International Journal of Cognitive Computing in Engineering 6 (2025) 100–108

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: http://www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/

Text clustering with large language model embeddings

Alina Petukhova [a,*], João P. Matos-Carvalho [a,b], Nuno Fachada [a,b]

[a] COPELABS, Lusófona University, Campo Grande, 376, Lisbon, 1700-921, Portugal
[b] Center of Technology and Systems (UNINOVA-CTS) and Associated Lab of Intelligent Systems (LASI), Caparica, 2829-516, Portugal

ARTICLE INFO

MSC:
68T50
62H30

Keywords:
Text clustering
Large language models
Text summarisation

ABSTRACT

Text clustering is an important method for organising the increasing volume of digital content, aiding in the structuring and discovery of hidden patterns in uncategorised data. The effectiveness of text clustering largely depends on the selection of textual embeddings and clustering algorithms. This study argues that recent advancements in large language models (LLMs) have the potential to enhance this task. The research investigates how different textual embeddings, particularly those utilised in LLMs, and various clustering algorithms influence the clustering of text datasets. A series of experiments were conducted to evaluate the impact of embeddings on clustering results, the role of dimensionality reduction through summarisation, and the adjustment of model size. The findings indicate that LLM embeddings are superior at capturing subtleties in structured language. OpenAI's GPT-3.5 Turbo model yields better results in three out of five clustering metrics across most tested datasets. Most LLM embeddings show improvements in cluster purity and provide a more informative silhouette score, reflecting a refined structural understanding of text data compared to traditional methods. Among the more lightweight models, BERT demonstrates leading performance. Additionally, it was observed that increasing model dimensionality and employing summarisation techniques do not consistently enhance clustering efficiency, suggesting that these strategies require careful consideration for practical application. These results highlight a complex balance between the need for refined text representation and computational feasibility in text clustering applications. This study extends traditional text clustering frameworks by integrating embeddings from LLMs, offering improved methodologies and suggesting new avenues for future research in various types of textual analysis.

1. Introduction

Text clustering has attracted considerable interest in text analysis due to its potential to reveal hidden structures in large volumes of unstructured textual data. With the exponential growth of digital text content generated through platforms such as social media, online news outlets, and academic publications, the ability to organise and analyse these data has become increasingly critical. Text clustering can organise large volumes of unstructured data into meaningful categories, facilitating efficient information retrieval and insightful thematic analysis across various domains, such as customer feedback, academic research, and social media content.

Text clustering serves as a preliminary step in various text analysis tasks, including topic modelling, trend analysis, and sentiment analysis. By grouping similar texts, subsequent analyses can proceed with enhanced accuracy and relevance, focusing on more homogeneous data sets with specific characteristics or themes. This process improves the precision of the analyses and ensures that the insights derived are more targeted and applicable to the context in which the data were generated.

As an analytical task, text clustering involves grouping text documents into clusters such that texts within the same cluster are more similar to each other than to those in different clusters. This process relies on the principle that text documents can be mathematically represented as vectors in a high-dimensional space, known as embeddings, where dimensions correspond to various features extracted from the documents, such as word frequency or context. Clustering algorithms use measures of proximity or resemblance to group documents that exhibit close correspondence in the feature space. This approach enables the identification of natural groupings within the data, facilitating more effective organisation and analysis of large text corpora.

The research presented in this article aims to contribute to the domain of text clustering by testing and identifying optimal combinations of embeddings – including those used in recently released large language models (LLMs) – and clustering algorithms that maximise clustering performance across various datasets. The primary objective of this

* Corresponding author.
E-mail addresses: alina.petukhova@ulusofona.pt (A. Petukhova), joao.matos.carvalho@ulusofona.pt (J.P. Matos-Carvalho), nuno.fachada@ulusofona.pt (N. Fachada).

https://doi.org/10.1016/j.ijcce.2024.11.004
Received 22 March 2024; Received in revised form 17 November 2024; Accepted 18 November 2024
Available online 25 November 2024

paper is to determine if embeddings derived from LLMs outperform traditional embedding techniques, such as Term Frequency–Inverse Document Frequency (TF–IDF). Additionally, experiments are conducted to evaluate the impact of model size and dimensionality reduction through summarisation techniques on clustering performance.

Results indicate that LLM embeddings are highly effective at capturing the structured aspects of language, with BERT demonstrating superior performance among lightweight models. Moreover, increasing model dimensionality and employing summarisation techniques do not consistently enhance clustering efficiency, suggesting that these strategies require careful evaluation for practical application. These findings underscore the need to balance detailed text representation with computational feasibility in text clustering tasks.

This paper is organised as follows. In Section 2, advancements in textual embeddings are described, and classical text clustering algorithms used in this domain are briefly mentioned. Section 3 outlines the main steps and components of this study, including dataset selection, data preprocessing, embeddings, and clustering algorithm configurations used to assess clustering quality. Section 4 presents the results of our study and provides a discussion of these findings. Limitations encountered during the study, along with recommendations for overcoming them, are acknowledged in Section 5. Finally, Section 6 synthesises the main conclusions of this research and suggests future developments.

2. Background

2.1. Text embeddings

The field of text representation in natural language processing (NLP) has undergone an impressive transformation over the past few decades. From simple representations to highly sophisticated embeddings, advancements in this domain have significantly improved the ability of machines to process and understand human language with increasing accuracy.

One of the earliest methods of text representation that laid the groundwork for subsequent advances was Term Frequency–Inverse Document Frequency (TF–IDF). This method quantifies the importance of a word within a document relative to a corpus by accounting for term frequency and inverse document frequency (Salton & McGill, 1983). While TF–IDF effectively highlights the relevance of words, it treats each term as independent and fails to capture word context and semantic meaning (Ramos, 2003).

Word embeddings, such as those produced by Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) and GloVe (Pennington, Socher, & Manning, 2014), marked a significant advancement by generating dense vector representations of words based on their contexts. These models leveraged surrounding words and large corpora to learn word relationships, successfully capturing a range of semantic and syntactic similarities. Despite their effectiveness in capturing semantic regularities, these models still provided a single static vector per word, which posed limitations in handling polysemous words—words with multiple meanings.

The arrival of BERT (Bidirectional Encoder Representations from Transformers) initiated a new phase of embedding sophistication (Devlin, Chang, Lee, & Toutanova, 2019). To generate contextual embeddings, BERT employs a bidirectional transformer architecture, pre-trained on a massive corpus. This allows for a deeper understanding of word relationships by considering the full context of a word in a sentence in both directions. BERT revolutionised tasks like text clustering by providing richer semantic representations.

