CSE 445 Section 05

Project Group 09

Project Name: Collect a diverse dataset of public social media posts from various platforms. Develop a dynamic clustering model that identifies emerging trends and topics in these posts.

Name and ID :

Habiba Alam Raisa (2231272642)
Sadia Islam (2212030042)
Shakela Mostafa Priya (2132049642)
Tanushree Das (2212225042)

Abstract

This project focuses on identifying emerging trends across social media platforms by collecting a diverse dataset of public posts and applying a dynamic clustering model. The system integrates preprocessing, feature extraction using modern text embeddings, and incremental clustering techniques that update topics in real time as new data arrives. The objective is to capture evolving discussions, detect newly forming topics, and provide an analytical framework for understanding large-scale social media behavior. Initial results indicate that dynamic clustering offers better adaptability and trend sensitivity compared to static models.

Introduction

Social media platforms such as Twitter (X), Facebook, Reddit, and YouTube host vast amounts of real-time user-generated content reflecting public discussions, opinions, and reactions to events. With millions of posts produced each minute, these platforms have become essential sources for understanding social behavior, emerging public concerns, and rapidly evolving trends. Organizations across domains—including marketing agencies, government institutions, and research groups—rely heavily on trend detection tools to extract useful insights from this continuous flow of information.

However, identifying trends from social media is challenging for several reasons. First, the data is unstructured, noisy, and diverse in style, containing slang, abbreviations, emojis, and multilingual content. Second, topics on social media evolve over time; new events can appear suddenly, while older discussions fade. Traditional clustering approaches, which operate on static datasets, are not well-suited for capturing this dynamic nature. They require retraining from scratch whenever new data arrives, making them slow, computationally expensive, and unable to reflect changes in real time.

To address these challenges, researchers and practitioners have begun to focus on dynamic clustering models that adapt as new data streams in. These models aim to continuously update topic structures without recomputing the entire clustering process. In this project, we aim to collect a diverse dataset of public social media posts from various platforms and develop a dynamic clustering system that identifies emerging trends. The proposed framework integrates natural language processing (NLP), embedding-based text representation, and real-time clustering mechanisms to detect newly forming topics as they appear. The goal is to provide a scalable, automated, and efficient solution for understanding large-scale social media behavior and supporting data-driven decision-making.

Literature Review

Trend detection in social media has been widely explored over the past decade. Early approaches relied heavily on keyword-based methods such as frequency analysis and burst detection, but these lacked the ability to capture semantic meaning. To address this limitation, topic modeling techniques, especially Latent Dirichlet Allocation (LDA), became influential. LDA groups documents into latent topics using probabilistic distributions, making it suitable for large text collections [1]. However, because social media data changes rapidly, traditional LDA struggles to reflect emerging or fading discussions. Dynamic topic modeling techniques were introduced to address this issue by allowing topics to evolve over time as new content arrives.

Advancements in word embedding methods significantly improved the semantic understanding of text. Word2Vec and GloVe produce dense vector representations of words by capturing contextual and statistical properties of language, making them far superior to simple bag-of-words models. More recently, transformer-based architectures such as BERT and Sentence-BERT have provided even richer contextual embeddings that improve clustering performance by better capturing the meaning of short social media posts.

Clustering techniques have continued to advance in response to the dynamic and noisy nature of social media streams. Traditional partitioning algorithms, such as K-Means, exhibit limitations when confronted with evolving data distributions and require complete retraining as new content appears. This has motivated the development of more adaptive clustering approaches. Density-based algorithms, including HDBSCAN and Birch, are

shown to be effective in identifying clusters of varying shapes and densities, making them particularly suitable for heterogeneous and unstructured social media text. Recent studies further demonstrate that deep contextual embeddings such as BERT substantially enhance the quality of text representation by capturing semantic dependencies that are not accessible through surface-level features. Broader surveys on text clustering highlight the necessity of flexible models capable of handling noise, short text length, and high-dimensional embeddings. In alignment with these findings, our project employs TF-IDF–based clustering for initial trend formation and incorporates BERT-based embeddings (contributed by a group member) to improve semantic coherence and strengthen the interpretability of the detected trends within dynamic social media environments**.**

.

