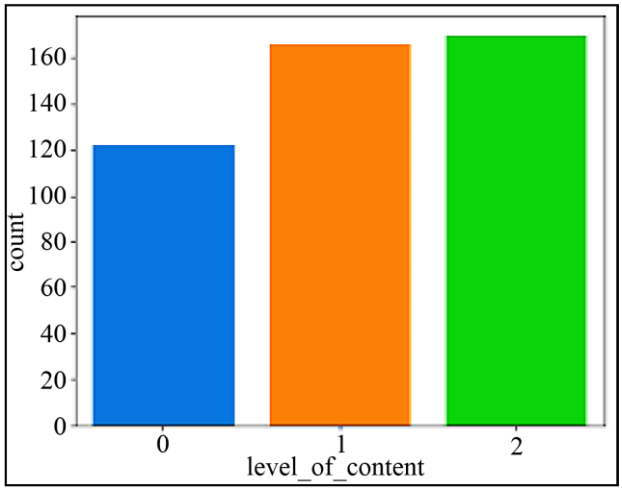

Fig. 19 No. of contents in each level

This approach is crucial for obtaining meaningful results in the Pattern Model Repository. Figure 19 illustrates the graph showing the number of contents present in each level of impact content. This visual representation helps convey the distribution and categorization of content based on their impact levels. It provides a clear overview of how much content falls into different impact categories, making it easier to understand the distribution of impactful content in the research or analysis. This approach aids users in efficiently navigating the vast sea of information on the internet, allowing them to extract the desired level of content precisely. Through a quantitative lens, the level of impact becomes a guiding factor, streamlining the content selection process. This enhances the effectiveness of microlearning and significantly reduces the time spent searching for relevant concepts. In the digital age, where information is abundant, this method empowers users to derive maximum value from their learning experiences by focusing on content that quantifiably aligns with their needs and objectives.

This kind of structured data is valuable for further analysis and can be utilized in the Personalization module as indicated. Overall, this approach demonstrates a systematic and organized way of handling clustering results and preparing data for subsequent stages in the research or system. Here, the approach of defining class labels for each cluster (‘Low’, ‘Medium’, and ‘High’) and mapping them to the ‘Level of content’ is a structured way to categorize the clusters based on impact levels. Creating a new column feature in the DataFrame and removing unnecessary columns streamlines the data to focus on the relevant information. The resulting Pattern Model Repository (PMR), containing features such as ‘spacy_keywords,’ ‘kmeans_cluster_labels,’ and ‘level_of_content,’ is then exported to a CSV file named ‘pattern_model_repo.csv.’ This CSV file serves as a repository of patterns identified by the clustering algorithm and their associated impact levels. Analyzing the impact of content from a quantitative perspective is instrumental in advancing the microlearning concept.

5. Conclusion

The pursuit of personalized eLearning experiences in the era of information abundance and digital learning stands as a noble and essential endeavor. At the heart of this mission lies the intricate process of clustering eLearning content. By harnessing the power of keyword extraction and employing suitable clustering algorithms, it embarks on a journey toward efficient knowledge acquisition and a more dynamic, personalized approach to education in the digital age.

This work specifically delves into the realm of eLearning web content, focusing on the classification of scraped and meticulously preprocessed materials. Through this process, the concept of microlearning can be significantly enhanced. Microlearning, with its emphasis on bite-sized, easily digestible content, becomes even more effective when learners can access precisely the materials they need. Among the various clustering algorithms, K-Means emerges as a standout performer in this work. Its ability to cluster content with high accuracy and consistency makes it the ideal choice for achieving the project’s objectives. Leveraging the classified “level of impact” of eLearning content, poised to provide users with remarkably accurate personalized recommendations. This tailored approach not only optimizes knowledge absorption but also ensures that learners engage with content that aligns seamlessly with their unique goals and preferences. Quantitatively assessing the impact of content plays a pivotal role in advancing the concept of microlearning. This approach empowers users to extract the precise content they need from the vast sea of information on the internet efficiently. Quantifying the significance of content streamlines the process of accessing relevant microlearning resources, ultimately saving valuable time that would otherwise be spent sifting through an overwhelming amount of data.

