Large Language Model (LLM)-Driven Document Clustering: Improving Real-time Security Intelligence Extraction and Threat Analysis

Patrick Serrano
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
pserrano@andrew.cmu.edu

Luwen Huangfu*
San Diego State University
San Diego, California, USA
lhuangfu@sdsu.edu

Chunhua Liao
Lawrence Livermore National Laboratory
Livermore, California, USA
liao6@llnl.gov

Dongqing Lin
University of California, Irvine
Irvine, California, USA
dongqinl@uci.edu

Kai Williams
San Diego State University
San Diego, California, USA
kwilliams2232@sdsu.edu

Jaden Perleoni
San Diego State University
San Diego, California, USA
jperleoni3571@sdsu.edu

Thomas Brettin
Argonne National Laboratory
Chicago, Illinois, USA
brettin@anl.gov

***Abstract*—The sheer volume and rapid expansion of unstructured text in various fields make it challenging for cybersecurity practitioners and researchers to extract essential information from documents. Inefficient clustering can hinder the timely extraction of intelligence, especially in areas such as network security, where real-time requirements are high. Currently, the use of descriptive tags for information clustering, although helpful, is not standardized within a given domain, causing useful information from other domains with different descriptive tags to be neglected. To address these issues, we introduce a novel Large Language Model (LLM)-based document clustering process that (1) generates a series of relevant keywords based on the content of the documents, and (2) leverages LLMs to cluster various datasets of intelligence information spanning different topics. We measure the quality of the clusters using several established indices: Overall Density, Overall Distinctiveness, Coherence and Overlap coefficients, and Label Entropy. Based on these evaluation metrics, the proposed approach performs better than traditional methods on multiple datasets in the field of Advanced Persistent Threat (APT). This demonstrates that it aids in the clustering of unlabeled and unstructured textual data, and illustrates the potential for improving intelligence document clustering practices through the use of LLMs in cybersecurity and other fields.**

***Index Terms*—Artificial Intelligence (AI), Large Language Models (LLMs), Generative Tagging, Document Clustering, Advanced Persistent Threat (APT).**

I. Introduction

Intelligence information extraction is a key process in many fields, including cybersecurity, scientific research, and business analysis [1]. This process enables researchers and professionals to improve decision-making efficiency and enhance intelligence monitoring and early warning capabilities. If information is overloaded, researchers may become overwhelmed by the large amount of data, resulting in a loss of insight and reduced work efficiency [2].

In the author list, * indicates the corresponding author

Two critical problems arise: One is the ever-evolving nature of intelligence information, and the second is the intelligence within particular domains can be obscured by the vast expanse of other unrelated information [3]. These descriptive tags can ease such problems, but they are often devoid of any cross source consistency due to the absence of a standardized form [4]. Furthermore, intelligence throughout many domains or over a set of domains is not always definable by the tags assigned to it, especially when its scope is too broad to be adequately described solely through the use of descriptive tags only [5].

Therefore, automated clustering methods offer a solution to the limited scope of descriptive tags. Including the most relevant aspects of documents and generating concise, high-quality labels are key to producing excellent intelligence information extraction clustering results. The rapid accumulation of intelligence text data requires clustering solutions that can scale efficiently without compromising performance. Because traditional algorithms may not be able to meet this requirement, more powerful methods are needed. The research question of this study focuses on: using Large Language Models (LLMs) in intelligence information extraction to improve document clustering performance in unstructured text, and compares the results with traditional clustering methods based on the BERT model and K-means clustering algorithm.

We believe that LLMs can significantly enhance the process of topic modeling and document clustering for intelligence information. LLMs have a stronger ability to both generate theme-aware keywords for a dataset [3] and understand the context in which keywords appear, allowing them to capture the nuanced relationships between words, topics, and concepts [6] [7]. This contextual understanding enables LLMs to extract more meaningful and relevant key-

words, reducing the occurrence of synonymous or redundant keywords that may hinder the clustering process [7]. Furthermore, LLMs can handle the complexities of natural language processing tasks with fewer hyperparameters, making them more user-friendly and adaptable to various domains [8]. By leveraging the power of LLMs, we aim to improve the accuracy and interpretability of topic modeling and document clustering, ultimately facilitating a more efficient and organized exploration of intelligence information. In this research, we present a topic extractor model which generates a series of relevant keywords as labels for a given piece of intelligence information, and we present a novel approach to document clustering whereby we leverage the natural language processing capabilities of LLMs.[1] We achieve this through the use of LLMs, which assist in explaining and highlighting the most prominent subjects in intelligence information and other unstructured data. Consequently, we facilitate a more efficient, precise, and explanatory research process.