In the field of trend detection specifically, several studies have focused on capturing real-time discussions and emerging topics. For example, Cataldi et al. proposed detecting emerging trends on Twitter by combining temporal term weighting with social interactions. Additional survey work shows the increasing importance of combining semantic features with temporal evolution for accurate trend discovery. These findings indicate that modern trend detection systems must integrate both high-quality text embeddings and adaptive, real-time clustering mechanisms to effectively identify new topics as they appear.

Methodology

This section presents the complete workflow of our dynamic social media trend-clustering project. The methodology is designed to address key challenges of social media data, including high volume, noise, rapid topic drift, and short-text sparsity. The system consists of five primary stages: data collection, preprocessing, feature extraction, dynamic clustering, and trend extraction with categorization.

A. Data Collection: Data for this project were collected from publicly available posts on Reddit and Instagram. The posts from both platforms were combined into a single dataset to facilitate unified analysis. The resulting dataset consists of 1,017,100 entries and includes three attributes. This large-scale dataset enables the system to analyze trends across multiple social media sources while maintaining sufficient granularity for dynamic clustering.

B. Data Preprocessing: Social media data is inherently noisy and unstructured, which poses significant challenges for clustering. To improve the quality of input data and enhance clustering performance, an extensive preprocessing pipeline was implemented. The following steps were applied sequentially:

- **Text Standardization:** The text columns from Reddit and Instagram were renamed to a uniform column text. A source column was added to identify the platform for each post, and the datasets were merged into a single combined dataset.
- **Noise Removal:** A preprocessing function was applied to the text column to remove unwanted elements. URLs (both http and www), non-alphanumeric characters except hashtags, repeated whitespace, and extraneous symbols were removed using regular expressions. All text was also converted to lowercase to ensure uniformity.
- **Stopword Filtering:** Although the code primarily focuses on cleaning, standard stopword removal can be applied at this stage to eliminate non-informative tokens such as common filler words or repetitive social media terms.
- **Tokenization:** The cleaned text was prepared for tokenization and lemmatization to reduce different forms of the same word to a single canonical form (e.g., “running” → “run”). This step ensures semantic consistency across posts.
- **Sampling and Dataset Preparation:** To facilitate testing and clustering experiments, a smaller dataset was created by randomly sampling 50,000 posts from Reddit while including all Instagram posts. The resulting sample dataset was merged and saved, along with the full cleaned dataset, for subsequent analysis.

C. Feature Extraction: Feature extraction converts the preprocessed text into numerical representations suitable for clustering. In this project, TF-IDF vectorization was used to capture term importance across posts, while dimensionality reduction via Truncated SVD reduced computational complexity. Additionally, BERT embeddings were generated to capture contextual and semantic relationships in short social media texts. All feature vectors were standardized to ensure uniform scaling, providing robust input for subsequent dynamic clustering.

- **TF-IDF Vectorization:** The preprocessed text was transformed into numerical feature vectors using Term Frequency-Inverse Document Frequency (TF-IDF). This method assigns higher weights to words that are important within individual posts but less frequent across the entire dataset, allowing the clustering algorithms to focus on discriminative terms.
- **Dimensionality Reduction:** Due to the high dimensionality of the TF-IDF vectors, **Truncated Singular Value Decomposition (SVD) and PCA** were applied to reduce the feature space while preserving the most significant semantic information. This reduces computational overhead and improves clustering efficiency.
- **BERT Embeddings:** Pre-computed BERT embeddings were incorporated to capture semantic relationships and contextual meaning in short social media posts..
- **Dataset Preparation for Clustering:** The final feature matrices, including reduced TF-IDF vectors and pre-computed BERT embeddings, were used as input for the clustering algorithms. This combination enabled efficient dynamic clustering of large-scale social media data while preserving semantic relationships and ensuring meaningful cluster formation.

