Large Language Model Enhanced Clustering for News Event Detection

Adane Nega Tarekegn, Fazle Rabbi, Bjørnar Tessem
Department of Information Science and Media Studies, University of Bergen, Norway
(email: adane.tarekegn@uib.no, fazle.rabbi@uib.no, bjornar.tessem@uib.no)

***Abstract*— The news landscape is continuously evolving, with an ever-increasing volume of information from around the world. Automated event detection within this vast data repository is crucial for monitoring, identifying, and categorizing significant news occurrences across diverse platforms. This paper presents an event detection framework that leverages Large Language Models (LLMs) combined with clustering analysis to detect news events from the Global Database of Events, Language, and Tone (GDELT). The framework enhances event clustering through both pre-event detection tasks (keyword extraction and text embedding) and post-event detection tasks (event summarization and topic labeling). We also evaluate the impact of various textual embeddings on the quality of clustering outcomes, ensuring robust news categorization. Additionally, we introduce a novel Cluster Stability Assessment Index (CSAI) to assess the validity and robustness of clustering results. CSAI utilizes latent feature vectors to provide a new way of measuring clustering quality. Our experiments indicate that combining LLM embeddings with clustering algorithms yields the best results, demonstrating greater robustness in terms of CSAI scores. Moreover, post-event detection tasks generate meaningful insights, facilitating effective interpretation of event clustering results. Overall, our findings indicate that the proposed framework offers valuable insights and could enhance the accuracy and depth of news reporting.**

Keywords-Cluster Analysis; LLM; Cluster Validation; News Events; Embeddings.

I. INTRODUCTION

News events are ubiquitous and dynamic, significantly shaping social, political, economic, and various other aspects of public life [1]. Over the past few years, the swift evolution of the internet has significantly altered how news is shared worldwide, resulting in vast volumes of news data being generated daily. News media discourse is typically centred around real-world events that catch media attention and give rise to news report streams. The process of detecting such events from a huge amount of web data involves identifying and categorizing related sets of significant events that occur in real life [2][3][4]. Automatic event-based organization of news data can lead to better structuring and classification of textual news from a variety of online news media sources, and thus provide users with a better online news experience.

In this study, we propose a clustering-based framework for global news event detection from the GDELT(Global Database of Events, Language, and Tone) [5] project news database by leveraging the recent advances in large language models (LLMs) alongside a novel cluster validation approach. In recent years, LLMs have been used as a new method for improving and guiding the clustering process by identifying subtle semantic representations [6][7]. To automatically identify and generate more cohesive event clusters, we chose to employ LLMs at three essential stages of the clustering process: keyword extraction, representation of those keywords as embeddings, and interpreting the cluster assignments. First, the traditional KeyBERT approach [8] is used to generate keywords and then OpenAI's GPT model is applied to refine and optimize the keyword extraction process. For the text embedding task, we utilize the LLM embedding model (*text-embedding-ada-002*) [9], a specialized variant of the GPT-3 architecture developed by OpenAI specifically designed for generating text embeddings. For a better understanding of the final clustering results and to characterize each candidate event cluster, we employed a few-shot text summarization technique using *GPT-3.5-turbo-instruct*, which was applied to a set of articles allocated to each cluster. In addition, each event cluster is semantically assigned to different news topics with the help of the International Press Telecommunications Council (IPTC) taxonomy [10].

Evaluation and understanding of the quality of event detection results is one of the key issues in cluster analysis [11]. In this paper, we used stability-based evaluation, which involves assessing the consistency of clustering derived from applying the same clustering algorithm to several independent and identically distributed samples [12]. This evaluation method involves dividing the data into training and testing data points, where the training sets are used for cluster construction, enabling the prediction of cluster memberships for the test data points. In this process, clustering stability is determined by assessing the correlation or similarity between the features of the training and testing data within each cluster. In general, our framework can improve the accuracy and depth of news reporting while maintaining journalistic integrity and ethical standards. The key contributions are summarized as follows:

- We have curated a large dataset involving main steps, such as news aggregation, cleaning, and pre-processing.
- We propose a framework for detecting events from GDELT using embeddings and clustering algorithms.
- We incorporate LLMs in the framework to enhance the clustering process through keyword extraction, text embedding, summarization, and cluster interpretability.
- We introduce a novel stability-based cluster validation index to measure the quality of clustering results using the similarity scores of input feature vectors.