Our paper makes the following contributions:

- We propose a LLM-based document clustering process. It is suitable for processing unstructured intelligence information in a variety of intelligence information fields, including cybersecurity.
- We evaluate our LLM-based document clustering process on multiple APT datasets using the silhouette score, Davies-Bouldin Index, Calinski-Harabasz Index, and Dunn Index.
- Our results outperform the traditional BERTopic document clustering model, based on SentenceTransformer encoding, and K-means clustering with the BERT model.
- We introduce an automated, scalable clustering method for professionals in cybersecurity and other intelligence information fields, using a simple yet effective approach to organize intelligence information documents.

II. Background

LLMs usually refer to language models with tens of billions (or more) of parameters. These models are constructed with deep learning techniques, and they learn from extensive text corpora to master the language's syntax and semantics patterns [9].

Retrieving useful information from unstructured data without any organization or predefined structures is often a complex task [10]. Therefore, expertise in natural language processing and machine learning is necessary to identify correlations between intelligence data [11]. Generative clustering is a method that aims to learn the underlying structure of data to generate clusters in an adaptive manner, utilizing unsupervised learning algorithms to identify patterns and relationships in the data [12]. In this context, LLMs play a crucial role in addressing the challenges associated with exploring intelligence information. Their advanced natural language processing capabilities enable them to analyze nuanced relationships between words, topics, and concepts, making them well-suited for generative clustering tasks. By utilizing LLMs for keyword extraction and highlighting prominent elements in intelligence information research, researchers can contribute to a more efficient and organized research process, ultimately enhancing the exploration of intelligence information [13].

Existing LLM-based clustering methods mainly use LLMs to generate embeddings, followed by K-Means for text clustering and dimensionality reduction [7], or combine Gaussian Mixture Models (GMMs) for clustering [14]. In addition, existing LLM-based clustering methods are mainly tested on general datasets, with limited attention to intelligence information [7] [15] [16].

As a result, existing LLM-based clustering methods face two challenges: First, while methods using GPT embeddings and K-Means have improved in terms of cluster cohesion, they often lack interpretability because they do not provide meaningful labels for clusters. Second, GMMs are suitable for manifold structures in low-dimensional space, but they may encounter problems with low computational efficiency and large memory usage when dealing with large-scale, high-dimensional security intelligence data.

Therefore, this study not only uses the text understanding ability of LLMs to make the clustering performance surpass the traditional clustering performance based on BERTopic and GMM on multiple benchmarks, but also LLMs generative abilities to dynamically extract and summarize labels, rather than clustering document embeddings directly. This study addresses the interpretability issue by integrating generative keyword extraction. This approach, which avoids the challenges of clustering using high-dimensional vectors and better aligns the process with practical applications of intelligence analysis. Our study addresses this gap by applying LLM-driven clustering to cybersecurity intelligence, using APT reports to enhance threat intelligence classification. Furthermore, the structured automated document processing pipeline we proposed eliminates the need for human intervention, making it scalable for intelligence analysis. This provides a scalable automated approach for clustering intelligence reports.

III. Methodology

A. Overview

This study describes a document processing and clustering system driven by Transformer-based LLMs, which differ from traditional document clustering models.

The core innovation of this study lies in the fully automated architecture without manual feature engineering in two stages: Keyword Extraction and Topic-based Clustering, which not only uses LLM for embedding, but also for semantic keyword extraction and document clustering, thus achieving the interpretability, scalability and automation of multi-label classification of intelligence documents, and is evaluated on a real-world cybersecurity dataset.

[1] `Link`: https://github.com/AI4B-Lab-SDSU/LLM-Sec-Tagger

B. Tagger model architecture

We take the application of advanced persistent threat (APT) documents with the LLM GPT-4o as an example. The proposed system utilizes a two-stage architecture, calling GPT-4o's API multiple times using different methods, highlight the excellent text extraction and classification capabilities of LLMs based on the Transformer.

- Stage 1: Keyword Extraction: First, the system includes extracting a list of the most important keywords and phrases of each APT document and saving them as labels of the document.
- Stage 2: Clustering: Second, it then groups the APT documents together by the generated labels. The second stage uses the LLM to combine keywords from multiple documents into merged topic groups, and categorize the documents into topics based on the combined keywords.

C. System Implementation and Automation

We take the application of advanced persistent threat (APT) documents with the LLM GPT-4o as an example. GPT-4o is a state-of-the-art LLM based on Transformers, which are used in various datasets for natural language understanding and generation tasks. They aid in automatically classifying document files into topics and generating clustering results.

To achieve repeatability and eliminate human intervention, This system can fully automatically perform text extraction, document classification, and cluster topics on APT documents, and it adopts a structured approach for input and output through a series of well-defined steps. The two phases of calling LLMs in this study involve dynamically obtaining input prompts from text files, automatically capturing the output generated by the model and recording it to files, thereby simplifying the process of subsequent analysis or practical application without human intervention. System prompts play a key role in defining the functions of each model interaction, and the prompts themselves are the text input for each iteration.