Today, LLMs like OpenAI's GPT are at the forefront of generating state-of-the-art embeddings (Brown et al., 2020). LLMs extend the capabilities of previous models by providing an unprecedented depth and breadth of knowledge encoded in word and sentence-level embeddings. These models are trained on extensive datasets to capture a broad spectrum of human language variations and generate embeddings that reflect a comprehensive understanding of contexts and concepts.

The progression from TF–IDF to sophisticated LLM embeddings represents a significant advancement towards more contextually aware text representation in NLP. This evolution continues to propel the field forward, expanding the possibilities for applications such as text clustering, sentiment analysis, and beyond. In the context of text clustering methodologies, there exists a considerable research gap that underscores the need for a comprehensive evaluation of LLM embeddings against traditional techniques such as TF–IDF.

2.2. Text clustering algorithms

Text clustering involves grouping a set of texts such that texts in the same group (referred to as a cluster) are more similar to each other than to those in different clusters. This section provides an overview of classic clustering algorithms widely used for clustering textual data.

K-means is perhaps the most well-known and commonly used clustering algorithm due to its simplicity and efficiency. It partitions the dataset into k clusters by minimising the within-cluster sum of squares, i.e., variance. Each cluster is represented by the mean of its points, known as the centroid (MacQueen, 1967). K-means is particularly effective for large datasets but depends heavily on the initialisation of centroids and the value of k, which must be known a priori.

Agglomerative hierarchical clustering (AHC) builds nested clusters by merging them successively. This bottom-up approach starts with each text as a separate cluster and combines clusters based on a linkage criterion, such as the minimum or maximum distance between cluster pairs (Ward, 1963). AHC is versatile and allows for discovering hierarchies in data, but it can be computationally expensive for large datasets.

Spectral clustering techniques use the eigenvalues of a similarity matrix to reduce dimensionality before applying a clustering algorithm such as k-means. It is particularly adept at identifying clusters that are not necessarily spherical, as k-means assumes. Spectral clustering can also handle noise and outliers effectively (Ng, Jordan, & Weiss, 2002). However, its computational cost can be high due to the eigenvalue decomposition involved.

Fuzzy c-means (FuzzyCM) is a clustering method that allows data points to belong to more than one cluster. This method minimises the objective function with respect to membership and centroid position, providing soft clustering and handling overlapping clusters (Bezdek, 1981). FuzzyCM is useful when the boundaries between clusters are not clearly defined, although it is computationally more intensive than k-means.

In addition to these classic approaches, recent years have seen the rise of alternative methods for text clustering that leverage the strengths of modern embeddings and consider the unique properties of textual data. These include using deep learning models, particularly those based on autoencoders, to learn meaningful low-dimensional representations ideal for clustering (Berahmand, Daneshfar, Golzari Oskouei, Dorosti, & Aghajani, 2022; Xie, Girshick, & Farhadi, 2016).

Ensemble clustering approaches, where multiple clustering algorithms are combined to improve the robustness and quality of the results, have also been gaining attention. These methods benefit from the diversity of the individual algorithms and can sometimes overcome the limitations of any single method (Strehl & Ghosh, 2002).

The field of text clustering is rapidly expanding, and keeping up with the latest findings is crucial for advancing the state of the art. Recent papers, such as those exploring the use of transformer-based embeddings for clustering (Pugachev & Burtsev, 2021) and integrating external knowledge bases into the clustering process (Zhang, Lertvittayakumjorn, & Guo, 2019), represent just the tip of the iceberg in this area of research. Additionally, very recent work by Keraghel, Morbieu, and Nadif (2024) offers a preliminary discussion on the potential of using LLM embeddings for text clustering. The authors analysed

embeddings from BLOOMZ, Mistral, Llama-2, and OpenAI using five clustering algorithms. This paper extends that work by focusing on different datasets, testing additional LLM embeddings, and comparing results using classical TF–IDF as a baseline. Furthermore, this study experiments with summarisation as a dimensionality reduction technique and evaluates the impact of model size on clustering results.

3. Methods

This research evaluates the effectiveness of various embedding representations in enhancing the performance of text clustering algorithms, aiming to identify the most informative embeddings for clustering tasks. To achieve this, we systematically experimented with multiple datasets, embeddings, and clustering methods, and performed an in-depth analysis of clustering results using different evaluation metrics. The following steps were undertaken during the course of this study:

1. Selection of datasets, ensuring the robustness of our findings across different types of textual data.
2. Preprocessing of datasets, including the removal of miscellaneous characters such as emails and HTML tags.
3. Utilisation of various embedding computations, including LLM-related ones, to retrieve numerical text representations.
4. Application of several clustering algorithms commonly used in text clustering.
5. Comparison of clustering results using different external and internal validation metrics.

The following subsections provide a comprehensive description of each of these steps, further detailing the employed methodologies.

3.1. Datasets

We selected five datasets to cover a variety of text clustering challenges. Table 1 shows these datasets and their characteristics. The CSTR abstracts dataset (Yamagishi, Veaux, & MacDonald, 2019) is a corpus of 299 scientific abstracts from the Centre for Speech Technology Research. The homogeneous and domain-specific nature of the CSTR dataset allowed us to investigate the effectiveness of clustering techniques in discerning fine-grained topic distinctions in scholarly text related to categories such as artificial intelligence, theory, systems, and robotics.

The SyskillWebert dataset (Pazzani, 1999), which includes user ratings for web pages, enabled exploratory clustering to analyse information used in recommendation systems. The 20Newsgroups dataset (Mitchell, 1999) is a famous collection of approximately 19,000 news documents partitioned across 20 different classes. With its broad assortment of topics and noisy, unstructured text, this dataset provided a realistic scenario for evaluating the robustness of clustering algorithms under less-than-ideal conditions.

To complement these, we added the MN-DS dataset (Petukhova & Fachada, 2023), a diverse compilation of multimedia news articles, which provides the opportunity to explore the effectiveness of clustering algorithms in handling multi-categorical data. The dataset is organised hierarchically in two levels; therefore, experiments were performed independently for each level. Additionally, we included the Reuters dataset (Lewis, 1997), a benchmark dataset widely used in text mining and information retrieval research, which contains stories from the Reuters news agency. To enhance the comprehensiveness of our evaluation, we selected documents assigned to single classes, reducing the dataset from 10,369 to 8654 text articles. This allowed us to assess the clustering algorithms' ability to accurately group similar documents according to their predefined categories.

Table 1
Tested datasets and their characteristics. 'Size' represents the number of documents in the dataset, while 'No. classes' is the number of categories.