II. RELATED WORKS

Event detection is one of the fundamental tasks for event extraction. The aim is to find documents that contain a particular event of interest from a large collection of texts or to obtain a set of clusters within a collection of texts, with each cluster comprising articles that discuss the same event [2][13]. The task of event detection can be traced back to 1998 when a collaborative effort aimed to define the problem within the broader field of Topic Detection and Tracking (TDT) [14]. Since then, a significant range of algorithms has been developed to address the problem across various domains, including social media [15], by leveraging the remarkable advancements in text mining and natural language processing. The frequent changes and constant flux of news data streams often lead researchers to favour unsupervised methods for event detection [16]. As a result, many recent news event detection tasks have focused on unsupervised learning [1] [2], [17], and [18], which employ clustering algorithms with traditional textual embedding.

In recent years, the emergence of LLMs has provided a new opportunity for improving tasks, such as data analysis, information retrieval, and question-answering systems. However, there is a paucity of adequate research on using LLMs to improve clustering and assess their effectiveness in event detection. In this paper, we aim to demonstrate the impact of LLMs to enhance event clustering through both pre-event detection and post-event detection tasks. In the pre-event detection (or pre-clustering) task, we used LLMs for keyword generation and text embedding by identifying subtle semantic connections. In the post-detection task, LLMs were used for event labelling, summarization, and ITPC topic identification. Moreover, we evaluate the clustering results using a stability-based cluster validation index, a newly proposed method designed to assess both the robustness and validation of event clustering.

III. STUDY FRAMEWORK

This section presents an overview of the overall architecture of the proposed framework for news event detection, as illustrated in Figure 1. The framework begins with the acquisition of GDELT data and preprocessing, which involves tasks, such as eliminating special characters, unnecessary words, symbols, and digits. Subsequently, keyword extraction and conceptual embedding of documents

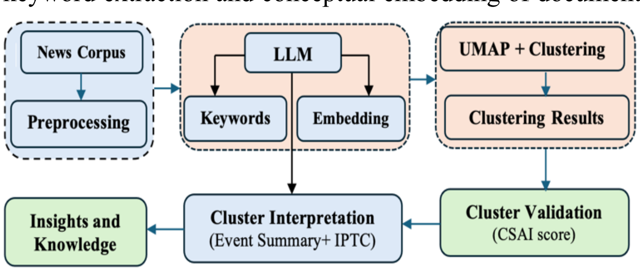


Fig.1. High-level overview of our framework for news event detection

are executed, using LLM and the traditional keyBERT model. UMAP representation of the data is performed to visualize the embedded feature vectors and mitigate processing time and storage complexity. The next phase of the pipeline involves selecting and applying a clustering algorithm to generate internally coherent clusters with distinct characteristics. The next crucial task is assessing the robustness and validation of the generated clusters, which is accomplished through a stability-based assessment of clustering results. Finally, we leverage LLM for interpreting the clusters, which involves labelling, summarizing, and extracting the event groups using IPTC topics.

A. Dataset and Preprocessing

A collection of news documents (about 15,000 news articles) was collected from a massive and regularly updated dataset of online, TV and news reporting from GDELT. The GDELT project diligently monitors global media through diverse perspectives, capturing and analyzing elements like themes, sentiments, geographic locations, and occurrences. It integrates instantaneous translation across 65 languages and tracks more than 2,300 emotions and themes in each news piece, with updates provided every 15 minutes. GDELT is an example of publicly accessible big data which is available on the Google Cloud Platform. Within GDELT, the Global Knowledge Graph (GKG) serves as a main component, housing essential data such as sentiment scores, themes, and locations derived from worldwide newspaper articles. The GKG can be populated through the use of Python and GDELT API, which analyze global newspaper articles in real-time [19].

The news data collected from GDLET requires pre-processing before it can be analyzed to improve the efficiency and accuracy of clustering algorithms [20]. The initial step involves removing miscellaneous elements like irrelevant metadata, HTML tags, and any other content that could distort the analysis. We systematically eliminated invalid non-Latin characters from the dataset, which might result from multilingual data sources or data collection artefacts like encoding errors. Keeping these characters could unnecessarily increase the feature space, hindering clustering performance in terms of computational efficiency and result interpretation.