The two-stage process is formally encapsulated in the following pipeline:

- Input: raw APT intelligence documents.
- LLM calling (Stage 1): extract keywords using prompt templates.
- Intermediate output: document label pairs.
- LLM call (Stage 2): Clustering based on keyword labels using prompt templates.
- Final output: The result of mapping topic classification labels to APT intelligence file dataset.

D. Baseline for Comparison

The comparative experiments in this study include a traditional model of K-Means clustering based on the BERTopic and Gaussian Mixture Model (GMM).

E. Dataset Description

This study uses three datasets for testing, all of which contain intelligence data are stored in PDF format. The three datasets are used to test and distinguish the performance of the model on general intelligence, APT intelligence, and homogenized APT intelligence.

Test Case 1: MLScholar

The dataset used for testing MLScholar is DS1. DS1 is a dataset of machine learning intelligence showcasing diverse topics across the field.

Test Case 2: APT-Diverse

The dataset used for testing APT-Diverse is DS2. DS2 is the APTnotes dataset [17], which is an open-source APT (Advanced Persistent Threat) intelligence repository. APTnotes contains various public reports, research analysis, and incident investigation information on APTs from major cybersecurity companies such as FireEye and Mandiant. The dataset covers the tactics, techniques, and procedures (TTPs) of APT organizations, attack indicators (IoC), and detailed descriptions of attack events. It provides examples of rich historical attack cases and threat intelligence information.

Test Case 3: APT-Technical

The dataset used for testing APT-Technical is DS3. DS3 is the APT & Cybercriminals Campaign Collection dataset [18]. Like DS2, this is an open-source APT intelligence repository: a collection of APT and cybercriminals' activities.

However, unlike DS2, DS3 is more homogeneous. DS3 places higher demands on the performance of clustering models because the documents contain highly similar content, making it more difficult to distinguish them.

F. Specific design of the LLM

Stage 1: Keyword Extraction

To ensure that only the important aspects of the documents get fed into Stage 2, we first classify documents by topics. In Stage 1 of our LLM-based approach, we first call GPT-4o's API with the original text of the APT documents and a prompt instruction to extract a list of the most important keywords and phrases based on the APT documents content. In this process, we aim to leverage the natural language processing capabilities of the model to extract a list of relevant, concise, and comprehensive keywords. Each list of keywords is saved as a label for each APT document.

Prompt:

"You are an advanced text-processing AI. Exclude generic words, extract 10-20 of the most significant keywords and phrases, which can best summarize the main topics and ideas from the following text."

{document text}

"Return them in a list format without numbering or dashes, instead just separate them with a new line. Do not produce extra commentary."

Stage 2: Clustering

In the second stage of our LLM-based approach, the APT documents labeled with a list of keywords are input into our

