Document Clustering of Irish Government Circulars using Machine Learning Techniques

MSc Research Project
Master of Science in Artificial Intelligence (MSc AI Top-up)

Gabriel Amariei
Student ID: 13130510

School of Computing
National College of Ireland

Supervisor: Faithful Onwuegbuche

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gabriel Amariei
Student ID:	13130510
Programme:	Master of Science in Artificial Intelligence (MSc AI Top-up)
Year:	2024
Module:	MSc Research Project
Supervisor:	Faithful Onwuegbuche
Submission Due Date:	12/08/2024
Project Title:	Document Clustering of Irish Government Circulars using Machine Learning Techniques
Word Count:	XXX
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gabriel Amariei
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Document Clustering of Irish Government Circulars using Machine Learning Techniques

Gabriel Amariei
13130510

Abstract

Text clustering has emerged as a powerful tool to address the issue of exponential growth in the volume of textual documents that are generated by organizations worldwide. It has enabled the organization of large document corpora into distinct groups based on content similarity thus enhancing the efficiency and effectiveness of information retrieval within vast collections.

In this study, we have clustered the Irish Government circulars with the goal of enhancing the accessibility and retrieval of information from these documents. Given the lack of prior categorization and the unknown number of clusters within this dataset, unsupervised learning methods were employed to discover the inherent structure of the documents. More specifically, we utilized three advanced document representation techniques: the TF-IDF, Word2Vec, and BERT, together with three clustering algorithms: K-Means, Eigenspace-based Fuzzy C-Means (EFCM), and a version of the Long Short-Term Memory (LSTM) neural network.

Our findings indicate that among the document representation techniques tested, **Word2Vec** outperformed both TF-IDF and BERT in effectively capturing the nuances of the documents within the Irish Government circulars. When it came to clustering, **K-Means** proved to be the most effective and consistent algorithm for this task. The exploratory use of the LSTM-based method showed promise, but further refinement and testing would be needed to fully assess its capabilities in this specific application.

1 Introduction

The growing volume of text documents across various knowledge fields has made the organization and extraction of relevant and effective information increasingly difficult. As a result, document clustering has emerged as a valuable asset for grouping text documents into relevant and appropriate categories. It has been successfully used to tackle this issue in various fields like market research, social media analysis, medical and biomedical, law, and technology Gabralla and Chiroma (2020). This has allowed us to enhance information retrieval and identify key topics within the collection of clustered documents.

However, document clustering presents its own set of challenges, and specialized tools and techniques have been developed to address these issues. The process typically involves three steps: document preprocessing including text cleaning, the numerical representation of documents such as word to vector representation, and clustering where various algorithms can be used in conjunction with optimization techniques like dimensionality reduction.

During the document preprocessing step, the raw text documents are refined to make them suitable for analysis. This includes tasks such as text cleaning, tokenization, stemming, and lemmatization, which simplify the text data and remove unnecessary characters or features to enable further processing Cozzolino and Ferraro (2022).

The document vector representation step involves converting documents into numerical forms. This enables the clustering algorithms to measure similarity distances between the documents. Techniques used for this include traditional models which count the words in each document, distributional models which predict semantic similarity based on context, and word embedding models which learn dense, low-dimensional representations of words from large corpora. Deep learning models further enhance this process by capturing both word semantics and contextual information, resulting in more detailed text representations Asudani et al. (2023); Ravi and Kulkarni (2023); Subakti et al. (2022).

In the final step, specific algorithms are used to try to separate the documents forming the corpus into clusters. Clustering algorithms are generally classified into hierarchical and partitional. They can also be classified into hard and soft clustering models where hard clustering assigns each document to only one cluster, while soft clustering calculates a membership degree for each document, allowing it to belong to multiple clusters Cozzolino and Ferraro (2022).

Deep learning has also been applied successfully to clustering, with popular techniques including convolutional neural networks, deep belief networks, recurrent neural networks, autoencoders, or hybrid methods combining two or more deep learning techniques Ezugwu et al. (2022).

This project aims to explore and organize the Irish Government Circulars by applying three unsupervised learning techniques: K-means, Eigenspace-based fuzzy c-means, and an adapted version of the Long-term short-term memory neural network (LSTM)). This will be done using three established document representation techniques: TF-IDF (term frequency, inverse document frequency), Word2Vec, and BERT (bidirectional encoder representations from transformers). During this exercise, we also aim to compare and contrast the results obtained and evaluate the performance of the different document representation techniques and the different clustering algorithms employed to perform this task.

Given the above, the research question that this paper aims to answer is as follows: Which combination of document representation out of the TF-IDF, Word2Vec, and BERT together with clustering algorithm out of K-Means, Eigenspace-based fuzzy c-means, and Long-term short-term memory network (LSTM) would provide better clustering results for the Irish Government Circulars.

2 Related Work

Clustering and document clustering topics are extensively covered in the academic literature. While data clustering and document clustering employ similar techniques, document clustering is distinct due to the unique nature of text data. These distinctive characteristics include high dimensionality, given by the fact that each document is represented by a potentially large number of distinct words or terms; sparsity, the result of the fact that the majority of documents contain only a small subset of the total vocabulary; semantic diversity, as the data must take into account semantic similarities between words and comprehend the meaning of the context and ambiguity. Specialised approaches and procedures customised to the characteristics of text data and the particular goals of the clustering task are required to address these issues.

Document clustering typically involves a three step process: preprocessing and data cleaning step, representing documents in a vector space model, and running the chosen clustering algorithms. As a preparation for the last step dimensionality reduction techniques can also be used to reduce the dimensionality data used in the clustering step thus decreasing the computing cost and increasing the capacity of the clustering algorithms to generalize the data to avoid overfitting. The results can be assessed by measuring the performance of a chosen clustering method using established techniques.

Although many studies focus on enhancing document clustering, for this paper we will focus on a few articles that explore the three specific clustering methods used in our exercise: K-means, Eigenspace-based Fuzzy C-Means (EFCM), and the Long Term Short Term Memory (LTSTM) neural network.

Before proceeding further, we should mention several recent reviews on document clustering which are noteworthy. Cozzolino and Ferraro (2022) offer a comprehensive overview of document clustering techniques. Gabralla and Chiroma (2020) present an excellent review of the status of deep learning for document clustering, providing an extensive survey of recent work in this area, detailed tables of deep learning algorithms and their comparisons, the datasets used, performance metrics, vectorization methods, and application domains. Asudani et al. (2023) deliver an extensive review focused on word embedding models within a deep learning context, summarizing the main word embedding and deep learning models currently in use, and including a list of prominent datasets, tools, APIs, and key publications.

2.1 Data cleaning

Before transforming the text into numerical vectors it is necessary to remove unnecessary characters and/or words and to standardise them. The techniques used to perform this can include filtering, tokenization, stemming and/or lemmatization Vijayarani and Ilamathi (2015).