Dataset	Size	No. classes	Ref.
CSTR	299	4	Yamagishi et al. (2019)
SyskillWebert	333	4	Pazzani (1999)
20Newsgroups dataset	18,846	20	Mitchell (1999)
MN-DS dataset level 1	10,917	17	Petukhova and Fachada (2023)
MN-DS dataset level 2	10,917	109	Petukhova and Fachada (2023)
Reuters	8,654	65	Lewis (1997)

3.2. Preprocessing

The motivation for preprocessing text data is to minimise noise and highlight key patterns, thereby improving the efficiency and accuracy of clustering algorithms (Petukhova & Fachada, 2022).

For all datasets, a series of preprocessing steps were taken to ensure the quality and uniformity of the input data. The initial step involved removing miscellaneous items such as irrelevant metadata, HTML tags, and any extraneous content that might skew the analysis. Continuing with this methodology, we systematically eliminated invalid non-Latin characters from the dataset. These characters could arise from multilingual data sources or artefacts of data collection, such as encoding errors. Given that the employed workflow is optimised for Latin-based languages, retaining such characters could increase the dimensionality of the feature space, thereby undermining clustering performance, both in terms of computational efficiency and interpretability of the results.

3.3. Text embeddings

For textual data representation, we compared classical embeddings (Devlin et al., 2019; Uther et al., 2010) and state-of-the-art LLM embeddings. Traditional TF–IDF vectors served as a baseline, providing a sparse but interpretable representation based on word importance within the analysed dataset. BERT embeddings were extracted from the BERT model, which was trained as a transformer-based bidirectional encoder on BookCorpus (Zhu et al., 2015) and English Wikipedia (Przybyła, Borkowski, & Kaczyński, 2022). These embeddings were utilised to achieve deep contextual understanding, capturing the semantic variations across the corpus.

We also employed "text-embedding-ada-002" (Greene, Sanders, Weng, & Neelakantan, 2023) embeddings, as they demonstrated the best results among OpenAI's embeddings on larger datasets for tasks such as text search, code search, and sentence similarity. Additionally, other LLM embeddings were extracted for Falcon (Almazrouei et al., 2023) and LLaMA-2-chat (Touvron et al., 2023) models, included for their respective advancements in performance and efficiency. Falcon embeddings were trained on a hybrid corpus consisting of both text documents and code, while the LLaMA-2-chat embeddings – built on the foundation of the LLaMA-2 model using an optimised auto-regressive transformer – underwent targeted fine-tuning for dialogue and question-and-answer tasks.

BERT, Falcon, and LLaMA-2-chat embeddings were obtained from Hugging Face's transformers library, a platform providing state-of-the-art models in accessible pipelines (Hugging Face, 2024). Table 2 describes the exact embeddings and configurations used.

3.4. Clustering algorithms

The selected clustering algorithms address the diverse nature of text data, which often contains complex patterns requiring robust methods for effective grouping. Standard k-means clustering was used for its simplicity and efficiency in dealing with large datasets. K-means++ was chosen as an enhanced variant of k-means, with careful initialisation to improve convergence and cluster quality (Arthur, Vassilvitskii, et al., 2007). AHC was utilised for its ability to reveal nested structures within

Table 2
Tested embeddings and their configuration. For TF–IDF, parameters min_df and max_df are used to exclude terms that have a document frequency strictly lower than or higher than the specified thresholds, respectively, during the vocabulary creation process; if real-valued, these parameters represent a proportion of documents; if integer-valued, they represent absolute document counts. In turn, max_features limits the vocabulary to only the top max_features terms, ordered by term frequency across the corpus.

Embeddings	Configuration
TF–IDF	min_df = 5, max_df = 0.95, max_features = 8000
BERT	huggingface.co/sentence-transformers/all-mpnet-base-v2
OpenAI	huggingface.co/Xenova/text-embedding-ada-002
Falcon	huggingface.co/tiiuae/falcon-7b
LLaMA-2	huggingface.co/meta-llama/Llama-2-7b-chat-hf

Table 3
Tested clustering algorithms and respective parameters. For k-means and k-means++, init refers to the method used for the initial selection of cluster centroids, n_{init} specifies the number of times the k-means algorithm is run with different centroid seeds, while seed defines the random number for centroid initialisation. For AHC, metric defines the metric used to compute the linkage, while linkage specifies the linkage criterion that determines the distance measurement used between sets of observations. For FuzzyCM, init is the initial fuzzy c-partitioned matrix–if set to $None$, the matrix will be randomly initialised–m is the degree of fuzziness, error defines the stopping criterion, and maxiter specifies the maximum number of iterations allowed. For Spectral clustering, assign_labels determines the strategy used to assign labels in the embedding space and seed is a pseudo-random number seed for initialising the locally optimal block preconditioned conjugate gradient eigenvectors decomposition. For all algorithms, the number of clusters was set to match the number of classes present in the dataset.

Algorithm	Parameters
k-means	init = random, n_{init} = 10, seed = 0
k-means++	init = k-means++, n_{init} = 1, seed = 0
AHC	metric = euclidean, linkage = ward
FuzzyCM	init = None, m = 2, error = 0.005, maxiter = 1000
Spectral	assign_labels = discretise, seed = 10

the data. Compared to k-means, which assigns each data point to a single cluster, FuzzyCM provides a probabilistic membership approach, accommodating polysemy and subtle semantic differences typical in text data. Finally, Spectral clustering was selected for its effectiveness in identifying clusters based on their data-induced graph structure, and it is particularly adept at discovering clusters with non-convex shapes.

To map the newly formed clusters to the original labels of the dataset, we calculated the Euclidean distance between the centroid of each derived cluster and the centroid of each ground truth cluster, assigning the closest ones. The Euclidean distance is a preferable metric for this process because it accurately reflects distances in multidimensional space, ensuring precise mapping. This method allows the use of external evaluation metrics such as the $F1$-score. In our work, we did not apply Euclidean distance at the word level, which can be problematic. Instead, the process computes an aggregation for each identified cluster, enabling a more reliable association with the existing labels.

The selected algorithms and their respective parameters are listed in Table 3. The *scikit-learn* library (Pedregosa et al., 2011) provided the implementations for all algorithms except for FuzzyCM, which was utilised from the *scikit-fuzzy* package (Warner et al., 2019).