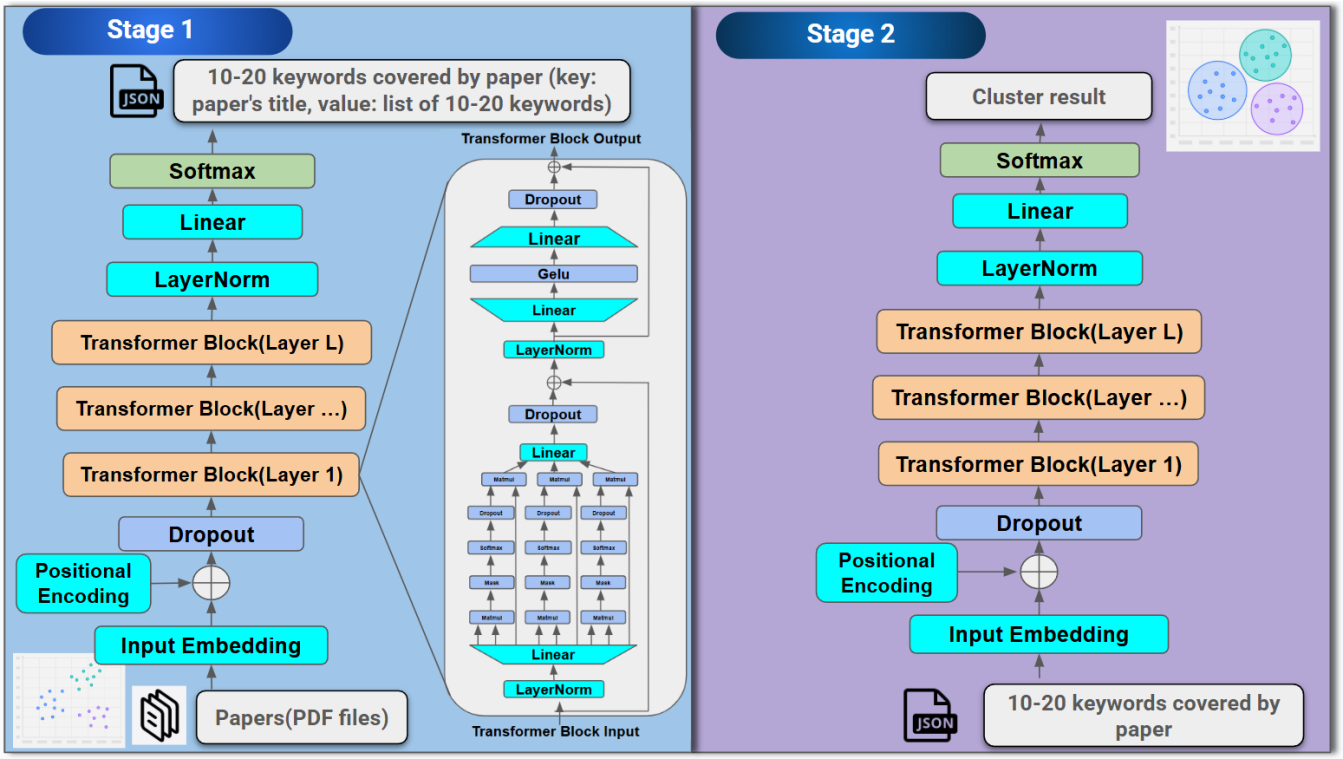

Fig. 1: Architecture of LLM topic extractor and clustering model. Stage 1: Keyword Extraction – It begins with positional encoding and text embedding, followed by multiple Transform blocks that polish the keywords to the maximum possible extent. Finally, the LLM produces label-like keywords that are to be used for clustering in further stages of processing. Stage 2: Document Clustering – The model utilizes additional processing Transformer layers for classifying documents based on common keywords. The result is a final set of document clusters, which the LLM automatically detects the fundamental themes contained in the entire documents and groups them together in an organized manner.

second LLM: GPT-4o for classification. This model groups the documents together according to their generated labels, reflecting common themes, and outputs a valid JSON object where each key is a label, and each value is a list of the document file names under that label.

Prompt:

You are a clustering AI that categorizes papers based on extracted tags.

We have the following papers, each with extracted tags. Please group them together by generating labels that reflect common themes.

Output a valid JSON object where each key is a label and each value is a list of the paper filenames under that label.

Papers must be in at least 1 label but each paper can be part of multiple labels if it fits.

Do not add extra commentary; only output valid JSON.

Papers and their tags:

{joined_papers_text}

Return a JSON of the form:

{"label1": ["PaperA.pdf", "PaperB.pdf"],
"label2": ["PaperC.pdf"]}

G. Traditional model: BERTopic

For traditional methods that directly process unstructured data (PDF) without manual conversion, BERTopic combined with BERT semantic embedding is one of the best performing models for dynamic topic discovery. BERTopic is a topic model technique that generates document embeddings with pretrained transformer-based language models, clusters these embeddings, and generates topic representations with the class-based procedure. BERT can directly process raw text with minimal manual feature engineering [19]. By leveraging BERT-based embedding, BERTopic can capture deep semantic information directly from unstructured text [20] in a way that extracts meaningful and diverse topics [21].

To ensure that documents can be assigned to multiple topics as in LLMs, we use a Gaussian Mixture Model (GMM) to generate multi-label topic assignments. GMM is a probabilistic model that assumes that all data points are generated by a mixture of a finite number of Gaussian distributions with unknown parameters. The mixture model can be thought of as a generalized K-means clustering method which incorporates information about the covariance structure of the data, as well as the potential Gaussian centers.

Data preprocessing

We first combine NLTK's built-in stop words with custom stop words optimized for research documents and remove unnecessary words. Our comparative experiment uses SentenceTransformer (all-MiniLM-L6-v2, based on BERT) to convert APT documents into semantic vectors, retaining contextual information. SentenceTransformer (all-MiniLM-