Tokenization refers to the process of dividing documents into smaller units known as tokens. The conventional approach entails dividing the text into its individual words (n-grams) using the white space as a separator. During the filtering process, special letters, punctuation and words that lack semantic meaning, such as pronouns and conjunctions (also known as stopwords), are removed. Stemming is the procedure of reducing each word to its base form by eliminating prefixes and suffixes. Lemmatization is a more advanced and complex procedure that seeks to identify and extract the root form of a word. This technique typically relies on dictionaries and can result in better outcomes compared to performing only stemming Balakrishnan and Ethel (2014).

To note that while text cleaning is essential for all embedding models, the extent and specific techniques can vary. Traditional models such as TF-IDF and Word2Vec benefit from more extensive cleaning to remove noise and standardize text. Models that use subword level embedding such as transformer-based models like BERT generally require less extensive text cleaning but still benefit from basic preprocessing such as removing typos and spelling mistakes to ensure consistency Kumar et al. (2020). The reason for this is that these models are inherently more robust to text variations and errors and can deal with case sensitivity, punctuation, and morphological variations more effectively due to their ability to learn from subword units.

2.2 Text-to-vector representation

The conversion of text documents into vectors is perhaps the most essential step in any NLP activity, as the accuracy of the analysis relies on the quality of the data source representation. The methods employed for representing text can be categorised into three main categories (see 1): conventional or count/frequency based models, distributional or static word embedding, and contextual word embedding.

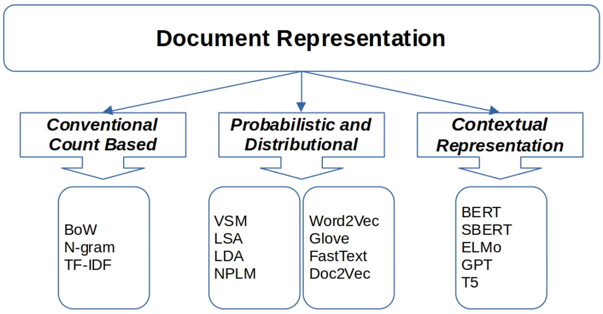


Figure 1: Word to vector representation models. Source: Adapted from Asudani et al. (2023)

2.2.1 Traditional count-based models

The primary traditional count-based models offer a basic depiction of the document which disregards the syntax and word arrangement inside it. These models include the bag of words (BoW), n-gram, and term frequency-inverse document frequency (TF-IDF) models. The difference between them is that while in the Bag of Words (BoW) model, each word is represented by its frequency count in the document the n-gram is considering contiguous sequences of words or characters while the TF-IDF is taking into account the significance of a word in a document compared to a collection of documents.

One of the most used models in the word clustering field is the TF-IDF Kumbhar et al. (2020); Murfi et al. (2024); Purohit et al. (2023); Gabralla and Chiroma (2020) and has been used in conjunction with many clustering algorithms. This is due to the fact the weighting of the words is done in a document is simple and efficient and has the benefit of a lower-dimensional and less sparse vector representation compared to the Bag-of-Words (BoW) model as the TF-IDF score for each term in a document is determined by multiplying its term frequency (TF) by its inverse document frequency (IDF) thus making the process of splitting the data in individual clusters easier. The TF component measures the frequency of a term in the document while the IDF component measures the rarity of a term across all documents in the collection. This is done by calculating the logarithm of the ratio between the total number of documents and the number of documents that contain the term.

2.2.2 Probabilistic and distributional-based models

However, even though these conventional approaches have the advantage of being easy to understand, relatively straightforward to compute and cost-effective in terms of computer resources utilised, these approaches have the drawback that they do not take into account the sequential arrangement and the contextual usage of words in the documents. These approaches are also affected by the polysemy phenomenon as various words can have identical meanings. Three often used models that are trying to overcome this are the VSM (vector space model), the LSA (latent semantic analysis), and the LDA (Latent Dirichlet Allocation). LDA is a generative probabilistic model used to discover latent topics within a collection of documents where each document is represented as a mixture of topics. The documents are represented as vectors of topic probabilities and each element in the vector corresponds to the proportion of a particular topic within the document. These topic vectors serve as features for clustering algorithms which promote grouping documents with similar topic distributions into clusters. This helps make the resulting clusters often more interpretable given the fact that they should reflect the thematic content of the documents Ahmed et al. (2023).

Other word embedding methods like as Word2Vec, GloVe, and fastText acquire compact, lower-dimensional representations of words by considering how they are used in context among a vast collection of texts Ravi and Kulkarni (2023). Word2Vec [1], also known as word-to-vector, is a technique that develops distributed representations of words by analysing their context within a large text corpus. This method effectively captures both syntactic and semantic relationships in the text leading to more meaningful clusters. This characteristic has made it one of the most widely used embedding techniques for document clustering Gabralla and Chiroma (2020). Doc2Vec is an advanced version of Word2Vec, developed by Google, which extends Word2Vec's functionality by learning distributed representations of entire documents or sentences, not just individual words Le and Mikolov (2014).

GloVe, another popular word embedding technique, creates word representations by factorizing a matrix of word co-occurrences. It aims to overcome a limitation of Word2Vec, which is its lack of consideration for global statistical information. GloVe embeddings are pre-trained models using a vocabulary of 400,000 words derived from Wikipedia Pennington et al. (2014).

2.2.3 Contextual representation-based models

Contextual representation models seek to comprehend and capture contextual information from textual material. Some examples of these models are Embeddings from Language Models (ELMo)[2], Generative Pretrained Transformer (GPT) produced by OpenAI[3], and Bidirectional Encoder Representations from Transformers (BERT) Devlin et al. (2019). All three models have undergone pretraining using large text corpora to acquire contextual representations of words, phrases, and sentences. The primary distinction among them is in their respective approaches: ELMo utilises a feature-based approach by including pretrained representations as supplementary features. GPT employs a fine-tuning approach by using task-specific parameters trained exclusively on downstream tasks. BERT incorporates a multi-layer bidirectional transformer encoder in its architecture.

[1] See:https://word2vec.com/

[2] See:https://github.com/allenai/allennlp-models

[3] See:https://openai.com/index/language-unsupervised/

These contextual models, particularly BERT and ELMo Asudani et al. (2023), have proven popular in the document clustering field due to the richness of the vectors captured.

2.3 Clustering

Document clustering aims at partitioning a corpus of N documents into k number of clusters to identify k homogeneous groups of documents. There have been many clustering techniques developed and these techniques can be classified in multiple manners including hierarchical and partitional models Ezugwu et al. (2022), distance measure-based, statistical and neural networks Károly et al. (2018), or prototype-based, graph-based, hierarchical, and model-based techniques Cozzolino and Ferraro (2022). Furthermore, clustering methods can be categorised into hard clustering where each document is allocated to a single cluster, and soft clustering where a document can be part of numerous clusters with different levels of participation Cozzolino and Ferraro (2022).