This quantitative perspective not only enhances the effectiveness of microlearning but also ensures that learners can derive maximum value from their educational endeavors. It’s a strategic solution that optimizes the learning experience in this digital age, aligning the wealth of information available with the specific needs of each user. In a digital age characterized by vast information resources and evolving learning paradigms, the convergence of personalized eLearning and advanced clustering techniques promises a future where education is more accessible, efficient, and engaging than ever before. This marks a pivotal step toward the realization of a dynamic and responsive educational landscape. In conclusion, the aspiration for personalized eLearning experiences in a time characterized by surplus data and digital education is a commendable undertaking. Clustering eLearning content through keyword extraction and applying suitable clustering algorithms is crucial in achieving this goal. It is a journey towards efficient knowledge acquisition and a more dynamic and tailored approach to education in the digital age.

References

[1] Kinsuk Giri, and Tuhin Kr. Biswas, "Determining Optimal Epsilon (eps) on DBSCAN Using Empty Circles," *International Conference on Artificial Intelligence and Sustainable*, pp. 265-275, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ragad M. Tawafak et al., "E-learning vs. Traditional Learning for Learners Satisfaction," *International Journal of Advanced Science and Technology*, vol. 29, no. 3, pp. 388-397, 2020. [Google Scholar] [Publisher Link]

[3] Mary Jo Dolasinski, and Joel Reynolds, "Microlearning: A New Learning Model," *Journal of Hospitality & Tourism Research*, vol. 44, no. 3, pp. 551-561, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Lucas Kohnke, *Microlearning as a Teaching and Learning Approach, In: Using Technology to Design ESL/EFL Microlearning Activities*, Springer, Singapore, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Nazanin Firoozeh et al., "Keyword Extraction: Issues and Methods," *Natural Language Engineering*, vol. 26, no. 3, pp. 259-291, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Minhua Huang, and Robert M. Haralick, "Identifying Patterns in Texts," *2009 IEEE International Conference on Semantic Computing*, Berkeley, CA, USA, pp. 59-64, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[7] Rachael L. Buchanan, Paul H. Ricks, and Terrell A. Young, "Narrative Blossoming: The Symbiotic Relationships of Newbery Novels and Their Graphic Adaptations," *Children's Literature in Education*, vol. 55, pp. 60-74, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Marwa Hendez, and Hadhemi Achour, "Keywords Extraction for Automatic Indexing of E-Learning Resources," *2014 World Symposium on Computer Applications & Research (WSCAR)*, Sousse, Tunisia, pp. 1-5, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ao Xiong et al., "News Keyword Extraction Algorithm Based on Semantic Clustering and Word Graph Model," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 886-893, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Harry Tursulistyono Yani Achsan et al., "Automatic Extraction of Indonesian Stopwords," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 166-171, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] I-Cheng Chang et al., "Applying Text Mining, Clustering Analysis, and Latent Dirichlet Allocation Techniques for Topic Classification of Environmental Education Journals," *Sustainability*, vol. 13, no. 19, pp. 1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kohei Arai, "Extraction of Keywords for Retrieval from Paper Documents and Drawings Based on the Method of Determining the Importance of Knowledge by the Analytic Hierarchy Process: AHP," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, pp. 48-55, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ahmed J. Obaid, Tanusree Chatterjee, and Abhishek Bhattacharya, "Semantic Web and Web Page Clustering Algorithms: A Landscape View," *EAI Endorsed Transactions on Energy Web*, vol. 8, no. 33, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Kavitha Jayaram, G. Prakash, and V. Jayaram, "Automatic Extraction of Rarely Explored Materials and Methods Sections from Research Journals Using Machine Learning Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 447-456, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] S. Anjali Devi, and S. Siva Kumar, "A Hybrid Document Features Extraction with Clustering Based Classification Framework on Large Document Sets," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 364-374, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hassan M. Najada et al., "Automatic Keyphrase Extractor from Arabic Documents," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 192-199, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] Sumit Vashishta, and Yogendra Kumar Jain, "Efficient Retrieval of Text for Biomedical Domain Using Data Mining Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 4, pp. 77-80, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mohammad Khaled A. Al-Maghasbeh, and Mohd Pouzi Bin Hamzah, "A Method of Automatic Domain Extraction of Text to Facilitate Retrieval of Arabic Documents," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 8, pp. 227-230, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Ketan Rajshekhar Shahapure, and Charles Nicholas, "Cluster Quality Analysis Using Silhouette Score," *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, pp. 747-748, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Xiaohui Cui, and Thomas E. Potok, "Document Clustering Analysis Based on Hybrid PSO+ K-means Algorithm," *Journal of Computer Sciences*, pp. 27-33, 2005. [Google Scholar] [Publisher Link]