B. LLMs in Clustering

In recent years, notable progress has been seen in LLMs, such as GPT-4 [21], and LLaMA-2 [22], which have shown remarkable abilities in zero-shot and few-shot learning. These models find application across diverse domains, effectively tackling tasks ranging from chatbots to language translation and content generation. In this work, we propose to use LLM for various tasks in the clustering process including keyword extraction, text embedding, and interpretation of cluster contents by generating summaries and IPTC topic categories. Keyword extraction is an important step in natural language processing, playing a pivotal role across diverse applications like content summarization, information retrieval, and clustering. We used LLM to extract keywords that provide high-level descriptions, themes, or concepts from the content, and then applied various embeddings to each of these keywords. As a result, each news document is passed through the keyword extraction model. We employ the *keyBERT* and *"GPT-3.5-turbo*" models for extracting and refining a list of keywords and phrases as well as to describe and label each cluster.

The generated keywords are then encoded by an embedding model. Embeddings can capture the semantic meaning and syntactic information in a text. Applying the right embeddings to represent text data is essential to achieve optimal outcomes. In this study, OpenAI's state-of-the-art text embedding model *(text-embedding-ada-002)*[9] is used which can generate high-dimensional vectors that effectively capture semantic similarities between words and phrases. Unlike traditional methods, such as bag-of-words or TF-IDF (term frequency-inverse document frequency), which often rely on shallow representations of text, *text-embedding-ada-002* leverages advanced techniques to embed text into continuous vector spaces where similar meanings are encoded closer together. We have also explored other text embedding techniques, such as BERT embedding and Glove and compared the results with LLM embedding. To characterize each of the event clusters, we applied few-shot text summarization and event categorization using IPTC taxonomy, leveraging the same family of GPT models (GPT-3.5-turbo-instruct). This makes the clusters understandable and aids in interpreting the characteristics and profiles of the clustered events.

C. Dimensionality Reduction

Dimensionality reduction aims to remove noise from data dimensions, enhancing clustering accuracy while also lowering computational costs. Furthermore, applying clustering to all dimensions is impractical for higher-dimensional data. In this paper, we primarily used UMAP (Uniform Manifold Approximation and Projection) [23] to represent the reduced version of our data. In comparison to t-SNE [24], UMAP not only preserves more global structures but also offers improved processing speed. UMAP's main parameters include n-neighbors and min-dist, which play a crucial role in balancing local and global structures during dimensionality reduction. The n-neighbours parameter controls how UMAP treats local versus global structure by specifying the size of the local neighbourhood. Smaller values encourage UMAP to prioritize local structures, whereas larger values prioritize global structures, potentially sacrificing finer details. On the other hand, the min-dist parameter determines the minimum distance between points in the low-dimensional space, affecting clustering tightness. Lower values promote finer topological structures, while higher values result in looser point clustering, emphasizing the preservation of broader topological structures. A 2D projection of the data using the t-SNE and UMAP methods is shown in Figure 2.

D. Clustering Algorithms

To identify patterns in the vector representations of news text, we use various clustering algorithms and evaluate their performance. We use agglomerative hierarchical clustering, k-means, HDBSCAN (hierarchical density-based spatial

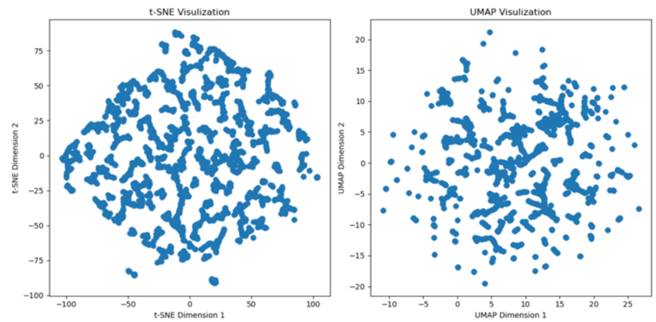


Fig. 2. 2D projections of the GDELT data via t-SNE (left), and UMAP (right)