3.5. Evaluation metrics

To comprehensively evaluate the quality of different embeddings and algorithm combinations, we used a diverse set of metrics. For external validation, since the original labels were available, we used the weighted $F1$-score (F1S) (Chinchor, 1992), the Adjusted Rand Index (ARI) (Steinley, 2004), and the Homogeneity score (HS) (Rosenberg & Hirschberg, 2007). The $F1$-score was computed to balance precision and recall in the presence of class imbalance. ARI was used to assess clustering outcomes while correcting for chance grouping, and HS was used to evaluate the degree to which each cluster is composed of data points primarily from one class. For internal validation, we employed

Table 4
Metrics used to assess clustering results, their type (external or internal), and their respective formulas. For the $F1$-score (F1S), C represents the number of classes, w_i is the weight assigned to the ith class, which is typically the proportion of that class within the dataset, and $F_{1,i}$ represents the $F1$-score computed for the ith class. For the Adjusted Rand Index (ARI), RI stands for Rand Index, $Expected\_RI$ refers to the expected value of the Rand Index under random label assignment (calculated using the contingency table marginals), and Max_RI is the maximum possible value of the Rand Index. For the Homogeneity Score (HS), $H(C|K)$ is the conditional entropy of the class distribution given the predicted cluster assignments, and $H(C)$ is the entropy of the class distribution. For the Silhouette Score (SS), N represents the total number of data points in the dataset, and $s(i)$ is the silhouette score for a single data point i, defined as $s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$, where $a(i)$ is the average distance from the ith data point to other points in the same cluster, and $b(i)$ is the minimum mean distance from the ith data point to points in a different cluster, minimised over all clusters. For the Calinski–Harabasz Index (CHI), $Tr(B_k)$ is the trace of the between-group dispersion matrix that measures the between-cluster dispersion, $Tr(W_k)$ is the trace of the within-cluster dispersion matrix, which quantifies the within-cluster dispersion, N refers to the number of data points, and k indicates the number of clusters. The optimal value for all these metrics is the maximum.

Metric	Type	Formula
F1S	External	$\sum_{i=1}^{C} w_i F_{1,i}$
ARI	External	$\frac{RI - Expected\_RI}{Max\_RI - Expected\_RI}$
HS	External	$1 - \frac{H(C\|K)}{H(C)}$
SS	Internal	$\frac{1}{N}\sum_{i=1}^{N} s(i)$
CHI	Internal	$\frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}$

the Silhouette Score (SS) (Rousseeuw, 1987) and the Calinski–Harabasz Index (CHI) (Caliński & JA, 1974), evaluating cluster coherence and separation without requiring ground truth. This multifaceted approach ensures a robust assessment, capturing both the alignment with known labels and the intrinsic structure of the generated clusters. These metrics collectively provide a balanced view of performance, accounting for datasets with varying characteristics and sizes. The metrics and their corresponding formulas are presented in Table 4.

3.6. Additional experiments

This section describes additional experiments in which we performed text summarisation (3.6.1) and tested LLM embeddings obtained from larger models (i.e., models with more parameters) prior to clustering (3.6.2). The purpose of these experiments is to investigate whether such representations can improve the discriminability of features within text clusters.

3.6.1. Summarisation

This experiment aims to evaluate summarisation as a tool for dimensionality reduction in text clustering by creating compact representations of the texts that encapsulate their semantic core without losing context. These experiments are hypothesised to streamline the clustering process, potentially leading to more coherent and interpretable clusters, even in large and complex datasets. This involves adding a summarisation step after preprocessing and before the retrieval of embeddings. The models used in summarisation are described in Table 5. As an alternative approach to LLM-based models, we used the BERT-large-uncased summarisation model (Devlin et al., 2019) implemented by the BERT summariser (Miller, 2019) to assess potential improvements in clustering achieved by utilising a lower-dimensionality model.

For the LLaMA-2 and Falcon models, we used the Hugging Face transformers library with the parameters described in Table 6.

The following zero-shot prompt was used for generating the summarised text with LLMs:

Table 5
Overview of text summarisation models used in this study, each employing a transformer-based architecture. 'Max tokens' is the maximum input sequence length. BERT is a bidirectional encoder-only model trained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables, and headers); OpenAI is a GPT-3-based decoder-only model built with multi-head attention blocks, trained on an extended dataset with reinforcement learning from human feedback; Falcon is a causal decoder-only model with a multi-query attention mechanism trained on 1,500B tokens of RefinedWeb enhanced with curated corpora; LLaMA-2-chat is a LLaMA-2-based auto-regressive decoder-only model trained on a 1,000B token dataset collected by Meta, enriched with supervised fine-tuning, reinforcement learning from human feedback, and the Ghost Attention mechanism.

Embeddings	Summarisation model	Max tokens
BERT	bert-large-uncased (Devlin et al., 2019)	512
OpenAI	gpt-3.5-turbo (gpt, 2024)	4096
Falcon	falcon-7b (fal, 2024; Almazrouei et al., 2023)	2048
LLaMA-2-chat	Llama-2-7b-chat-hf (Lla, 2024; Touvron et al., 2023)	4096

Table 6
Parameters used for summarisation with LLaMA-2 and Falcon models: temperature represents the value to modulate probabilities of the next token, max_length defines the maximum length of the sequence to be generated, do_sample is a parameter that determines whether to use sampling rather than greedy decoding, top_k restricts the selection to the top k tokens with the highest probabilities during top-k filtering, and num_return_sequences is the number of independently computed returned sequences for each element in the batch.

Parameter name	Value
temperature	0
max_length	800
do_sample	True
top_k	10
num_return_sequences	1

Write a concise summary of the text. Return your responses with maximum 5 sentences that cover the key points of the text. {text}

SUMMARY:

3.6.2. Increasing model dimension

The original publications on LLMs underline the performance increase with larger model sizes in tasks such as common sense reasoning, question answering, and code tasks (Almazrouei et al., 2023; Touvron et al., 2023). This experiment evaluates embeddings from various LLM sizes to analyse the impact of higher-dimensional models on the performance of clustering algorithms, aiming to determine if they enhance cluster cohesion and separation. For this purpose, we used embeddings obtained from models presented in Table 7.

To visualise the different embeddings and capture their intrinsic structures, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbour Embedding (t-SNE) (Van der Maaten & Hinton, 2008) were employed. Initially, PCA was applied for preliminary dimensionality reduction while preserving variance. Subsequently, t-SNE was used to project the data into a lower-dimensional space, emphasising local disparities between embeddings. This sequential application of PCA and t-SNE allows us to capture both global and local structures within the embeddings, providing a richer visualisation than using PCA alone.