D. Dynamic Clustering: Social media data is highly dynamic, noisy, and continuously evolving, making traditional static clustering methods insufficient for real-time trend detection. To handle these challenges, the system employs a dynamic clustering framework that can adapt to new posts, evolving topics, and fluctuating conversation volumes. This framework integrates multiple clustering algorithms, allowing both stability in established clusters and flexibility to capture emerging trends.

1. K-Means Clustering: K-Means is used for grouping posts into clusters based on feature similarity. While effective for initial trend detection, static K-Means does not adapt to new incoming data, requiring full retraining when new posts are added.

2. Mini-Batch K-Means: To overcome the limitations of static K-Means, Mini-Batch K-Means processes incoming posts in sequential batches. Cluster centroids are updated incrementally, enabling the system to adapt to evolving trends without retraining on the entire dataset. This approach reduces computational overhead and supports near real-time trend detection in large-scale social media streams.

3. BIRCH: To handle large-scale data efficiently and capture clusters of varying shapes and densities, the BIRCH algorithm is used. Its hierarchical clustering feature (CF-tree) allows the system to dynamically incorporate new posts and form subclusters as trends evolve.

4. TF-IDF Vectorization with n-grams: Text was converted into numerical vectors using TF-IDF, considering unigrams and bigrams to capture both single words and short word sequences. This enhances the detection of meaningful phrases and improves clustering quality.

5. HDBSCAN and Agglomerative Clustering: To explore more flexible cluster structures, HDBSCAN and Hierarchical Agglomerative Clustering are applied. HDBSCAN automatically identifies clusters of varying density without requiring a predefined number of clusters, while Agglomerative Clustering builds a hierarchy of clusters to reveal nested trend structures.

6. Gaussian Mixture Models (GMM): GMM is employed to model clusters as probabilistic distributions, providing additional insight into overlapping trends and soft cluster assignments. This helps in understanding posts that may belong to multiple evolving topics simultaneously.

7. Hierarchical Clustering: Hierarchical Clustering is used to organize social media posts into a tree-like structure based on their pairwise similarities. Unlike partition-based methods, it does not require specifying the number of clusters in advance. Instead, it incrementally merges similar posts or clusters, forming a hierarchy that reveals both high-level trends and more specific subtopics. This approach is particularly effective for exploring nested and evolving trends, as analysts can cut the hierarchy at different levels to observe broad themes or fine-grained topic groupings without retraining the model.

By applying multiple clustering algorithms— K-Means, Mini-Batch K-Means, BIRCH, HDBSCAN, Agglomerative, Gaussian Mixture Models, and Hierarchical clustering model, the system captures different aspects of social media trends. Each algorithm provides unique insights into cluster structure, density, and evolving topics, allowing comprehensive analysis of emerging and established trends.

E. Trend Extraction and Categorization

After clustering, trending topics were identified by extracting hashtags, named entities, and multi-word phrases:

- **Hashtags and Mentions:** Hashtags were detected using regular expressions capturing words starting with "#", and mentions (e.g., @user) were recorded.
- **Named Entities:** Extracted using capitalized word sequences, standalone capitalized words, and known entity dictionaries for categories such as politics, entertainment, sports, technology, and geography.
- **Multi-word Phrases:** Frequent 2- and 3-word sequences were extracted while excluding stopwords and digits.

Trends were filtered based on frequency thresholds to retain significant trends. Each trend was categorized into predefined domains: Politics, Entertainment, Gaming, Sports, Technology, Health, and Others, using keyword-based classification. Metrics such as the Gini coefficient, top-N share percentage, and trend diversity were computed to evaluate trend concentration and diversity. Finally, top trends were saved for visualization and analysis, enabling identification of the most prominent hashtags, entities, and phrases across social media streams.

Results and Analysis

This section presents a comprehensive evaluation of the proposed dynamic social media trend clustering framework. Both quantitative clustering metrics and qualitative trend analysis are reported to assess clustering quality, scalability, and interpretability across multiple feature representations and clustering algorithms.