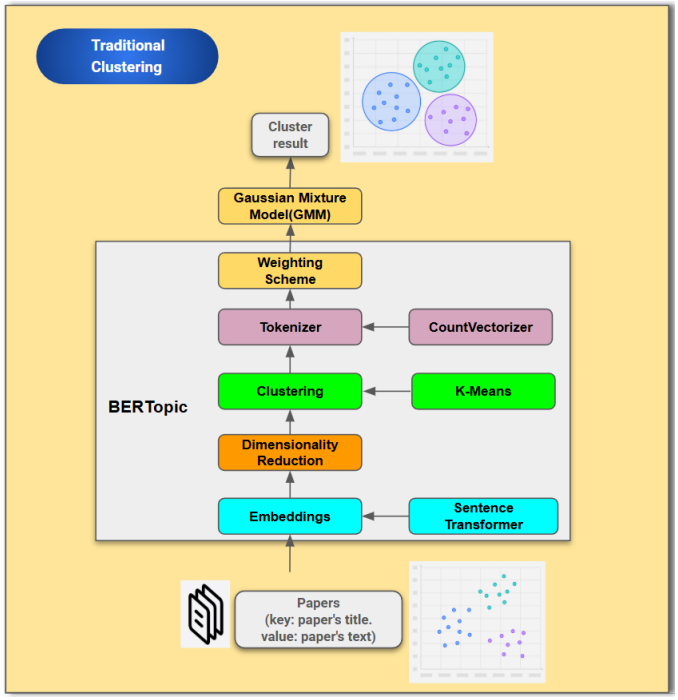

Fig. 2: Architecture of traditional model BERTopic. The figure shows the traditional BERTopic-based clustering model, which processes and groups documents using several steps: The model first performs tokenization and then uses CountVectorizer to turn the words into numerical values. It applies SentenceTransformer to produce text embeddings and subsequently reduces the dimensions for more optimal clustering. During this stage, K-Means is used to cluster documents, while a Gaussian Mixture Model (GMM) is employed to refine topic assignments. The model finally generates topic clusters, but unlike the LLM-based approach, it does not create meaningful labels for the topics.

L6-v2) is a sentence-transformer model that maps sentences and paragraphs to a 384-dimensional dense vector space.

BERTopic for preliminary K-Means clustering

This study uses K-Means along with BERTopic and integrates CountVectorizer for word frequency statistics to achieve dual optimization based on semantics and word frequency. CountVectorizer converts a collection of text to a matrix of token counts. K-means is a technique used for grouping data according to their similarity. To create a cluster, first a number of centroids is selected at random. With each data point, the closest centroid gets assigned. After calculating the average position of the points within each cluster, the centroids are updated, which happens after all points get allocated. This cycle continues until the centroids stop changing, which results in forming clusters. The aim of clustering is to partition the given data points into groups or clusters in a manner that enables the grouping of similar data points in a single cluster.

Multi-label topic assignments

After obtaining preliminary results, we calculate the center vector of each topic and use Gaussian Mixture Model (GMM) to generate multi-label topic assignments. In this process, we calculate the similarity between the text embedding and the topic center, and use cosine similarity to determine whether the document should be assigned to multiple topics.

IV. Results

The key metrics we use to evaluate and compare the performance of LLMs and traditional model include:

- **Overall Density:** Measures the mean cosine similarity between documents that are centered around the same topic. A higher value indicates greater internal similarity among documents and suggests that the quality of the topic is high.

$$\text{Overall Density} = \frac{1}{|C|}\sum_{c\in C}\frac{1}{|c|(|c|-1)}\sum_{\substack{x,y\in c\\ x\neq y}}\text{cosine\_similarity}(x,y) \tag{1}$$

  Where:
  - C denotes the set of all clusters.
  - c denotes a specific cluster (topic).
  - $|c|$ denotes the number of documents within the cluster.
  - x, y denote the vector representations of different documents within the cluster c.
  - $\text{cosine\_similarity}(x, y)$ denotes the cosine similarity calculation between documents.
  - The cosine similarity is defined as:

$$\text{cosine\_similarity}(x,y) = \frac{x\cdot y}{\|x\|\|y\|} \tag{2}$$

- **Overall Distinctiveness:** Measures differentiation in distribution of words across various topics. A high degree of differentiation suggests that the topics are independent of each other and strongly differ in content.

$$\text{Overall Distinctiveness} = \frac{1}{|C|(|C|-1)}\sum_{i\in C}\sum_{\substack{j\in C\\ j\neq i}} D(\theta_i,\theta_j) \tag{3}$$

  Where:
  - C denotes the set of all topics (clusters).
  - θ_i, θ_j denote the feature distributions of topic i and topic j (e.g., keyword vectors or topic centroids).
  - $D(\theta_i, \theta_j)$ denotes a distance or divergence function between two topics.

- **Coherence C_V:** Measures the topic coherence of a particular subject using a co-occurrence matrix and cosine similarity. A greater C_V suggests that the words within the topic tend to co-occur with each another within the text which means the topic is well-defined and coherent.

$$C_V(T) = \frac{1}{n}\sum_{i=1}^{n}\text{cosine\_similarity}\left(v(w_i), \frac{1}{n}\sum_{j=1}^{n} v(w_j)\right) \tag{4}$$

  Where:
  - $T = (w_1, w_2, \ldots, w_n)$ denotes a topic consisting of n keywords.
  - w_i denotes the i-th keyword in the topic.
  - $P(w_i)$ denotes the probability of word w_i occurring in the document collection.

- $P(w_i, w_j)$ denotes the probability of w_i and w_j co-occurring in the same document.
- $v(w_i)$ denotes the vector representation of keyword w_i, defined as:

$$v(w_i) = (\text{NPMI}(w_i, w_1), \ldots, \text{NPMI}(w_i, w_n)) \tag{5}$$

- NPMI (Normalized Pointwise Mutual Information) is computed as:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)+\epsilon}{P(w_i)P(w_j)}}{-\log(P(w_i, w_j)+\epsilon)} \tag{6}$$

  - where ϵ is a small constant to avoid division by zero.