4. Results and discussion

Table 8 presents the clustering metrics for the "best" algorithm for the tested combinations of dataset, embedding, and clustering algorithm. By "best", we mean the algorithm with the highest $F1$-score value. The complete results are available as supplementary material.[1]

[1] https://doi.org/10.5281/zenodo.10844657

Table 7
Embeddings and corresponding models used in the dimensionality experiments. Here, 'Size' represents the number of parameters denoted in billions (bp), and 'Tokens' refers to the embedding token size in trillions (trl).

Model	Model reference	Size (bp)	Tokens (trl)
Falcon-7b	huggingface.co/tiiuae/falcon-7b	7	1.5
Falcon-40b	huggingface.co/tiiuae/falcon-40b	40	1
LLaMA-2-7b	huggingface.co/meta-llama/Llama-2-7b-chat-hf	7	2
LLaMA-2-13b	huggingface.co/meta-llama/Llama-2-13b-chat-hf	13	2

Results demonstrate that OpenAI embeddings generally yield superior clustering performance on structured, formal texts compared to other methods based on most metrics (column 'Total' in Table 8). The combination of the k-means algorithm and OpenAI's embeddings yielded the highest values of ARI, $F1$-score, and HS in most experiments. This may be attributed to OpenAI's embeddings being trained on a diverse array of Internet text, rendering them highly effective at capturing the diversity of language structures.

Low values of SS and CHI for the same algorithm could indicate that, while clusters are homogeneous and aligned closely with the ground truth labels (suggesting good class separation and cluster purity), they may not be well-separated or compact as evaluated in a geometric space. This discrepancy can arise in cases where data has a high-dimensional or complex structure that external measures such as ARI, F1S, and HS capture effectively. However, when projecting into a lower-dimensional space for SS and CHI, the clusters appear to overlap or vary widely in size, leading to a lower score for these spatial coherence metrics. This highlights a challenge in text clustering: achieving both high cluster purity alongside tight cluster geometry.

Interestingly, for the case of the Reuters dataset (DS6), we observed that FuzzyCM consistently outperformed other clustering algorithms when paired with different embeddings. The potential reason behind this result may be related to the flexibility of fuzzy clustering, which allows data points to belong to multiple clusters with varying degrees of membership, making it particularly suitable for datasets with a large number of classes such as the Reuters dataset.

In the domain of open-source models, namely Falcon, LLaMA-2, and BERT, the latter emerged as the frontrunner. Given that BERT is designed to understand context and potentially due to the model's lower dimensionality, these embeddings demonstrate good effectiveness in text clustering. In the comparative analysis of open-source LLM embeddings, Falcon-7b outperformed LLaMA-2-7b across most datasets, demonstrating improved cluster quality and distinctiveness. This superiority may be attributed to Falcon-7b embeddings' ability to better capture salient linguistic features and semantic relationships within the texts since these embeddings were trained on a mixed corpus of text and code, as opposed to the LLaMA-2 embeddings, which are specialised for dialogues and Q&A contexts. Additionally, the CHI metric – measuring the dispersion ratio between and within clusters – is higher for Falcon-7b embeddings, suggesting that clusters are well-separated and dense.

Experiments with the MN-DS dataset, featuring a hierarchical label structure, indicate that clustering at a higher, more abstract label level (17 classes) produces better class separation. However, the $F1$-score is higher when clustering is assessed at the more specific level (109 classes). This likely indicates better precision and recall for individual classes, while lower values for other metrics reflect the increased difficulty in maintaining overall clustering quality and cohesion with a higher number of classes. These results indicate that clustering at higher levels can produce more cohesive and interpretable clusters that align with natural categorical divisions, though at the expense of extracting less specific information for each document.

Table 8
Results of text clustering for the best-performing clustering algorithms for each combination of dataset and embedding. The best algorithm was determined by choosing the algorithm with the highest $F1$-score value. DS1 represents the CSTR dataset, DS2 is the SyskillWebert dataset, DS3 is the 20newsgroups dataset, DS4 is the MN-DS dataset for level 1 labels, DS5 is the MN-DS dataset for level 2 labels, and DS6 is the Reuters dataset. 'Total' represents the number of metrics per given embeddings/algorithm combination that outperform other combinations.

Dataset	Embed.	Best Alg.	F1S	ARI	HS	SS	CHI	Total
DS1	TF–IDF	*k*-means	0.67	0.38	0.46	0.016	4	0/5
	BERT	Spectral	**0.85**	**0.60**	0.63	**0.118**	25	3/5
	OpenAI	*k*-means	0.84	0.59	**0.64**	0.066	13	1/5
	LLaMA-2	*k*-means	0.41	0.09	0.17	0.112	**49**	1/5
	Falcon	*k*-means	0.74	0.39	0.48	0.111	34	0/5
DS2	TF–IDF	Spectral	0.82	0.63	0.58	0.028	8	0/5
	BERT	AHC	0.74	0.58	0.53	0.152	37	0/5
	OpenAI	AHC	**0.90**	**0.79**	**0.75**	0.070	19	3/5
	LLaMA-2	*k*-means	0.51	0.21	0.25	0.137	69	0/5
	Falcon	*k*-means++	0.45	0.26	0.30	**0.170**	**85**	2/5
DS3	TF–IDF	Spectral	0.35	0.13	0.28	−0.002	37	0/5
	BERT	*k*-means	0.43	0.25	0.44	0.048	412	0/5
	OpenAI	*k*-means	**0.69**	**0.52**	**0.66**	0.035	213	3/5
	LLaMA-2	AHC	0.17	0.11	0.26	0.025	264	0/5
	Falcon	*k*-means	0.26	0.15	0.30	**0.071**	**1120**	2/5
DS4	TF–IDF	*k*-means	0.29	0.13	0.48	0.034	17	0/5
	BERT	*k*-means	0.35	0.24	0.55	**0.072**	61	1/5
	OpenAI	*k*-means	**0.38**	**0.26**	**0.58**	0.053	42	3/5
	LLaMA-2	*k*-means	0.21	0.11	0.40	0.053	88	0/5
	Falcon	*k*-means++	0.27	0.16	0.48	0.071	**92**	1/5
DS5	TF–IDF	AHC	0.31	0.09	0.29	0.010	37	0/5
	BERT	*k*-means++	0.43	**0.27**	**0.42**	0.060	178	2/5
	OpenAI	Spectral	**0.45**	0.25	0.41	0.036	120	1/5
	LLaMA-2	AHC	0.23	0.10	0.23	0.031	263	0/5
	Falcon	*k*-means++	0.28	0.12	0.25	**0.070**	**359**	2/5
DS6	TF–IDF	FuzzyCM	0.51	0.19	0.20	0.01	74	0/5
	BERT	Spectral	0.51	**0.32**	0.35	0.02	37	1/5
	OpenAI	FuzzyCM	**0.52**	0.23	0.21	**0.10**	**1095**	3/5
	LLaMA-2	k-means++	0.19	0.08	**0.63**	0.07	518	1/5
	Falcon	FuzzyCM	0.22	−0.03	0.21	0.00	930	0/5

Table 9
Results of summarisation effect on text clustering for the best-performing clustering algorithms in comparison to clustering without summarisation. The 'DS' column indicates the dataset being analysed; in particular, DS1 represents the CSTR dataset, while DS2 denotes the SyskillWebert dataset.