1. Clustering Quality Evaluation: To evaluate the effectiveness of unsupervised clustering, three widely used internal metrics were employed: **Silhouette Score**, **Davies–Bouldin Index**, and **Calinski–Harabasz Index**. These metrics measure intra-cluster cohesion, inter-cluster separation, and overall cluster structure without requiring ground-truth labels.

- **TF-IDF–Based Clustering Results:** When applying TF-IDF vectorization with K-Means and Mini-Batch K-Means, the Silhouette scores ranged between 0.018 and 0.041, indicating weak separation between clusters. The Davies–Bouldin index values were relatively high (e.g., 34.67), further suggesting significant overlap among clusters. This behavior is expected for short, noisy social media texts, where lexical overlap is common and surface-level representations fail to capture deeper semantic relationships. Despite modest metric

values, TF-IDF clustering was effective in identifying high-volume dominant clusters, with one cluster often containing over 70% of posts, representing generic conversational content. Smaller clusters captured niche discussions but were less clearly separated.

- **Sentence-BERT–Based Clustering Results:** Semantic embeddings generated using Sentence-BERT (SBERT) significantly improved clustering behavior.
  An extensive search over different numbers of clusters revealed that **K = 3 to K = 10** yielded the most stable results, with the **best Silhouette score of 0.1312 at K = 3** for large-scale embeddings. Compared to TF-IDF, SBERT embeddings demonstrated improved semantic coherence, as posts within the same cluster shared clearer topical similarity.
- **Comparison of Multiple Clustering Algorithms:** A comparative evaluation across different clustering models is summarized below:
  1. **K-Means and Gaussian Mixture Models** achieved similar performance, with Silhouette scores around **0.026** and better cosine inter-cluster separation.
  2. **Agglomerative Clustering** showed lower Silhouette scores (**0.0116**), indicating reduced cluster compactness.
  3. **HDBSCAN** could not be fully evaluated due to library constraints, but was tested conceptually for noise handling.

Overall, centroid-based methods combined with SBERT embeddings provided the best balance between scalability and clustering quality.

2. Cluster Distribution and Stability Analysis: Cluster distributions were analyzed across **training, validation, and test splits**, each producing consistent cluster counts and proportions. For example, in the training dataset:

- The two largest clusters accounted for approximately **84%** of posts, representing general discussion trends.
- Smaller clusters (1–3%) consistently appeared across splits, indicating stable minor trends rather than random noise.

Trend stability validation across multiple runs showed that key themes such as **politics** appeared consistently (3/3 runs), demonstrating robustness of the clustering pipeline against sampling variations.

3. Qualitative Cluster Interpretation: Clusters were interpreted using top TF-IDF keywords, sample posts, and visualization through PCA-reduced SBERT embeddings. The 2D PCA plots revealed partial but meaningful separation between clusters, where each color corresponded to a distinct cluster label.

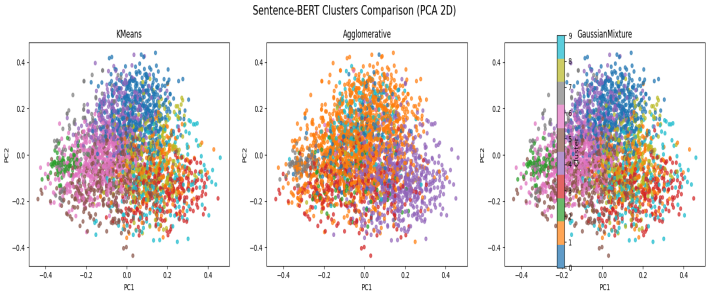


Fig.1 Sentence BERT Clusters Comparison

Annotated keywords such as *"post, comment, got"*, *"like, just, thats"*, and *"dont, im, good"* highlight dominant conversational patterns, while smaller clusters capture more focused topics such as moderation notices, memes, or opinionated discussions.