- **Coherence C_UCI:** Measures the coherence relationship of topic words using pointwise mutual information (PMI) by calculating topic consistency. A high C_UCI value means that the topic words co-occur frequently, which means that the topic is semantically coherent.

$$C_{UCI}(T) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{PMI}(w_i, w_j) \tag{7}$$

Where:
- $T = (w_1, w_2, \ldots, w_n)$ denotes a topic consisting of n keywords.
- $\text{PMI}(w_i, w_j)$ denotes the Pointwise Mutual Information between keyword pairs w_i and w_j.

The PMI is calculated as:

$$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_i)P(w_j)} \tag{8}$$

Notes:
- $P(w_i)$ denotes the probability of word w_i occurring in the corpus.
- $P(w_i, w_j)$ denotes the probability of w_i and w_j co-occurring in the same document.
- ϵ denotes a small constant added to avoid division by zero.

- **Coherence C_NPMI:** Measures the semantic coherence of topic words with normalized PMI. A high value of C_NPMI indicates that the words within the topic are more meaningful, thus making it easier to interpret the topic.

$$C_{NPMI}(T) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{NPMI}(w_i, w_j) \tag{9}$$

Where:
- $T = (w_1, w_2, \ldots, w_n)$ denotes a topic represented by n keywords.
- $\text{NPMI}(w_i, w_j)$ denotes the Normalized Pointwise Mutual Information between keyword pairs w_i and w_j.

The NPMI is calculated as:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)+\epsilon}{P(w_i)P(w_j)}}{-\log(P(w_i, w_j)+\epsilon)} \tag{10}$$

Notes:
- $P(w_i)$ denotes the probability of word w_i in the document collection.
- $P(w_i, w_j)$ denotes the probability that w_i and w_j co-occur in the same document.
- ϵ denotes a small constant added to prevent division by zero.

- **Overlap Quality Index:** Measures the reasonableness of the multi-topic classification which is done on the identical document. The calculation approach is based on the how alike documents with the same topics are. A high value of overlap quality index indicates that the result of multi-label classification is valid and indeed there is a relationship ascribed to meaning among many labels.

$$\text{OQI} = \frac{1}{|D'|} \sum_{d_i \in D'} \frac{1}{|C_{d_i}|} \sum_{c_j \in C_{d_i}} \text{cosine\_similarity}(d_i, c_j) \tag{11}$$

Where:
- D denotes the entire set of documents, where each document can belong to multiple topics.
- D' denotes the subset of documents that belong to at least two topics.
- For a document d_i that belongs to multiple topics, $C_{d_i} = \{c_1, c_2, \ldots, c_k\}$ denotes the set of centroid vectors corresponding to those topics.
- $\text{cosine\_similarity}(d_i, c_j)$ calculates the similarity between document d_i and topic centroid c_j.
- The cosine similarity is defined as:

$$\text{cosine\_similarity}(d_i, c_j) = \frac{d_i \cdot c_j}{\|d_i\| \|c_j\|} \tag{12}$$

- **Label Entropy:** Measures the uniformity of label allocation for various topics. This is captured through information entropy. Higher values in entropy suggest greater balance in the distribution of documents across multiple topics.

$$H = -\sum_{k=1}^{K} p_k \log(p_k) \tag{13}$$

Where:
- K denotes the total number of topics.
- n_k denotes the number of documents assigned to topic k.
- N denotes the total number of documents.
- $p_k = \frac{n_k}{N}$ denotes the proportion of documents assigned to topic k.

For all 100 calls, we examined our LLM-driven topic modeling technique along with the statistical techniques

while considering multiple datasets and a wide range of metrics for evaluation. In contrast with the traditional statistical approach, the LLM approach consistently generates clearer conceptualization with more precise topic boundaries, stronger semantic coherence within concepts, more balanced document distribution, and more interpretable descriptive labels alongside comprehensive characterization using the documents' multi-label assignments.

Metric	LLM	Traditional	Difference	Better Model
Overall Density	0.0501	**0.1652**	-0.1151	Traditional
Overall Distinctiveness	**0.4721**	0.2124	0.2597	LLM
Coherence C_V	**0.3564**	0.3317	0.0248	LLM
Coherence C_Uci	-0.3546	**-0.1743**	-0.1803	Traditional
Coherence C_Npmi	**-0.0133**	-0.0195	0.0063	LLM
Overlap Quality Index	**0.0864**	0.0279	0.0585	LLM
Label Entropy	**3.3249**	2.1890	1.1359	LLM