DS	Embed.	Version	Best Alg.	F1S	ARI	HS	SS	CHI
DS1	BERT	Full	Spectral	**0.85**	**0.60**	**0.63**	**0.118**	**25**
		Summary	Spectral	0.81	0.50	0.56	0.114	24
	OpenAI	Full	*k*-means	**0.84**	**0.59**	**0.64**	**0.066**	13
		Summary	*k*-means	0.81	0.53	0.58	0.061	13
	LLaMA-2	Full	*k*-means	0.44	0.12	0.21	**0.099**	**53**
		Summary	AHC	**0.47**	**0.16**	**0.30**	0.072	24
	Falcon	Full	*k*-means	**0.74**	**0.39**	**0.48**	0.111	34
		Summary	*k*-means	0.40	0.03	0.02	**0.224**	**329**
DS2	BERT	Full	AHC	0.74	**0.58**	0.53	**0.152**	**37**
		Summary	AHC	**0.75**	0.57	**0.54**	0.089	22
	OpenAI	Full	AHC	**0.90**	**0.79**	**0.75**	**0.070**	**19**
		Summary	Spectral	0.79	0.71	0.64	0.054	18
	LLaMA-2	Full	*k*-means	**0.51**	**0.21**	**0.25**	0.137	69
		Summary	FuzzyCM	0.25	0.04	0.06	**0.548**	**603**
	Falcon	Full	*k*-means++	**0.45**	**0.26**	**0.30**	0.170	85
		Summary	FuzzyCM	0.34	0.04	0.07	**0.269**	**577**

Table 10
Results for the model size experiment, displaying the best performing clustering algorithm comparing two sizes of the LLaMA and Falcon models on the DS1 (CSTR) and DS2 (SyskillWebert) datasets. Model sizes are given in billions of parameters.

Dataset	Embed.	Best Alg.	F1S	ARI	HS	SS	CHI
DS1	LLaMA-2-7b	*k*-means	0.41	0.09	0.17	**0.112**	**50**
	LLaMA-2-13b	AHC	**0.82**	**0.53**	**0.61**	0.084	21
	Falcon-7b	*k*-means	**0.74**	**0.39**	**0.48**	0.111	34
	Falcon-40b	AHC	0.46	0.17	0.30	0.111	**44**
DS2	LLaMA-2-7b	*k*-means	0.51	0.21	0.25	**0.137**	**69**
	LLaMA-2-13b	*k*-means++	**0.60**	**0.49**	**0.39**	0.095	37
	Falcon-7b	*k*-means++	**0.45**	**0.26**	**0.30**	0.170	85
	Falcon-40b	*k*-means++	0.40	0.15	0.13	**0.188**	**131**

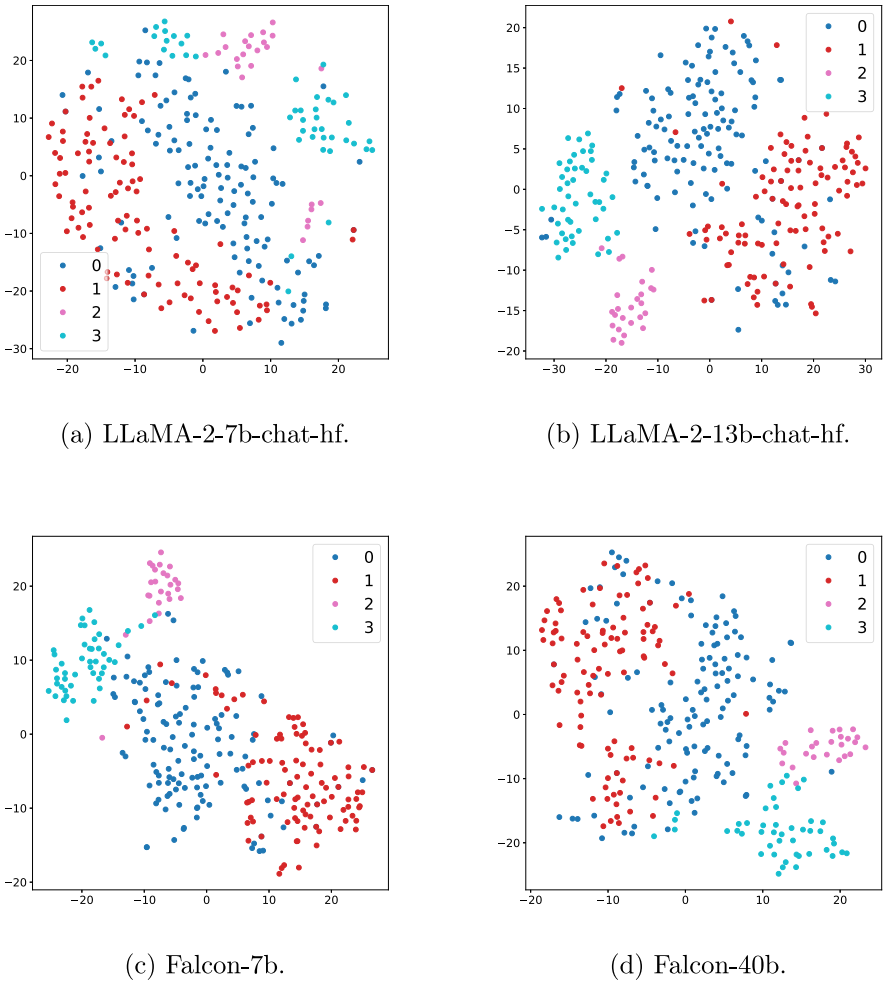
Results of the summarisation experiment, depicted in Table 9, show that using summarisation as a dimensionality reduction technique does not consistently benefit all models. Clustering results for the original texts without generated summaries are generally higher than those with summarisation. This finding suggests that essential details necessary for accurate clustering might have been lost during the summarisation process. Alternatively, the inherent complexity and variations of textual representation might require a more sophisticated approach to text summarisation that can maintain essential information while reducing complexity. Additionally, it is important to highlight that we observed low-quality clustering results when using the summarisation output from the smaller-sized LLaMA-2-7b and Falcon-7b models, likely due to their limited ability to capture and reproduce the complexity and subtle gradations in the source texts.