clustering of applications with noise) and the Gaussian mixture model, which have different ways of grouping data into finite sets of categories [25][26]. Hierarchical clustering approaches can be agglomerative or divisive. An agglomerative clustering begins with one object for each cluster and recursively merges two or more of the most appropriate clusters. A divisive clustering starts with the dataset as one cluster and recursively splits the most appropriate cluster. The process continues until a stopping criterion is reached. K-means is one of the most widely used clustering techniques. The reasons behind the popularity of K-means clustering are the ease of implementation, simplicity, efficiency, and empirical success [27]. However, it is sensitive to the initial configuration and may converge to local optima. HDBSCAN is a density-based method used to discover clusters of non-spherical shapes. To find clusters of arbitrary shapes, clusters are modelled as a dense region in the data space, separated by a sparse region [28]. Instead of finding clusters with specific shapes, HDBSCAN identifies denser regions compared to their surroundings. It is sensitive to noise and capable of detecting clusters with arbitrary shapes. Unlike K-means clustering, which requires specifying the number of clusters, HDBSCAN does not require the user to specify the number of clusters to be generated. HDBSCAN is strongly dependent on various parameters, such as min-cluster-size and min_samples. Gaussian mixture model is a model-based method that optimizes the fit between the given data and some (predefined) mathematical model. It assumes that the data is generated by a mixture of underlying probability distributions. Also, it leads to a way of automatically determining the number of clusters based on standard statistics, taking noise into account, and thus yielding a robust clustering method.

IV. Quality Assessment and Embeddings

This section presents the experimental results, evaluation of clustering algorithms, and impact of various embeddings on event clustering, along with a detailed discussion.

A. Evaluation of Clustering Results

To assess the quality of generated event clusters, we proposed a New Cluster Stability Assessment Index (CSAI), which is based on the similarity of feature vectors between the training data used to create clusters and the validation data used to verify the stability of clusters. CSAI is based on the idea of prior work on multi-label data clustering [29]. However, instead of using a set of output labels and their associated probabilities to compute cluster indices, we rely on a set of input features and their similarities for inference on textual news datasets. Moreover, in CSAI, clusters are formed by partitioning the training part of the dataset into multiple subsamples, with no portion reserved for validation purposes. Evaluation of results is conducted separately using a dedicated validation dataset. CSAI employs normalised root mean squared error (NRMSE) as the distance measure between feature values in the validation dataset and feature scores in the training dataset corresponding to a specific cluster. Formally, CSAI is expressed in (1), while the dataset, algorithm, and source code for CSAI are available at https://github.com/adane04/CSAI.

$$\mathrm{CSAI} = \frac{1}{\mathrm{k}}\sum_{j=1}^{K}\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{(T_{max}-T_{\min})}\sqrt{\frac{1}{F}\sum_{i=1}^{F}(V_i-T_i)^2}\right) \qquad (1)$$

where K represents the total number of partitions in the training data, and N is the total number of clusters generated from each partition. T_{max} and T_{min} denoting the maximum and minimum values of features in the training data, respectively. F is the total number of features, T_i and V_i represent the numeric value of i^{th} feature in the training and validation set, respectively. When the CSAI values are low, it indicates greater stability in the outputs of the clustering algorithm. High cluster stability occurs when minor variations in the dataset do not affect the cluster memberships.

B. *Effect of Embeddings on Clustering Results*

We carried out a set of experiments to evaluate the impact of various embeddings on clustering outcomes, comparing traditional embedding methods, such as TF-IDF [30], GloVe [31] and BERT [32] with the state-of-the-art LLM embeddings. The traditional TF-IDF vectors acted as a baseline, providing a sparse yet interpretable representation by focusing on word significance across the analyzed dataset. GloVe generated word embeddings using a word co-occurrence matrix, with each matrix element indicating how often two words appeared together within a given context window. BERT embeddings, derived from the BERT model trained on English Wikipedia [33] and BookCorpus [34], were utilized to gain deep contextual understanding through a transformer-based bidirectional encoder. These embeddings could capture subtle semantic differences across the corpus. Additionally, we used "text-embedding-ada-002", as it had shown superior performance among OpenAI's embeddings for text search, code search, and sentence similarity on larger datasets. We also aim to identify which clustering algorithm performs best on which embedding type. For each embedding type, we apply five clustering algorithms: two partitionings (k-means and k-medoid), one

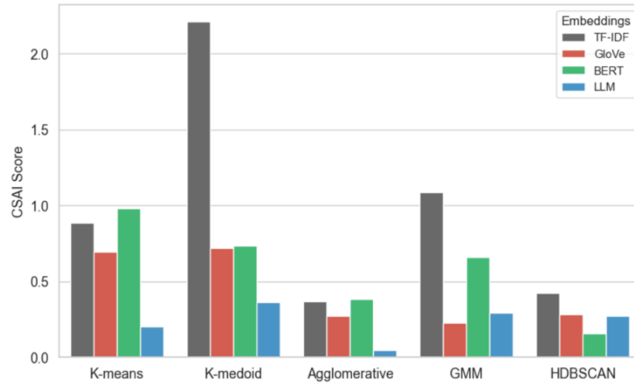


Fig. 3. Text clustering results based on CSAI scores on various word embedding models.