In keeping with the focus of our current paper and given the multitude of clustering algorithms, in the following part, we will specifically focus on three widely used clustering algorithms: K-means which has the advantage of being computationally efficient, Long Short-Term Memory (LSTM), and Eigenspace-based Fuzzy C-means (EFCM), both recognised for their high accuracy.

To note that before the clustering algorithm is deployed a dimensionality reduction method such as principal component analysis (PCA), truncated singular-value decomposition (truncated-SVD), t - Distributed Stochastic Neighbor Embedding (t-SNE) or non-negative matrix factorisation (NMF) is used to alter the large collection of dimensions into a smaller one that retains the characteristics of the larger dataset thus reducing the resources needed to compute the distance between the clusters and tackle the curse of dimensionality Kumbhar et al. (2020); George and Sumathy (2023); George (2022).

2.3.1 K-means clustering

K-means clustering is one of the most used unsupervised clustering algorithms for text clustering. It was firstly introduced in 1957 by Lloyd (1982) and popularised by MacQueen et al. (1967). It uses distances to group data points together and is especially useful for grouping big sets of documents due to its simplicity and efficiency Xu et al. (2024); Gabralla and Chiroma (2020). The algorithm's objective is to divide a collection of N documents into K clusters, with each document assigned to the cluster that has the closest centroid. The centroids are first chosen randomly, and the method progressively improves these centroids to minimise the variance within each cluster.

The drawbacks of K-means include its susceptibility to the original choice of centroids and the fact that the number of clusters needs to be stated from the beginning. Inadequately selected starting centroids might result in poor clustering outputs. It may also need numerous iterations with varied initialisations to attain better results. Pre-specificating the number of clusters might also be challenging when the optimum number of clusters is not known beforehand.

Various distance metrics are used to compute the distance that determines the similarity between documents. Some of the most used are the cosine distance, which quantifies the cosine of the angle formed by two vectors, and the euclidean distance, which calculates the direct distance between two points in a multidimensional space. Research has shown that, due to its ability to mitigate the consequences of different document lengths, the cosine distance provides better performance over the euclidean distance in the text clustering domain Cozzolino and Ferraro (2022).

2.3.2 Fuzzy c-means (FCM)

Fuzzy c-means (FCM) and its derivatives such as Eigenspace-based Fuzzy C-means (EFCM) is a soft clustering technique which assigns each document a probability of membership to all clusters to reflect the inherent ambiguity and overlap often present in complex datasets Bezdek (1973). This gives it the ability to capture semantic relationships and thematic overlaps between documents which can result in more nuanced and accurate results. EFCM includes eigenspace decomposition techniques such as principal component analysis (PCA) or singular value decomposition (SVD) in order to reduce the sparsity caused by high-dimensional data vectors such as text documents.

The advantage provided by FCM is adding to the computational complexity, parameter sensitivity, interpretability, scalability and noise sensitivity which require careful tuning of the algorithm and a more detailed need of data prepossessing Aditiyo et al. (2023).

2.3.3 Long Short Term Memory Networks (LSTM)

Long Short-Term Memory Networks (LSTMs) are an enhanced type of recurrent neural network (RNN). They are being designed to address the limitations of traditional RNNs, particularly the vanishing gradient problem. This issue, common in standard RNNs, impedes the network's ability to learn long-range dependencies in sequential data as gradients used in backpropagation tend to diminish exponentially through time, making it difficult to capture relationships over long sequences. This weakness of the RNN is tackled using memory cells and gating mechanisms. This improvement has made LSTMs one of the most preferred deep learning tools for text classification and other NLP tasks, owing to their ability to understand and process complex sequences of text data Gabralla and Chiroma (2020).

However, using LSTM for text clustering comes with its own difficulties as they requires high computational complexity and extensive prepossessing requirements. They are also very sensitive to hyperparameters and the fact that they have been designed for solving classification problems rather than unsupervised learning. All these disadvantages can limit the practicality and effectiveness of using LSTMs for text clustering. To overcome these limitations LSTM is often embedded with classical clustering algorithms such as k-means in a technique named deep embedded clustering. This combines the strengths of deep learning for feature representation with clustering algorithms to improve clustering performance, especially in high-dimensional data like text, and can significantly enhance the clustering results by leveraging the rich feature representations learned by LSTMs and refining these representations to form better separated clusters Akram et al. (2022); Guan et al. (2020).

3 Methodology

Document clustering has been successfully applied across various fields such as news categorization and medical document organization Gabralla and Chiroma (2020) but there has been no research conducted specifically on the clustering of Irish Government Circulars. These circulars are official written statements that provide detailed information and guidelines on laws, procedures, and policies. Currently, there are approximately three thousand four hundred such circulars available online [4], predominantly in PDF format.

The present work proposes to cluster the corpus of the circulars and to evaluate the clustering performance of three document vectoring techniques (TF-IDF, Word2Vec and BERT) in conjunction with three clustering techniques (K-means, EFCM, LSTM). We hope that effectively managing to cluster similar documents would help us in getting a better understanding of the contents of these circulars and be able to use this at a later stage in helping with information retrieval from them.

As per Figure 2 in the next part, steps we will briefly describe the steps taken to achieve this.

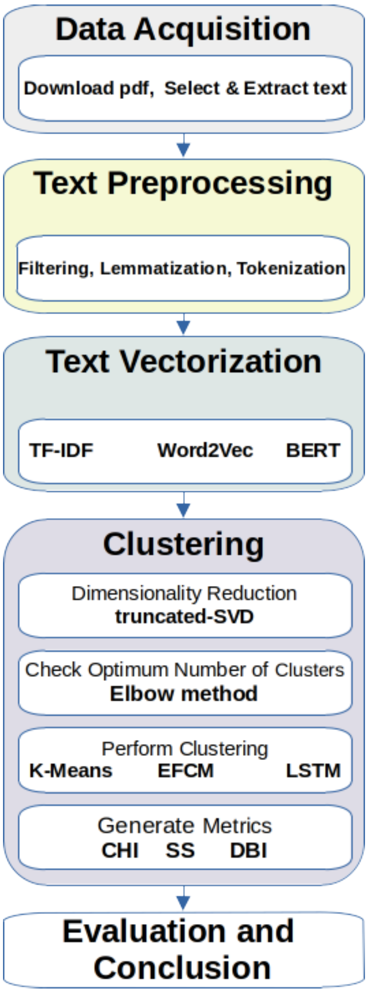


Figure 2: Project Map

3.1 Data Acquisition

The initial step is to download and extract the text content from these documents. This is a critical step because the documents are in three different formats (pdf, doc, docx). This format is not immediately suitable for text processing and analysis. In this step we will also select only the first document for each circular and remove a small number of documents which could not be read or which were restricted for download.