Results for the model size experiment, presented in Table 10, highlight the negative influence of the number of parameters on clustering outcomes for Falcon, given that the larger model, Falcon-40b, produced lower ARI, $F1$-score, and HS for the CSTR and SyskillWebert datasets. This may be because embeddings for the Falcon-40b model were created over a subset of the data used for the Falcon-7b embeddings (1.5 trillion tokens for Falcon-7b and 1 trillion tokens for Falcon-40b). In the case of LLaMA-2, models with sizes 7b and 13b were reportedly trained on identical datasets (Lla, 2024). Results indicate that embeddings from the larger model, LLaMA-2-13b, outperform those from LLaMA-2-7b. This outcome is anticipated, as larger models with more parameters generally have a greater capacity to capture complex patterns and relationships in the data, leading to richer and more expressive embeddings.

Another perspective is given by Fig. 1, which shows a visualisation of different LLM parameter sizes for the CSTR dataset using PCA and *t*-SNE. A noticeable artefact for LLaMA-2-7b (Fig. 1(a)) and Falcon-40b (Fig. 1(d)) is the lack of coherence in the four document classes, indicating an unclear delimitation in the feature space. This observation is consistent with the results in Table 10, where these two models have the lowest F1S, ARI, and HS values. Conversely, when employing LLaMA-2-13b (Fig. 1(b)) and Falcon-7b (Fig. 1(c)), the classes exhibit better separation, which aligns with the clustering results observed in Table 10, where these two models achieved the highest values in 3 out of the 5 metrics calculated. Increasing the number of tokens in Falcon yielded better results, aligning with existing literature that suggests models trained on larger and more diverse corpora are more capable of capturing complex patterns in the data (Naveed et al., 2023).

Balancing computational requirements with clustering quality involves weighing the benefits of larger embeddings against their resource demands. As results show, a larger number of parameters do not necessarily lead to improved clustering and require significantly more computational power and memory. Empirical evaluation is essential to determine if the improvements justify the additional costs, considering the specific task and resource constraints.

In summary, these results provide valuable insights into the relationship between text embeddings, clustering algorithms, and dimensionality reduction techniques. We evaluated five embeddings and four clustering algorithms across five datasets, including the MN-DS dataset with hierarchical labels. Embeddings from the OpenAI model generally outperformed traditional techniques such as BERT and TF–IDF, aligning with previous studies. Conversely, LLaMA-2 and Falcon models produced inferior results compared to BERT across most metrics. The *k*-means algorithm was the most effective on most datasets, while FuzzyCM performed better on the Reuters dataset. Additionally, summarisation models used for dimensionality reduction did not



(a) LLaMA-2-7b-chat-hf. (b) LLaMA-2-13b-chat-hf.

(c) Falcon-7b. (d) Falcon-40b.

Fig. 1. Representation of different embeddings for the CSTR dataset, where PCA was used as a preliminary dimensionality reduction algorithm and t-SNE for data projection into a lower-dimensional space.

enhance clustering performance, and the use of higher-dimensional models displayed mixed results.

5. Limitations

Despite the insights revealed in this study, the available computational resources constrained the experiments. This limitation prohibited additional trials with summary generation, higher-dimensional models on more voluminous datasets, and fine-tuning of model parameters. Consequently, the potential benefits of summaries in large-scale text clustering have yet to be thoroughly evaluated. Summarisation might behave differently when applied at scale or with distinct prompts, potentially providing more pronounced benefits or drawbacks depending on the complexity and diversity of the text data.

Moreover, embeddings computed for extra-large models like Falcon-180b or LLaMA-2-70b could potentially deliver substantial gains. However, practical application is restrained by the significant computational demands associated with larger model sizes. This limitation potentially skews the understanding of the absolute efficacy of the embeddings computed for these models, as performance improvements could only be inferred up to a specific size.

Testing the selected clustering algorithms with a wider range of parameters could also provide additional insights into the results. However, due to the aforementioned computational limitations, a breadth-first approach was chosen – testing several different clustering algorithms – rather than a depth-first approach, which would involve experimenting with more parameters for each algorithm. For example, in AHC, the analysis was limited to using Euclidean distance with Ward linkage. Ward linkage was chosen because of its ability to minimise the variance within clusters, resulting in more compact and cohesive clusters, which is particularly beneficial for text data clustering. While this approach provides a comprehensive comparison across different methods, future research could explore parameter optimisation in greater depth to further refine clustering outcomes.

These constraints prompt essential considerations for future research. Specifically, experiments with larger datasets and higher-dimension models would enable a more comprehensive and accurate understanding of the potentials and limitations of text clustering algorithms and their scalability in real-world applications.

6. Conclusions

In this study, we examined the impact that various embeddings – namely TF–IDF, BERT, OpenAI, LLaMA-2, and Falcon – and clustering algorithms have on grouping textual data. Through detailed exploration, we evaluated the efficacy of dimensionality reduction via summarisation and the role of model size on the clustering efficiency of various datasets. We found that OpenAI's embeddings generally outperform other embeddings, with BERT's performance excelling among open-source alternatives, underscoring the potential of advanced models to positively influence text clustering results.

A key finding from the experiments with summarisation is that summary-based dimensionality reduction does not consistently improve

clustering performance. This indicates that careful consideration is required when preprocessing text to avoid losing essential information.

This research highlights the trade-off between improved clustering performance and the computational costs of using embeddings from larger models. Although results indicate that an increase in model size may yield superior clustering performance, potential benefits must be weighed against the practicality of available computing resources.

These findings point towards continued research focused on developing strategies that leverage the strengths of advanced models while mitigating their computational demands. It is also critical to expand the scope of research to include more diverse text types, which will provide a more comprehensive understanding of clustering dynamics across different contexts. Analysing embeddings of very recent or yet-to-be-released models will also be important. Additionally, conducting parameter searches for clustering algorithms is essential to optimise their performance and ensure robust clustering outcomes across various datasets.

In conclusion, our findings highlight the relationship between embedding types, dimensionality reduction, model size, and text clustering effectiveness in the context of structured, formal texts. While more advanced embeddings such as those from OpenAI offer clear advantages, researchers and practitioners must weigh the trade-offs regarding cost, computational resources, and the effects of text preprocessing techniques.