hierarchical (agglomerative clustering), one model-based (gaussian mixture model (GMM)), and one density-based (HDBSCAN). We then evaluate the performance of each algorithm using our proposed index (CSAI), as shown in Figure 3. We aim to investigate how various text embedding techniques impact the effectiveness of clustering algorithms.

The top-performing algorithm was identified by selecting the one with the lowest CSAI score for each type of embedding. Our results indicate that OpenAI's LLM embeddings outperform other embedding methods, consistently achieving the lowest CSAI scores across all clustering algorithms, with the exception of HDBSCAN, where it worked well with the BERT embedding. By "lowest," we mean the algorithm with the best CSAI score. Among the clustering algorithms, agglomerative clustering combined with the LLM embeddings yielded the best CSAI scores, followed by k-means clustering on the same embeddings. This outcome may be linked to the fact that LLM embeddings are trained on a vast and varied corpus of Internet text, making them exceptionally adept at understanding the intricacies of language patterns. Conversely, lower CSAI scores for this algorithm could suggest that the resulting clusters may lack clear boundaries or stability when assessed across multiple samples. In terms of BERT, HDBSCAN has achieved better CSAI scores compared to other clustering models.

After identifying the best-performing clustering

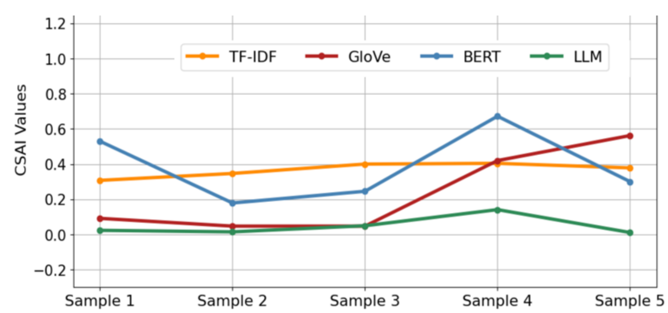


Fig. 4. Agglomerative clustering score in terms of CSAI, across five samples using GloVe, BERT, and LLM embeddings.

algorithm with the CSAI validity measure, the next step is to evaluate the robustness or stability of clustering results.

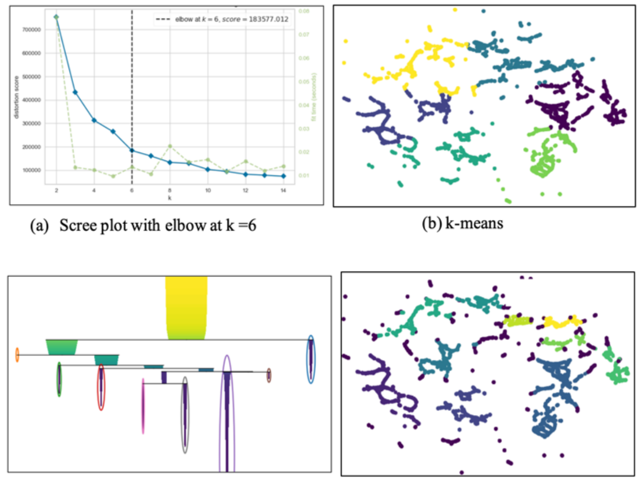
Validity measures how accurately the clustering results capture the actual structure of the data, while robustness assesses the consistency of the clusters when subjected to different conditions like random initialization, resampling, data perturbation, or variations in the algorithm [35]. Since the agglomerative hierarchical clustering algorithm yielded the best results in terms of the CSAI validity analysis, we further examined the robustness of the clusters using a method known as stability analysis. We used the CSAI score again to assess the robustness of clusters across five different partitions of the data using various text embedding methods. The results in Figure 4 indicate that the agglomerative algorithm shows greater variability across the five partitions when using the BERT embedding, with the clustering model yielding a different CSAI value for each partition. There is also a slight variation in CSAI scores with GloVe, with the highest and lowest values observed on partitions 3 and 5, respectively. On the other hand, LLM and TF-IDF demonstrate greater stability across the five partitions. LLM consistently delivers the best CSAI values for each partition, while TF-IDF has the worst CSAI scores for partitions 2 and 3. The average CSAI values for agglomerative hierarchical clustering across the five partitions were 0.0483 and 0.3676 for LLM and GloVe, respectively. Generally, when comparing embeddings based on CSAI scores, it is clear that clusters generated using the agglomerative algorithm with LLM embeddings demonstrate greater robustness compared to other embeddings across all five samples.