TABLE I: Comparison of LLM and Traditional Models Across Metrics on MLScholar

Metric	LLM	Traditional	Difference	Better Model
Overall Density	**0.0844**	0.0796	0.0048	LLM
Overall Distinctiveness	**0.1837**	0.0976	0.0861	LLM
Coherence C_V	**0.3687**	0.3640	0.0048	LLM
Coherence C_Uci	**-0.4966**	-0.8452	0.3486	LLM
Coherence C_Npmi	**-0.0349**	-0.0394	0.0045	LLM
Overlap Quality Index	0.0781	**0.0795**	-0.0014	Traditional
Label Entropy	**2.6301**	2.2125	0.4176	LLM

TABLE II: Comparison of LLM and Traditional Models Across Metrics on APT-Diverse

Metric	LLM	Traditional	Difference	Better Model
Overall Density	0.0840	**0.0900**	-0.0060	Traditional
Overall Distinctiveness	**0.1237**	0.1049	0.0188	LLM
Coherence C_V	**0.2993**	0.2865	0.0128	LLM
Coherence C_Uci	**-0.4897**	-0.6791	0.1894	LLM
Coherence C_Npmi	**-0.0497**	-0.0593	0.0096	LLM
Overlap Quality Index	0.0289	**0.0758**	-0.0469	Traditional
Label Entropy	**2.5411**	2.2407	0.3004	LLM

TABLE III: Comparison of LLM and Traditional Models Across Metrics on APT-Technical

A. Density and Distinctiveness

Across the various datasets, available density metrics — used to measure the cohesiveness of documents within topic clusters — show imbalanced results. For the three datasets, MLScholar (0.1652 vs 0.1249) and APT-Technical (0.0900 vs 0.0839), the traditional approach demonstrated higher density, while the LLM strategy outperformed in APT-Diverse (0.0839 vs 0.0796). This indicates that in some cases, traditional methods give a better aggregate measure of document clusters.

When measuring the separation struggle between distinct topics, distinctiveness metrics show a considerably better outcome from the LLM approach. Particularly impressive results were observed in the MLScholar (0.4182 vs 0.2124) and APT-Diverse (0.2959 vs 0.0976) notable subsets. This emphasizes the LLM's capability of generating greater semantic understanding of content as it creates more differentiated topics.

B. Topic Coherence

In relation to coherence, the LLM approach proved to be more effective across all three datasets. For instance, in the MLScholar dataset, LLM-based clustering achieved a CV Coherence score of 0.3564, compared to 0.3317 offered by the traditional approach. Further extensive discrepancies were noted in the APT-Diverse dataset, where the UCI Coherence was drastically better for the LLM approach (-0.4966) than the traditional method (-0.8452). In the APT-Technical dataset, the LLM approach also showed effectively better coherence in all three metrics than the traditional method.

The magnitude of coherence improvement varied notably across dataset types, with the APT-Diverse dataset showing the largest difference. This suggests that the LLM approach is particularly effective at maintaining topic coherence when handling documents with varied formatting and reporting styles. Interestingly, while the absolute values of NPMI Coherence metrics were relatively small across all datasets (ranging from 0.0133 to 0.0593), the LLM approach consistently achieved better scores, with improvements of 0.0063, 0.0045, and 0.0096 across the three datasets, respectively. The consistently superior performance on all three coherence metrics provides robust evidence that the LLM-based approach generates more semantically coherent topic clusters compared to traditional methods, regardless of domain or dataset homogeneity.

C. Label Entropy and Label Quality Analysis

While the traditional approach delivers higher density scores in two datasets (0.1652 vs 0.0501 in MLScholar and 0.0900 vs. 0.0840 in APT-Technical), the LLM method consistently achieves superior distinctiveness across all three datasets, with particularly significant margins in MLScholar (0.4721 vs. 0.2124). This pattern indicates that traditional clustering prioritizes tightly grouped documents, whereas LLM-based clustering creates more clearly delineated topic boundaries that minimize semantic overlap between categories. The superior distinctiveness and higher label entropy values of the LLM approach translate directly to practical advantages in information management systems. Well-separated topic clusters enable more precise information retrieval, clearer navigation pathways, and reduced ambiguity when categorizing new documents. The balanced information distribution confirmed by consistently higher entropy values (3.32 vs. 2.19) in MLScholar demonstrates that the LLM method avoids creating dominant categories that could obscure meaningful distinctions, ultimately delivering a more intuitive and usable document organization system aligned with how human experts conceptualize information domains.

V. Conclusion and Future Work

In this study, we developed an approach for clustering documents for intelligence information with the use of LLMs, particularly in the context of cybersecurity with the analysis of Advanced Persistent Threats (APTs). Our approach applies LLMs to capture important keywords and cluster intelligence documents according to the generated tags, which is a much more coherent and interpretable method than traditional clustering methods. It is observed from the results of extensive evaluation on multiple datasets that the LLM-based method outperforms traditional approaches with regards to topic coherence, distinctiveness, and label entropy. Moreover, the novel automated clustering pipeline enables a fully automated and scalable intelligence processing system that requires no human interaction, which is beneficial for cyber practitioners and researchers facing the challenge of working with large volumes of unstructured text documents.