3.2 Text Preprocessing

In the second step, the text will be further preprocessed to make it suitable for vectorization and clustering. The main steps taken to ensure this will be filtering the unwanted characters and words, reducing the words to their root form to ensure that different forms of a word are treated as a single entity (lemmatization) and splitting the text into individual words or tokens (tokenization).

3.3 Text Vectorization

In the third step, the text will be transformed and vectorized to make it suitable for clustering. Three text vectorization techniques will be used: term frequency, inverse document frequency (TF-IDF), word-to-vector (Word2Vec), and the Bidirectional Encoder

[4] See: `https://www.gov.ie/en/circulars/`

Representations from Transformers (BERT). It should be noted that the cleaning done in the previous step is not suitable for the BERT model which favours much lighter cleaning and has its own tokenizer.

3.4 Clustering

3.4.1 Dimensionality Reduction

The clustering part of the project will be further split into four steps. In the first step, truncated-SVD will be used for dimensionality reduction. This will optimize the resources needed and reduce the risk of overfitting the data. The truncated-SVD has been selected as it has been proven to be superior to other models in dealing with text data Kumbhar et al. (2020); Kumar (2009). This is because it handles large sparse matrices like TF-IDF well while at the same time preserving relevant semantic information.
Its linear nature and computational efficiency make it a preferred choice over other dimensionality reduction techniques, especially for large-scale text data.

3.4.2 Get the Optimum Number of Clusters

The optimum number of clusters will be researched using the elbow model, silhouette metric, Calinsky Harabasz and the Davies-Bouldin Score. For calculating this we will use the TF-IDF vectorization to which the truncated-SVD dimensionality reduction was performed. We should note that choosing the optimal number of clusters involves a degree of subjectivity despite using multiple methods in conjunction. The reason for this is that each method evaluates clustering quality based on different criteria (e.g., compactness, separation, silhouette). These criteria might not always align leading to different suggestions for the optimal number of clusters. The metrics also have varying levels of sensitivity to noise, outliers and preprocessing variations in the data which can affect the perceived optimal number of clusters. This is why domain experts might prefer a different number of clusters based on their understanding of the data's significance and practical applications.

3.4.3 Perform Clustering

Three clustering techniques will be assessed during this exercise: K-means, enhanced fuzzy c-means (EFCM), and a version of the neural network Long Short Term Memory (LSTM) as part of the deep embedded clustering (DEC) approach. For the LSTM a three-step approach was taken. In the first step, we have pretrained an autoencoder to learn a compressed representation of the data. In the second step, we clustered a compressed representation of the data using k-means. In the final step we fine-tuned the autoencoder and cluster assignments simultaneously. To note that the hyperparameters for each model will be tuned by running various configurations of the model.

3.4.4 Generate Metrics

Three metrics will be used to measure the quality of the clusters: the Calinski-Harabasz index (CHI), the silhouette score (SS) and the Davies Bouldin index (DBI). By doing this we hope to obtain a comprehensive assessment of the obtained clustering quality as these metrics can balance each other's strengths and weaknesses and increase confidence in the validity of the results.

Each of the metrics selected has its strengths and weaknesses Gagolewski et al. (2021). The Calinski-Harabasz index evaluates the quality of clustering by considering the dispersion within clusters and the dispersion between clusters but tends to be biased towards a higher number of clusters. The silhouette score measures the quality of clustering by evaluating how similar each data point is to its own cluster when compared to other clusters and provides an easy to interpret number (between -1 and 1) showing how well separated and cohesive are the clusters obtained. Davies-Bouldin index measures clustering quality by evaluating the average similarity ratio of each cluster with its most similar cluster. It produces an index in which the lower values are interpreted as the best but can be sensible to outliers Hassan et al. (2021).

For illustration purposes we will also plot the clustering results using two widely used models: PCA (Principal Component Analysis) and the t-SNE (t-Distributed Stochastic Neighbor Embedding). Although both methods can reduce the data to two dimensions, they are achieving this using different techniques. PCA is linearly reducing the data dimensionality by finding the directions (principal components) that maximize variance. It is simpler and less computationally intensive but may miss non linear relationships. T-SNE on the other hand, is trying to preserve the local neighborhood structure of the data, mapping high-dimensional points that are close together to nearby points in a low-dimensional space. It has a higher computational cost and comes with the risk of distorting global relationships.

By executing these steps we hope to reach our goal of organizing these documents into meaningful clusters which can help in better categorization, retrieval and analysis of the circulars based on their content. We also hope to get an insight on the performance of each selected vectorization technique in conjunction with each of the clustering method used in this exercise.

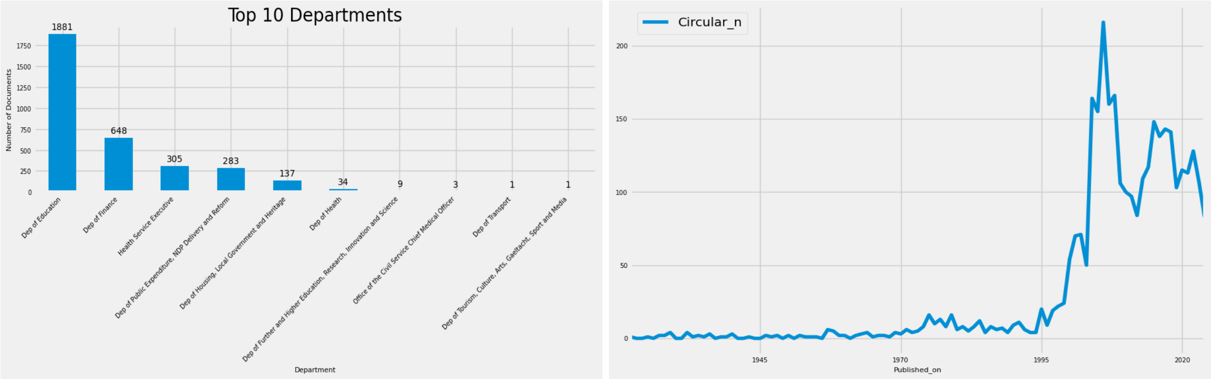
4 Design Specification

The Python programming language with the Jupyter Notebook interface was used at all stages of this research. This is because they are open-source tools for text documents processing and clustering. Some of the packages that were used include Pandas and NumPy for analytics tools, data manipulation and numerical processing, pdfplumber for extracting text from pdf documents, Natural Language Toolkit (NLTK) for natural language processing, Scikit-learn which was used for k-means clustering, dimensionality reduction and producing the cluster evaluation metrics, skfuzzy for running the fuzzy c-means and Tensorflow and Keras for deep learning. The packages used for visualising the results were matplotlib and plotly.