V. CLUSTERING VISUALIZATION AND POST-DETECTION

A different number of event clusters have been discovered using various clustering algorithms on the GDELT news dataset. Determining the optimal number of clusters always varies from algorithm to algorithm. For example, in k-means clustering, we specify the optimal number of clusters (k) using the "elbow" method applied to the variance explained (i.e., the Within-Cluster-Sum of Squared Errors (WSS)) [36], where six potential clusters were identified by computing WSS against k, as depicted in Figure 5(a). likewise, in hierarchical clustering, a dendrogram visually represents the arrangement of clusters, allowing us to find the optimal number of clusters by analysing where to make effective "cuts." HDBSCAN starts by providing a clustering hierarchy from which a simplified tree of significant clusters can be constructed. The hierarchy can easily be visualized as a traditional dendrogram or related representations from which only significant clusters are extracted. Figure 5 presents a scree plot with the elbow point for k-means clustering and a condensed cluster tree for HDBSCAN, along with their respective clustering plots.

Besides visualization, post-event detection analysis is necessary for most applications, which includes tasks such as event labelling, event summarization and topic identification. Event clusters resulting from the clustering process must be labelled to facilitate the interpretation of the clustering outcomes. Interpretability is the ability of humans to understand the meanings of the outputs of the models in the context of domain knowledge [37]. Clustering, as part of machine learning, requires interpretability since generic labels such as "0," "1," or "2" fail to convey meaningful insights about cluster contents. In order to grasp the essence of the generated event clusters, this study explores the use of LLMs to label clusters using representative keywords, generate summaries for description, and categorize them according to the IPTC (International Press Telecommunications Council) taxonomy. The labelling entails the assignment of descriptive or explanatory terms to the clusters, aiding in understanding their contents and characteristics. We also assigned the type of IPTC taxonomy to each cluster facilitating the categorization of clusters according to predefined news topics, helping in understanding and comprehending the thematic focus of each news event. The other important post-detection task is event summarization, which can be essential for highlighting key aspects of an event. It can also support newsworthiness ranking and event veracity identification (rumour detection), providing value to news agency users. Figure 6 displays a snapshot of the news event clusters alongside their corresponding keywords, ten-word summaries, and associated IPTC topics, all derived from LLM analysis.

The summary aids in conveying the results by providing a concise description of each news event. As we can see



(a) Scree plot with elbow at k =6 (b) k-means

(c) Condensed cluster tree (d) HDBSCAN

Fig. 5. Illustration of the optimal selection of clusters in k-means and HDBSCAN via the UMAP subspace and the corresponding clustering plots. Plot (a) shows a scree plot with 'elbow' to determine the optimal number of clusters (k) in k-means based on WSS, and (c) represents the condensed tree visualization help to indicate clusters identified by HDBSCAN. Plots (a) and (d) represent the 2D scatter plots of clustering results using k-means and HDBSCAN, respectively.

Cluster	Text	keywords	Summary	IPTC_Type
3	raphael lavoie s...	[ahl, lavoie, co...	Raphael Lavoie s...	Sports
5	the university o...	[oxford, univers...	Oxford Universit...	Technology/Science
0	secure energy se...	[analysts, ratin...	Secure Energy Se...	Business
1	for most of its ...	[freshwater, irr...	Study shows 81% ...	Environment
4	microsoft is cur...	[xbox, microsoft...	Microsoft workin...	Technology/Science

Fig. 6. Event clusters with keywords, summary, and topic type

from Figure 6, one can readily grasp the heterogeneity and core characteristics of each cluster through concise summaries and IPTC types. For example, clusters 0 and 1 have news stories related to business and the environment, respectively, while clusters 3 and 5 are focused on sports and technology news. Keywords and summaries provide quick insights into each article. For instance, cluster 1 includes an article discussing the Colorado River, revealing that 81% of its content is used by people for various activities. Such results suggested by our framework seem to be logical and provide additional assurance for the validity of clustering results.