[21] Marcos Forte, Wanderley Lopes de Souza, and Antonio Francisco do Prado, "A Content Classification and Filtering Server for the Internet," *Proceedings of the 2006 ACM symposium on Applied Computing*, pp. 1166-1171, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[22] Goran Matošević, Jasminka Dobša, and Dunja Mladenić, "Using Machine Learning for Web Page Classification in Search Engine Optimization," *Future Internet*, vol. 13, no. 1, pp. 1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Safae Lassri, El Habib Benlahmar, and Abderrahim Tragha, "Machine Learning for Web Page Classification: A Survey," *International Journal of Information Science and Technology*, vol. 3, no. 5, pp. 38-50, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Guandong Xu, Yanchun Zhang, and Lin Li, *Web Content Mining*, *Web Mining and Social Networking: Techniques and Applications*, Springer US, vol. 6, pp. 71-87, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[25] Spirin Oleg et al., "The Blended Methodology of Learning Computer Networks: Cloud-Based Approach," *Proceedings of the 15th International Conference, ICTERI 2019*, Kherson, Ukraine, vol. 2, pp. 68-80, 2019. [Google Scholar] [Publisher Link]

[26] Yuli Vasiliev, *Natural Language Processing with Python and SpaCy: A Practical Introduction*, No Starch Press, pp. 1-216, 2020. [Google Scholar] [Publisher Link]

[27] Radhika Goyal, "Evaluation of Rule-Based, CountVectorizer, and Word2Vec Machine Learning Models for Tweet Analysis to Improve Disaster Relief," *2021 IEEE Global Humanitarian Technology Conference (GHTC)*, Seattle, WA, USA, pp. 16-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Dongkuan Xu, and Yingjie Tian, "A Comprehensive Survey of Clustering Algorithms," *Annals of Data Science*, vol. 2, pp. 165-193, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[29] Santosh Kumar Uppada, "Centroid Based Clustering Algorithms-A Clarion Study," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, pp. 7309-7313, 2014. [Google Scholar] [Publisher Link]

[30] Ricardo J.G.B. Campello et al., "Density-Based Clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 2, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[31] Frank Nielsen, *Hierarchical Clustering*, Introduction to HPC with MPI for Data Science, Springer, Cham, pp. 195-211, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[32] Xiaowei Xu et al., "A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases," *Proceedings 14th International Conference on Data Engineering*, Orlando, FL, USA, pp. 324-331, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[33] Hailei Zou, "Clustering Algorithm and its Application in Data Mining," *Wireless Personal Communications*, vol. 110, pp. 21-30, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek, "The Global K-Means Clustering Algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451-461, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[35] Dingsheng Deng, "DBSCAN Clustering Algorithm Based on Density," *2020 7th International forum on Electrical Engineering and Automation (IFEEA)*, Hefei, China, pp. 949-953, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Marcel R. Ackermann et al., "Analysis of Agglomerative Clustering," *Algorithmically*, vol. 69, pp. 184-215, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[37] Mengyao Cui, "Introduction to the k-Means Clustering Algorithm Based on the Elbow Method," *Accounting, Auditing and Finance*, vol. 1, no. 1, pp. 5-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]