Below are the key design specification decisions taken throughout this project.

4.1 Data

There were 3,400 circulars on the website at the time of downloading the data (July 2024). After cleaning and removing the documents that were restricted for reading the final dataset had 3,304 left. Out of these 57 percent (1,881) were issued by the Department of Education 3a. The earliest document in the dataset is from the year 1922 but most of the documents are newer and have been published from 2004 on with the highest number (216) being published in 2006 as per 3b.



(a) Number of Documents per Department

(b) Number of Documents per Year

Figure 3: Selected Data Statistics

The noticeable uneven distribution of circulars across various departments coupled with the absence of circulars from many departments raises concerns that not all issued circulars are being published or made available on this platform. This observation suggests a potential gap in the dataset where certain departmental communications might be underrepresented or missing entirely. As clustering relies on distinguishing patterns and grouping similar items together this could complicate the clustering process. This is because that if the documents are too similar because they originate from the same departments with overlapping content, it may be challenging to identify distinct clusters. Therefore the cluster result obtained may not accurately reflect meaningful differences but rather group documents based on superficial similarities.

4.2 Cleaning Data

A systematic and standardized process was employed for cleaning the text data, ensuring consistency and accuracy in preparing the corpus for analysis. This text-cleaning process was essential to eliminate noise and irrelevant elements. This involved removing the stopwords, punctuation, numbers and special characters. The text was also subjected to lemmatization which involves transforming words to their base or dictionary form known as a lemma which is grouping different forms of a word under a single representation. This helps in improving the accuracy of the analysis by treating variations of a word as a single entity.

To note that the resulting cleaned text was used with TF-IDF and Wod2Vec as both models rely on the frequency and co-occurrence of words to calculate the distance of various documents. A much lighter cleaning was done for BERT as this model is designed to understand the context of words in a sentence. Cleaning text by removing stopwords, punctuation, or converting to lowercase might strip away useful semantic information that BERT can leverage for better embeddings thus altering the context and reduce the effectiveness of the embeddings.

4.3 Vectorization

4.3.1 TF-IDF

When setting up the TF-IDF we have aimed at enhancing its ability to capture the distinctive characteristics of each document by filtering out the words that were too rare (appearing in less than 1% of the documents) or too frequent (appearing in more than 97% of the documents). By doing this we improved the overall robustness and interpretability of the results as the words included have a more balanced presence in the corpus.

4.3.2 Word2Vec

There are several ways to vectorize a Word2Vec model for document representation which include averaging the word vectors and TF-IDF weighted averaging where each word vector is weighted by its TF-IDF score. We have opted for the concatenation of min, max, and average vectors for the words in a document which we hope that it will help us to capture different aspects of the word distributions in our corpus. As we have set the hyperparameter vector_size at 100 we produced word vectors of size 100. This resulted in document representation with 300 vectors for each document.

4.3.3 BERT

BERT (Bidirectional Encoder Representations from Transformers) uses a multi-layer transformer architecture. The base model of BERT, which is BERT-base, includes 12 transformer layers which are also called transformer blocks, 12 attention heads per layer and a hidden size of 768 which means that each token is represented by a vector of 768 dimensions. To obtain the vector representation we have averaged the token embeddings along the token dimension.

Other ways we could have performed this include using the embedding of the [CLS] token as the document embedding, concatenating/summing Last N Layers or using specialized models such as SBERT (sentence BERT).

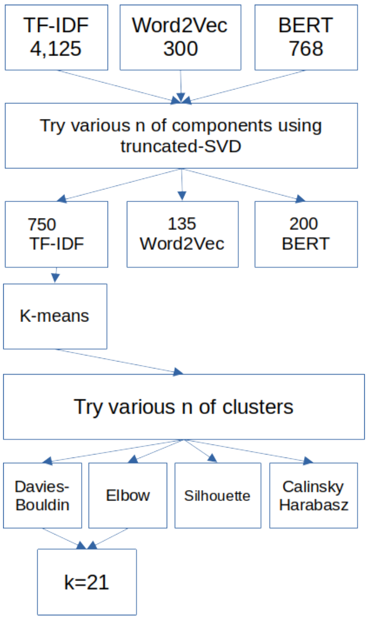
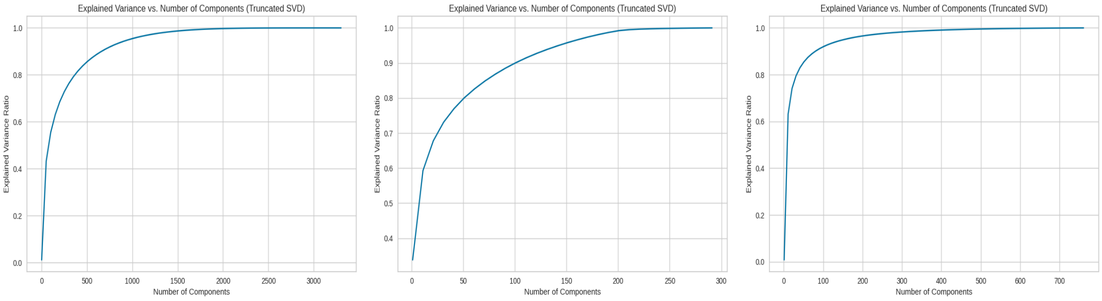


Figure 4: Dimensionality Reduction and Optimizing the Number of Clusters Process Map

4.4 Dimensionality reduction

The dimensionality reduction was done by performing a search for the optimal number of components to use in a TruncatedSVD dimensionality reduction process and plotting the results. TruncatedSVD was chosen as it is better suited for sparse matrices as TF-IDF.

As shown in 4 and 5a, by performing this process we have achieved a significant reduction in dimensionality while retaining most of the original data variance. In the case of TF-IDF for example, the number of features was reduced from 4,125 to 750 while retaining 92% of the explained variance. In the case of Word2Vec 5b, we have reduced the number of components from 300 to 135 and keeping 94% of the explained difference while in the case of BERT 5c, we have reduced the number of components to 200 from 768 and kept 96% of the explained difference. This optimization improved the computational efficiency and increased the potential generalization of the models trained on this reduced data.