VI. Conclusions

In this paper, we propose an event detection framework leveraging the global GDLET news database. The framework comprises various components, including keyword extraction, text embedding, and event summarization aided by LLM, together with dimensionality reduction and clustering algorithms. We explore the use of different text embeddings to represent news articles and study how these embeddings impact the effectiveness of various clustering algorithms. Our analysis compares the LLM embedding model with traditional techniques such as TF-IDF, GloVe, and BERT, investigating their influence on a range of clustering methods. Specifically, we evaluate how these embeddings influence the results of various clustering algorithms in terms of stability-based cluster validation index (CSAI). CSAI is introduced in this study to measure the quality of clustering results, which operates by initially assigning new data points to existing clusters constructed from the training dataset. Then, it computes the similarity between feature vectors in training and testing samples allocated to the same cluster. Experimental results demonstrate that LLM embedding yields superior clustering performance compared to other methods based on CSAI score, although they have a disadvantage in terms of inference time. Additionally, LLM contributes to enhancing cluster interpretability through summary generation and topic assignment for each event cluster. In the future, we plan to explore state-of-the-art AI approaches for analyzing event progression and monitoring, specifically focusing on societal events of interest including the ones defined in the CAMEO ontology [38].

Acknowledgement

This research is funded by SFI MediaFutures partners and the Research Council of Norway (grant number 309339).

References

[1] Z. Lin, J. Xie, and Q. Li, “Multi-modal news event detection with external knowledge,” Inf. Process. Manag., vol. 61, no. 3, p. 103697, May 2024, doi: 10.1016/j.ipm.2024.103697.

[2] Z. Liu, Y. Zhang, Y. Li, and Chaomurilige, “Key News Event Detection and Event Context Using Graphic Convolution, Clustering, and Summarizing Methods,” Appl. Sci., 2023, doi: 10.3390/app13095510.

[3] R. Toujani and J. Akaichi, “Event news detection and citizens community structure for disaster management in social networks,” Online Inf. Rev., 2019, doi: 10.1108/OIR-03-2018-0091.

[4] A. N. Tarekegn, T. A. Alemu, and A. K. Tegegne, “A cluster-genetic programming approach for detecting pulmonary tuberculosis,” Ethiop. J. Sci. Technol., 2021, doi: 10.4314/ejst.v14i1.5.

[5] K. Leetaru and P. A. Schrodt, “GDELT: Global Data on Events, Location and Tone, 1979-2012,” Annu. Meet. Int. Stud. Assoc., 2013.

[6] I. Keraghel, S. Morbieu, M. Nadif, I. Keraghel, S. Morbieu, and M. Nadif, “Beyond words : a comparative analysis of LLM embeddings for effective clustering To cite this version : HAL Id : hal-04488175 Beyond words : a comparative analysis of LLM embeddings for effective clustering,” 2024.

[7] V. Viswanathan, K. Gashteovski, C. Lawrence, T. Wu, and G. Neubig, “Large Language Models Enable Few-Shot Clustering”, arXiv:1612.06830 [cs.CL], 2023.

[8] M. Grootendorst, “KeyBERT: Minimal keyword extraction with BERT,” Zenodo, 2020. .

[9] A. N. Ryan Greene, T. Sanders, L. Weng, “New and improved embedding model.” https://openai.com/blog/new-and-improved-embedding-model (accessed May 06, 2024).

[10] S. Agarwal, A. Singhal, and P. Bedi, “IPTC based ontological representation of Educational News RSS Feeds,” 2013, doi: 10.1007/978-1-4614-3363-7_39.

[11] R. Xu and D. Wunsch, “Survey of clustering algorithms,” IEEE Transactions on Neural Networks. 2005, doi: 10.1109/TNN.2005.845141.

[12] T. Liu, H. Yu, and R. H. Blair, “Stability estimation for unsupervised clustering: A review,” Wiley Interdisciplinary Reviews: Computational Statistics. 2022, doi: 10.1002/wics.1575.

[13] Q. Li, Y. Chao, D. Li, Y. Lu, and C. Zhang, “Event Detection from Social Media Stream: Methods, Datasets and Opportunities,” 2022, doi: 10.1109/BigData55660.2022.10020411.