CRediT authorship contribution statement

Alina Petukhova: Writing – review & editing, Writing – original draft, Visualization, Investigation, Formal analysis, Conceptualization. **João P. Matos-Carvalho:** Writing – review & editing, Supervision, Formal analysis, Data curation. **Nuno Fachada:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was financed by the Fundação para a Ciência e a Tecnologia, Portugal, in the framework of projects UIDB/04111/2020, UIDB/00066/2020, CEECINST/00002/2021/CP2788/CT0001 and CEECINST/00147/2018/CP1498/CT0015, as well as by the Instituto Lusófono de Investigação e Desenvolvimento (ILIND), Portugal under project COFAC/ILIND/COPELABS/1/2022.

References

Falcon-7b. (2024). https://huggingface.co/tiiuae/falcon-7b/ (Online; Accessed 28 May 2024).

Gpt-3-5-turbo. (2024). https://platform.openai.com/docs/models/gpt-3-5-turbo/ (Online; Accessed 28 May 2024).

Llama-2-7b-chat-hf. (2024). https://huggingface.co/meta-llama/Llama-2-7b-chat-hf/ (Online; Accessed 28 May 2024).

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., et al. (2023). The falcon series of open language models. arXiv:2311.16867.

Arthur, D., Vassilvitskii, S., et al. (2007). K-means++: The advantages of careful seeding. *7*, In *Proceedings of the 18th annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). SIAM.

Berahmand, K., Daneshfar, F., Golzari Oskouei, A., Dorosti, M., & Aghajani, M. (2022). An improved deep text clustering via local manifold of an autoencoder embedding. *SSRN Electronic Journal*, http://dx.doi.org/10.2139/ssrn.4295242.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Caliński, T., & JA, H. (1974). A dendrite method for cluster analysis. *Communications in Statistics. Theory and Methods*, *3*, 1–27. http://dx.doi.org/10.1080/03610927408827101.

Chinchor, N. (1992). MUC-4 evaluation metrics. In *MUC4 '92*, *Proceedings of the 4th conference on message understanding* (pp. 22–29). USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/1072064.1072067.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding.. In J. Burstein, C. Doran, & T. Solorio (Eds.), *NAACL-HLT (1)* (pp. 4171–4186). Association for Computational Linguistics.

Greene, R., Sanders, T., Weng, L., & Neelakantan, A. (2023). New and improved embedding model. https://openai.com/blog/new-and-improved-embedding-model (Online; Accessed 28 May 2024).

Hugging Face (2024). https://huggingface.co/ (Online; Accessed 28 May 2024).

Keraghel, I., Morbieu, S., & Nadif, M. (2024). Beyond words: A comparative analysis of LLM embeddings for effective clustering. (pp. 205–216). Springer Nature Switzerland, http://dx.doi.org/10.1007/978-3-031-58547-0_17.

Lewis, D. (1997). Reuters-21578 text categorization collection. http://dx.doi.org/10.24432/C52G6M, Online; (Accessed 28 May 2024).

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using *t*-SNE.. *Journal of Machine Learning Research*, *9*(11).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability, vol. 1* (pp. 281–297). Berkeley, Calif: University of California Press.

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *International conference on learning representations* (pp. 1–12). URL https://api.semanticscholar.org/CorpusID:5959482.

Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. arXiv:1906.04165.

Mitchell, T. (1999). Twenty newsgroups. http://dx.doi.org/10.24432/C5C323, Online; (Accessed 28 May 2024).

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., et al. (2023). A comprehensive overview of large language models. ArXiv, arXiv:2307.06435, URL https://api.semanticscholar.org/CorpusID:259847443.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems, vol. 2* (pp. 849–856). Cambridge, MA, USA: MIT Press.

Pazzani, M. (1999). SyskillWebert Web Page Ratings. https://kdd.ics.uci.edu/databases/SyskillWebert/ (Online; Accessed 28 May 2024).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing, vol. 14* (pp. 1532–1543). Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1162.

Petukhova, A., & Fachada, N. (2022). TextCL: A python package for NLP preprocessing tasks. *SoftwareX*, *19*, Article 101122. http://dx.doi.org/10.1016/j.softx.2022.101122, Online; accessed 28 May 2024.

Petukhova, A., & Fachada, N. (2023). MN-DS: A multilabeled news dataset for news articles hierarchical classification. *Data*, *8*(5), http://dx.doi.org/10.3390/data8050074.

Przybyła, P., Borkowski, P., & Kaczyński, K. (2022). Wikipedia complete citation corpus. http://dx.doi.org/10.5281/zenodo.6539054, Online; (Accessed 28 May 2024).

Pugachev, L., & Burtsev, M. (2021). Short text clustering with transformers. In *International conference in computational linguistics and intelligent technologies* (pp. 571–577). CEUR-WS.org, http://dx.doi.org/10.28995/2075-7182-2021-20-571-577.

Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (pp. 29–48).

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. In J. Eisner (Ed.), *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-coNLL)* (pp. 410–420). Prague, Czech Republic: Association for Computational Linguistics, URL https://aclanthology.org/D07-1043.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

Steinley, D. (2004). Properties of the hubert-arabie adjusted rand index. *Psychological methods*, *9*, 386–396. http://dx.doi.org/10.1037/1082-989X.9.3.386.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*, 583–617.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. http://dx.doi.org/10.48550/arXiv.2307.09288, arXiv:2307.09288.

Uther, W., Mladenić, D., Ciaramita, M., Berendt, B., Koł cz, A., Grobelnik, M., et al. (2010). *TF–IDF* (pp. 986–987). Springer US, http://dx.doi.org/10.1007/978-0-387-30164-8_832.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.

Warner, J., Sexauer, J., Scikit-fuzzy, Twmeggs, Alexsavio, Unnikrishnan, A., et al. (2019). JDWarner/scikit-fuzzy: Scikit-fuzzy version 0.4.2. http://dx.doi.org/10.5281/zenodo.3541386, (Online; Accessed 28 May 2024).

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478–487). JMLR.org.

Yamagishi, J., Veaux, C., & MacDonald, K. (2019). CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92). http://dx.doi.org/10.7488/ds/2645, Online; (Accessed 28 May 2024).

Zhang, J., Lertvittayakumjorn, P., & Guo, Y. (2019). Integrating semantic knowledge to tackle zero-shot text classification. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 1031–1040). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1108.

Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., et al. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE international conference on computer vision*, 19–27. http://dx.doi.org/10.1109/ICCV.2015.11.