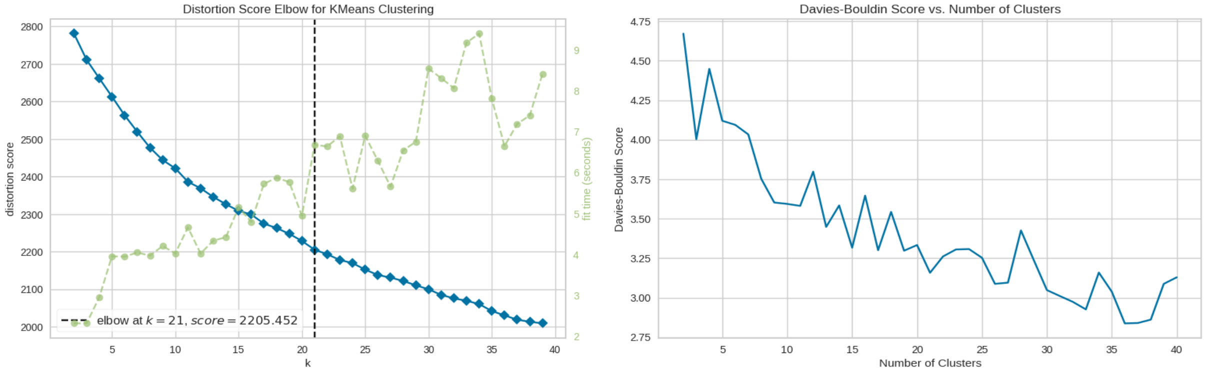


(a) TF-IDF explained variance vs n of components (b) Word2Vec explained variance vs n of components (c) BERT explained variance vs n of components

Figure 5: Explained variance vs Number of components for each embedding technique

4.5 Check for the optimum number of clusters

In order to determine the optimal number of clusters for our analysis, we have evaluated the data using four different clustering evaluation methods 4: the Elbow method 6a, Silhouette score, Davies-Bouldin index, and the Calinski-Harabasz index 6b. Given the fact that each method has different underlying assumptions and metrics the recommendations obtained varied. For our exercise, we proceeded with the number of clusters suggested by the Elbow method which was supported by the validation provided by the Davies-Bouldin index. They indicated that the number of clusters chosen (21) was reasonable given the fact that they were compact and well-separated.



(a) Distortion Score Elbow for KMeans Clustering (b) Davies-Bouldin index for KMeans Clustering

Figure 6: Selecting the Optimum number of clusters

4.6 Algorithm tuning

For K-Means clustering careful consideration was given to initialization, convergence, and reproducibility. We have used k-means++ for initialization as it generally leads to faster and more accurate clustering. We have also tried to enhance the convergence and robustness against poor initialization by setting the maximum number of iterations that the algorithm will perform during the optimization process at 500 (default being 300) and the number of times the algorithm will be run with different centroid seeds at 20 (default being 10).

For the EFCM function, we have incorporated best practices for datasets where we want to obtain distinct clusters. This was achieved by setting the fuzziness factor to 1.1. The algorithm was set to stop when the improvement between iterations was below 0.005 in order to prevent computations when the model has already stabilized. The maximum number of iterations for the clustering process was set at 1000, allowing the algorithm ample time to converge.

For the LSTM we have combined an LSTM-based autoencoder with a custom clustering mechanism inspired by the Deep Embedded Clustering (DEC) technique. The process involved pretraining an autoencoder to reduce dimensionality, initializing cluster centres with k-means, and then refining these clusters using a clustering layer. Due to time constraints, the model used is an exploratory model in need of improvement which is reflected in the results obtained.

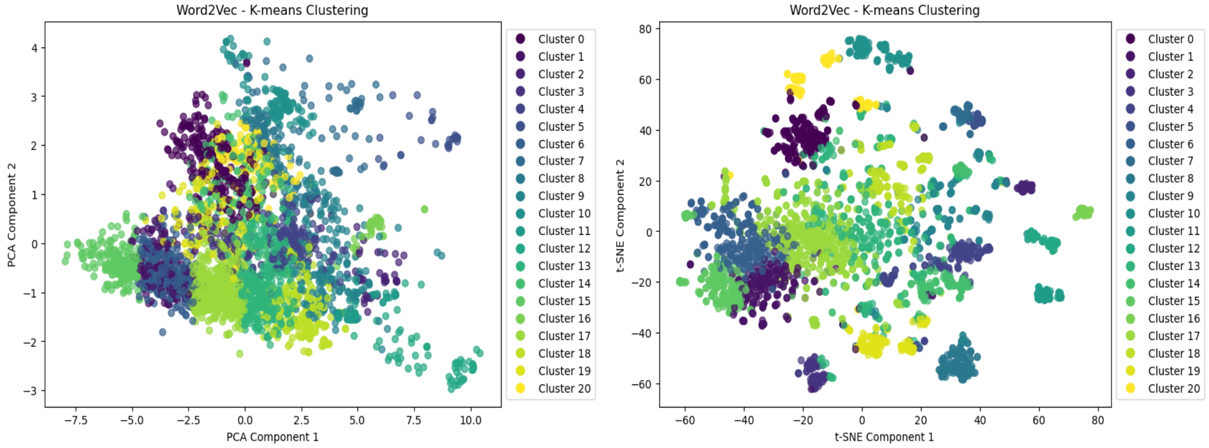
5 Evaluation

The quality of the clusters obtained was measured using three metrics: the Calinski-Harabasz Index, the Silhouette Score and the Davies Bouldin Index. These results can be found in Table 1. Although using these metrics has its limits they do give a clear indication of the performance of each model.

It should be noted that these metrics do not give any indication regarding the computational power needed to perform the clustering or the text vectorization. Nonetheless, our experience was that, as expected, the higher the complexity the higher the resources needed to perform these calculations with BERT for vectorization and LSTM model for clustering being the most expensive in this regard while the TF-IDF and the K-means being the cheapest.

Table 1: Clustering Evaluation Results

Vectorization Model	**Clustering Model**	**Calinski-H Index**	**Silhouette Score**	**Davies Bouldin Index**
TF-IDF	K-means	47.341056	0.080188	3.368405
TF-IDF	EFCM	10.359452	-0.013488	5.346192
TF-IDF	LSTM	7.916713	0.001586	17.216528
Word2Vec	K-means	188.047219	0.116202	2.444379
Word2Vec	EFCM	182.461331	0.113900	2.703798
Word2Vec	LSTM	33.59888	0.001298	8.931823
BERT	K-means	176.916251	0.086493	2.279753
BERT	EFCM	160.486638	0.084421	2.483703
BERT	LSTM	20.725396	-0.040572	10.159768