[14] J. Allan, R. Papka, and V. Lavrenko, “On-line New Event Detection and Tracking,” 1998, doi: 10.1145/290941.290954.

[15] G. Akiladevi, M. Arun, and J. Pradeepkandhasamy, “Event Detection in Social Media Analysis: A Survey,” 2023, doi: 10.1007/978-981-99-5166-6_4.

[16] H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, and M. M. Gaber, “Embed2Detect: temporally clustered embedded words for event detection in social media,” Mach. Learn., 2022, doi: 10.1007/s10994-021-05988-7.

[17] S. Carta, S. Consoli, L. Piras, A. S. Podda, and D. R. Recupero, “Event Detection in Finance Using Hierarchical Clustering Algorithms on News and Tweets,” PeerJ Comput. Sci., 2021, doi: 10.7717/PEERJ-CS.438.

[18] P. Goyal, P. Kaushik, P. Gupta, D. Vashisth, S. Agarwal, and N. Goyal, “Multilevel Event Detection, Storyline Generation, and Summarization for Tweet Streams,” IEEE Trans. Comput. Soc. Syst., 2020, doi: 10.1109/TCSS.2019.2954116.

[19] K. H. Leetaru, T. K. Perkins, and C. Rewerts, “Cultural computing at literature scale: Encoding the cultural knowledge of tens of billions of words of academic literature,” D-Lib Mag., 2014, doi: 10.1045/september2014-leetaru.

[20] A. Petukhova and N. Fachada, “TextCL: A Python package for NLP preprocessing tasks,” SoftwareX, 2022, doi: 10.1016/j.softx.2022.101122.

[21] OpenAI et al., “GPT-4 Technical Report,” vol. 4, pp. 1–100,

2023.
[22] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” 2023.
[23] L. McInnes, J. Healy, N. Saul, and L. Großberger, “UMAP: Uniform Manifold Approximation and Projection,” J. Open Source Softw., 2018, doi: 10.21105/joss.00861.
[24] M. Wattenberg, F. Viégas, and I. Johnson, “How to Use t-SNE Effectively,” Distill, 2017, doi: 10.23915/distill.00002.
[25] D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms,” Ann. Data Sci., 2015, doi: 10.1007/s40745-015-0040-1.
[26] W. S. Sarle, A. K. Jain, and R. C. Dubes, “Algorithms for Clustering Data,” Technometrics, 1990, doi: 10.2307/1268876.
[27] A. K. Jain, “Data clustering: 50 years beyond K-means,” Pattern Recognit. Lett., 2010, doi: 10.1016/j.patrec.2009.09.011.
[28] R. J. G. B. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” 2013, doi: 10.1007/978-3-642-37456-2_14.
[29] A. N. Tarekegn, K. Michalak, and M. Giacobini, “Cross-Validation Approach to Evaluate Clustering Algorithms: An Experimental Study Using Multi-Label Datasets,” SN Comput. Sci., 2020, doi: 10.1007/s42979-020-00283-z.
[30] Y. Gu, Y. Wang, J. Huan, Y. Sun, and S. Xu, “An improved TFIDF algorithm based on dual parallel adaptive computing model,” Int. J. Embed. Syst., 2020, doi: 10.1504/IJES.2020.108278.
[31] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” 2014, doi: 10.3115/v1/d14-1162.
[32] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2019.
[33] Zenodo.org, “Wikipedia Complete Citation Corpus.” https://zenodo.org/records/6539054 (accessed May 06, 2024).
[34] Y. Zhu et al., “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” 2015, doi: 10.1109/ICCV.2015.11.
[35] Y. Z. Hu, Y. Hu, X. B. Ai, H. Y. Zhao, and Z. Meng, “Clustering stability evaluation method based on SSIM,” J. Algorithms Comput. Technol., 2019, doi: 10.1177/1748302619873592.
[36] A. Hartigan and M. A. Wong, “A K-Means Clustering Algorithm,” J. R. Stat. Soc., 1979.
[37] A. Erasmus, T. D. P. Brunet, and E. Fisher, “What is Interpretability?,” Philos. Technol., 2021, doi: 10.1007/s13347-020-00435-2.
[38] P. A. Schrodt, Ö. Yilmaz, D. J. Gerner, and D. Hermreck, “The CAMEO (conflict and mediation event observations) actor coding framework,” 2008 {Annual} {Meeting} {International} {Studies} {Association}, 2008.