Even with these excellent results, numerous problems still exist. Initially, while our LLM-driven clustering has enhanced interpretability, refinement of the strategies for label generation is required, particularly in terms of granularity and redundancy. Additionally, there is concern regarding computational efficiency as LLM inference is more costly in terms of resources when compared to traditional clustering systems. Future work may involve implementing the concept of real-time intelligence updates in the scope of clustering, where new information can be used for dynamic reclassification. Clustering could also benefit from the use of different LLMs, for example, RAG models. In the end, operational intelligence analysis will help us understand how our approach can be useful and what challenges it may bring.

These challenges will be solved in future work; hence, we expect to improve LLM-driven intelligence clustering solutions to be effective and scalable for processing enormous quantities of unstructured text data in the domains of security and intelligence.

Acknowledgment

This material is based upon work supported in part by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (SC-21). Prepared by LLNL under Contract DE-AC52-07NA27344 (LLNL-CONF-2001933). This material is also based upon work supported in part by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357. This research is supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. In addition, this material is supported in part by the National Security Agency under Award #H98230-20-1-0387, National Science Foundation under Award #2346701, and intramural funding from the Division of Research and Innovation at San Diego State University. We thank Jerry Sheehan from Salk Institute, Michael Farley, John Owens, and Christopher Leong from San Diego State University for their exceptional Information Technology support.

References

[1] L. Huangfu, W. Mao, D. Zeng, and L. Wang, "Occ model-based emotion extraction from online reviews," in *Proceedings of the 2013 IEEE International Conference on Intelligence and Security Informatics*, June 2013, pp. 116–121.

[2] Y. Sun, Q. Kong, L. Huangfu, and P. J., "Domain-oriented news recommendation in security applications," in *Proceedings of the 2021 IEEE International Conference on Intelligence and Security Informatics*, November 2021, pp. 1–6.

[3] R. Y. Maragheh, C. Fang, C. C. Irugu, P. Parikh, J. Cho, J. Xu *et al.*, "Llm-take: Theme-aware keyword extraction using large language models," in *Proceedings of the 2023 IEEE International Conference on Big Data (BigData)*, December 2023, pp. 4318–4324.

[4] K. Scarfone and M. Souppaya, "Data classification practices: Facilitating data-centric security management," in *Policy Commons*, 2021.

[5] M. Vantard, C. Galland, and M. Knoop, "Interdisciplinary research: Motivations and challenges for researcher careers," *Quantitative Science Studies*, 2023.

[6] Y. Liu, H. He, T. Han, X. Zhang, M. Liu, J. Tian *et al.*, "Understanding llms: A comprehensive overview from training to inference," *arXiv preprint*, 2024.

[7] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, "Text clustering with large language model embeddings," *International Journal of Cognitive Computing in Engineering*, 2025.

[8] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang *et al.*, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 6, pp. 1–32, 2024.

[9] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman *et al.*, "A comprehensive overview of large language models," *arXiv preprint*, 2023.

[10] W. Elouataoui, "Ai-driven frameworks for enhancing data quality in big data ecosystems: Error detection, correction, and metadata integration," *arXiv preprint*, 2024.

[11] C. M. Graham, "Ai skills in cybersecurity: global job trends analysis," *Information & Computer Security*, 2025.

[12] B. Chander and K. Gopalakrishnan, "Data clustering using unsupervised machine learning," in *Statistical Modeling in Machine Learning*. Academic Press, 2023, pp. 179–204.

[13] S. Sun, J. Li, Y. Dong, H. Liu, C. Xu, F. Li, and Q. Liu, "Multi-agent application system in office collaboration scenarios," *arXiv preprint*, 2025.

[14] J. K. Miller and T. J. Alexander, "Human-interpretable clustering of short text using large language models," *Royal Society Open Science*, 2025.

[15] Z. Fei, Y. Shao, L. Li, Z. Zeng, C. He, H. Yan *et al.*, "Query of cc: Unearthing large-scale domain-specific knowledge from public corpora," *arXiv preprint*, 2024.

[16] H. Jo, H. Lee, and T. Park, "Zerodl: Zero-shot distribution learning for text clustering via large language models," *arXiv preprint*, 2024.

[17] K. Blanda, "APTnotes," https://github.com/aptnotes/, 2016, [Online; accessed May 2025].

[18] CyberMonitor, "Apt & Cybercriminals Campaign Collection," https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections, 2023, [Online; accessed May 2025].

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *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)*, June 2019, pp. 4171–4186.

[20] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint*, 2022.

[21] S. V. Raju, B. K. Bolla, D. K. Nayak, and J. Kh, "Topic modelling on consumer financial protection bureau data: An approach using bert-based embeddings," in *Proceedings of the IEEE 7th International Conference on Convergence in Technology (I2CT)*, 2022, pp. 1–6.