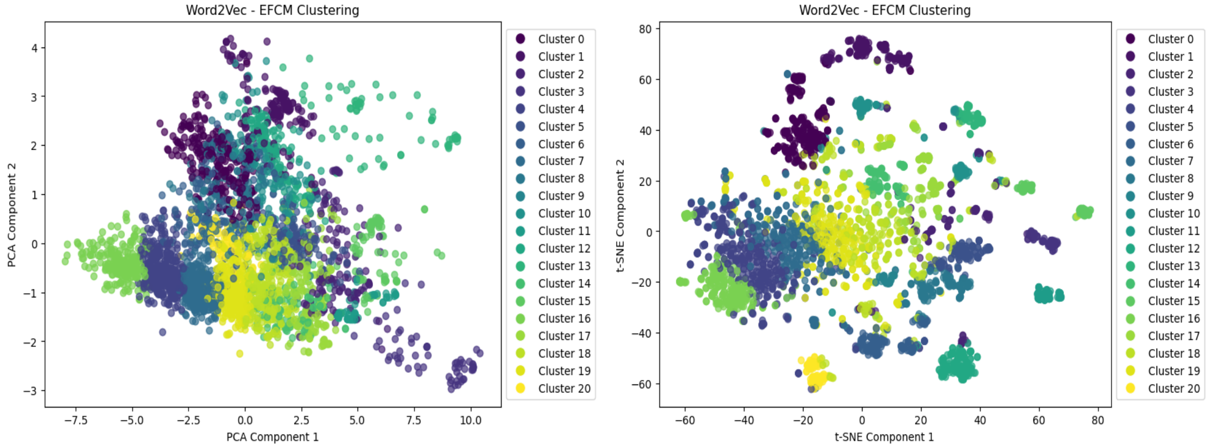
(a) K-means using Word2Vec with PCA for plotting

(b) K-means using Word2Vec with t-SNE for plotting

Figure 7: K-means using Word2Vec

5.1 K-means clustering results

The k-means algorithm was the best performing when compared with the other two models used. It has obtained relatively good performance with all three vectorization models 1. The best scores were obtained with the Word2Vec vectorization having the highest silhouette and CHI scores (0.116 and 188.05) and the lowest DBI score (2.44, lowest being the best). For illustration purposes, we have also plotted the clustering results for the highest obtained score (Word2Vec and k-means) using both the PCA 7a and the t-SNE 7b. In both cases, we can see that the clusters are distinct and well defined but there is ample space for further improvement.



(a) EFCM using Word2Vec with PCA for plotting

(b) EFCM using Word2Vec with t-SNE for plotting

Figure 8: EFCM using Word2Vec

5.2 EFCM clustering results

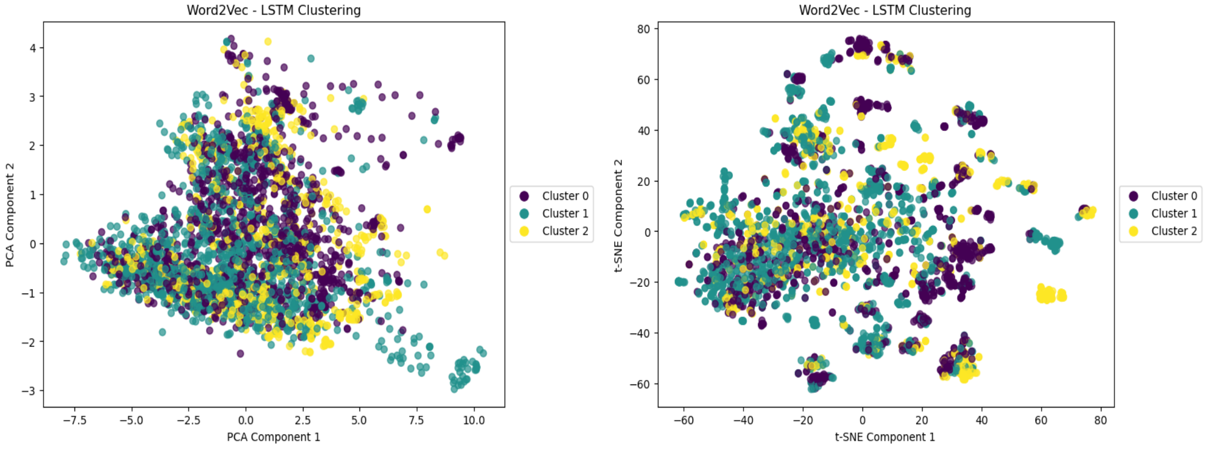
The best results with the EFCM clustering were obtained using the Word2Vec vectorization 1 (CHI: 182.46, SS: 0.1139, DBI: 2.703) as it has the highest Calinski-Harabasz Index and the lowest Davies-Bouldin Index. Plotting the data using either the PCA 8a or the t-SNE 8b would indeed indicate that in our case EFCM did reasonably well in conjunction with the Word2Vec.

However, the negative Silhouette Score in conjunction with the TF-IDF (-0.0135) indicates that the performance was not ideal in that case with poor separation between clusters and high within-cluster dispersion.

5.3 LSTM clustering results

The results obtained with LSTM attest to the fact that it was an exploratory model to assess the viability of this technique. Despite of the complexity of the model employed it has failed to form meaningful clusters. As the other two models it has performed best with the Word2Vec 1 (CHI: 33.598, SS: 0.001298, DBI: 8.931) but the general indication was that, overall, there was little difference between the tree vectorization models when considering the performance of this clustering technique.

Plotting the data using either the PCA 9a or the t-SNE 9b would also indicate that in our case LSTM model may not have been the best choice as a clustering algorithm or that needs further improvement.



(a) EFCM using Word2Vec with PCA for plotting

(b) EFCM using Word2Vec with t-SNE for plotting

Figure 9: LSTM using Word2Vec

5.4 Discussion

The results of the clustering analysis revealed that the best and most balanced performance in terms of both the results obtained and the resources employed was achieved using the K-means algorithm in conjunction with Word2Vec for vectorization. However, despite this combination yielding the most favourable outcomes, the performance metrics indicate that there is still considerable room for improvement. The metrics obtained were not exceptionally high particularly as obtained by the silhouette score, suggesting that the clustering results, while better than those obtained with other methods, are far from optimal.

Interestingly, the LSTM model, which was selected due to its advanced capabilities and complexity, produced some of the poorest results despite the resources needed to develop and run it. The disappointing performance can likely be attributed to the inherent complexity, lack of familiarization of the author with this model and the unavailability of more time to further tune this complex model. All of this have resulted in inconsistencies in the clustering output obtained with this model.

Our exercise showed that several factors critically impact the clustering results with the clustering algorithm choice first and the vectorization choice secondly being the most significant. Multiple strategies can be employed to enhance the performance of these clustering models such as refining the numbers of clusters, testing alternative dimensionality reduction methods, as well as different number of dimensions retained and testing different clustering algorithms. Improving the preprocessing pipeline, such as by refining the text cleaning steps or exploring alternative vectorization methods, might also lead to better clustering results.

All three models used could also be improved by fine-tuning the hyperparameters. This is particularly the case of the LSTM where this can include changing the distance metric used, the number of layers, or the learning rate. These changes might help unlock the potential of these complex models.

Finally, it should be noted that to assist with the information retrieval from our corpus of documents the main topic of each cluster can be extracted and each cluster labelled with its topic. This can be done using established techniques such as the LDA. As this was outside the scope of this exercise further work with this dataset may also include this step.

6 Conclusion and Future Work

This study demonstrates the potential of advanced text clustering methods to improve information retrieval in large, unstructured document collections. By leveraging the strengths of different document representation techniques and clustering algorithms, we were able to enhance the organization and accessibility of the Irish Government circulars. The insights gained from this research can serve as a foundation for future work in the field, potentially leading to even more effective methods for managing and retrieving information from these large text datasets.

During the data collection process, we not only gathered the primary data needed for this specific exercise but also successfully extracted additional data points from both the website and the associated PDF documents. While these extra data points fall outside the immediate scope and objectives of the current analysis, they present a valuable opportunity for further exploration and in-depth analysis of the dataset.

This supplementary information can enhance our understanding of the corpus and may provide insights that could be beneficial for future projects or research endeavours such as identifying trends, uncovering hidden patterns, or supporting more complex analyses, These additional data points can serve as a resource that can be leveraged to expand the scope of our analysis and contribute to a more comprehensive understanding of the subject matter.

References

Aditiyo, S. P., Sumarminingsih, E. and Fitriani, R. (2023). Fuzzy c-means in content-based document clustering for grouping general websites based on their main page contents, *ComTech: Computer, Mathematics and Engineering Applications* **14**(2): 119–127.

Ahmed, M. H., Tiun, S., Omar, N. and Sani, N. S. (2023). Short Text Clustering Algorithms, Application and Challenges: A Survey, *Applied Sciences* **13**(1): 342. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
URL: *https://www.mdpi.com/2076-3417/13/1/342*

Akram, M. W., Salman, M., Bashir, M. F., Salman, S. M. S., Gadekallu, T. R. and Javed, A. R. (2022). A novel deep auto-encoder based linguistics clustering model for social text, *Transactions on Asian and Low-Resource Language Information Processing* .

Asudani, D. S., Nagwani, N. K. and Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review, *Artificial Intelligence Review* **56**(9): 10345–10425.
URL: *https://link.springer.com/10.1007/s10462-023-10419-1*

Balakrishnan, V. and Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances, *Lecture Notes on Software Engineering* **2**(3): 262–267.
URL: *http://www.lnse.org/show-34-165-1.html*

Bezdek, J. C. (1973). *FUZZY-MATHEMATICS IN PATTERN CLASSIFICATION.*, Cornell University.

Cozzolino, I. and Ferraro, M. B. (2022). Document clustering, *WIREs Computational Statistics* **14**(6): e1588.
URL: *https://wires.onlinelibrary.wiley.com/doi/10.1002/wics.1588*

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] version: 2.
URL: *http://arxiv.org/abs/1810.04805*

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I. and Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, *Engineering Applications of Artificial Intelligence* **110**: 104743.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S095219762200046X*

Gabralla, L. and Chiroma, H. (2020). Deep Learning for Document Clustering: A Survey, Taxonomy and Research Trend, *Journal of Theoretical and Applied Information Technology* **98**(22).

Gagolewski, M., Bartoszuk, M. and Cena, A. (2021). Are cluster validity measures (in) valid?, *Information Sciences* **581**: 620–636.

George, A. (2022). *Python text mining: perform text processing, word embedding, text classification and machine translation*, BPB, Delhi.

George, L. and Sumathy, P. (2023). An integrated clustering and BERT framework for improved topic modeling, *International Journal of Information Technology* **15**(4): 2187–2195.
URL: *https://link.springer.com/10.1007/s41870-023-01268-w*

Guan, R., Zhang, H., Liang, Y., Giunchiglia, F., Huang, L. and Feng, X. (2020). Deep feature-based text clustering and its explanation, *IEEE Transactions on Knowledge and Data Engineering* **34**(8): 3669–3680.

Hassan, I. H., Abdullahi, M. and Ali, Y. (2021). Analysis of techniques for selecting appropriate number of clusters in k-means clustering algorithm, *no. November* .

Kumar, A. C. (2009). Analysis of unsupervised dimensionality reduction techniques, *Computer science and information systems* **6**(2): 217–227.

Kumar, A., Makhija, P. and Gupta, A. (2020). Noisy Text Data: Achilles' Heel of BERT. arXiv:2003.12932 [cs].
URL: *http://arxiv.org/abs/2003.12932*

Kumbhar, R., Mhamane, S., Patil, H., Patil, S. and Kale, S. (2020). Text Document Clustering Using K-means Algorithm with Dimension Reduction Techniques, *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Coimbatore, India, pp. 1222–1228.
URL: *https://ieeexplore.ieee.org/document/9137928/*

Károly, A. I., Fullér, R. and Galambos, P. (2018). Unsupervised Clustering for Deep Learning: A tutorial survey, *Acta Polytechnica Hungarica* **15**(8): 29–53.
URL: *http://acta.uni-obuda.hu/Karoly$_F$uller$_G$alambos$_8$7.pdf*

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. https://arxiv.org/abs/1405.4053 [cs].
URL: *https://arxiv.org/abs/1405.4053*

Lloyd, S. (1982). Least squares quantization in pcm, *IEEE transactions on information theory* **28**(2): 129–137.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, Oakland, CA, USA, pp. 281–297.

Murfi, H., Agung, Y. J., Nurrohmah, S., Satria, Y., Za'in, C. and Rahayu, D. (2024). Eigenspace-based Fuzzy C-Means with Large Language Model BERT for Topic Detection, *Journal of Big Data* .
URL: *https://www.researchsquare.com/article/rs-3637575/v1*

Pennington, J., Socher, R. and Manning, C. D. (2014). Glove: Global vectors for word representation.
URL: *https://nlp.stanford.edu/projects/glove/*

Purohit, K., Vats, S., Saklani, R., Kukreja, V., Sharma, V. and Yadav, S. P. (2023). Improvement in K-Means Clustering for Information Retrieval, *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE,

Coimbatore, India, pp. 1239–1245.
URL: *https://ieeexplore.ieee.org/document/10193031/*

Ravi, J. and Kulkarni, S. (2023). Text embedding techniques for efficient clustering of twitter data, *Evolutionary Intelligence* **16**(5): 1667–1677.
URL: *https://link.springer.com/10.1007/s12065-023-00825-3*

Subakti, A., Murfi, H. and Hariadi, N. (2022). The performance of BERT as data representation of text clustering, *Journal of Big Data* **9**(1): 15.
URL: *https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00564-9*

Vijayarani, D. S. and Ilamathi, J. (2015). Preprocessing Techniques for Text Mining - An Overview, *International Journal of Computer Science & Communication Networks* **5**(1): 7–16.
URL: *https://www.researchgate.net/publication/339529230_Preprocessing_Techniques_for_Text_Mini*

Xu, Q., Gu, H. and Ji, S. (2024). Text clustering based on pre-trained models and autoencoders, *Frontiers in Computational Neuroscience* **17**: 1334436.
URL: *https://www.frontiersin.org/articles/10.3389/fncom.2023.1334436/full*