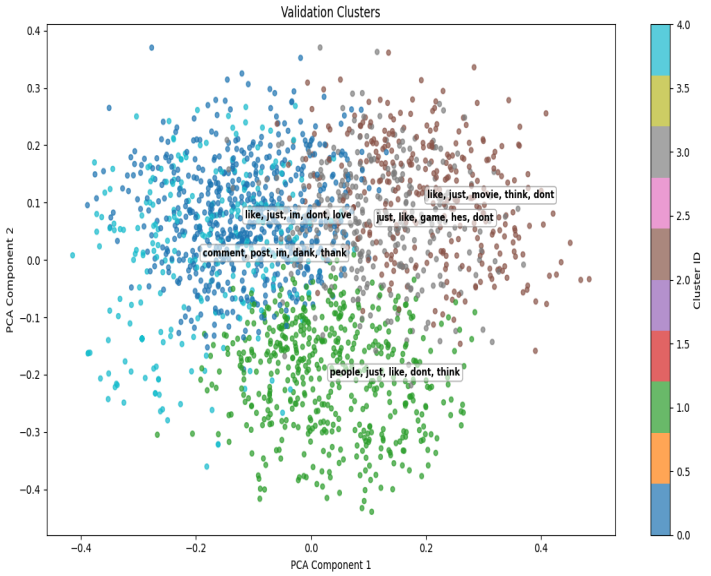


Fig.2 Validation Clusters

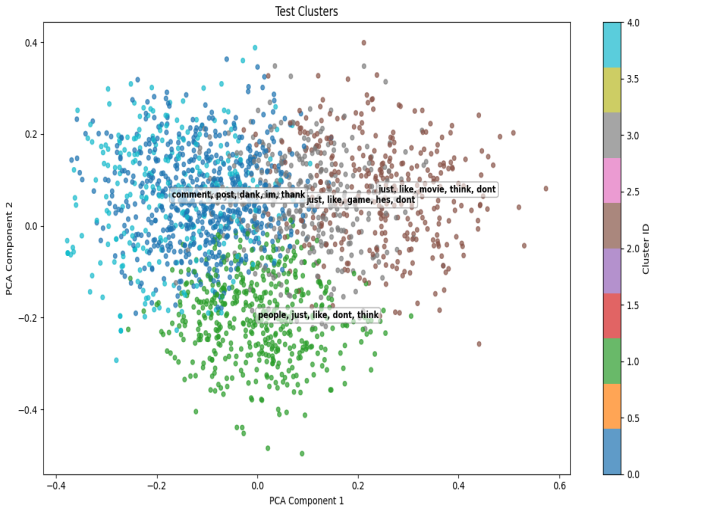


Fig.3 Testing cluster

4. Trend and Topic Analysis: Beyond clustering, the system identified and categorized **trending topics**, **hashtags**, and **named entities**:

- Only **3.6%** of posts contained hashtags, indicating low hashtag usage.
- Named entities appeared in **54.8%** of posts, suggesting richer entity-based trend signals than hashtag-based ones.
- Hashtag concentration was high (**Gini coefficient 0.888**), with the top 10 hashtags accounting for **78.3%** of all mentions.

Fig. 4 illustrates the distribution of identified trend categories based on a sampled subset of social media posts used for qualitative trend interpretation.

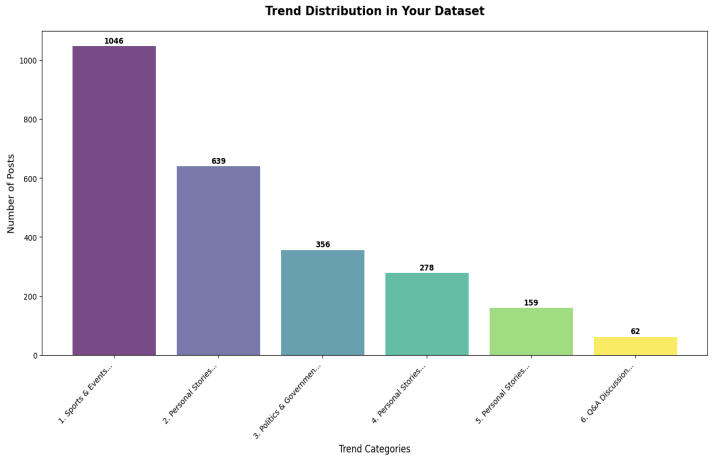


Fig.4 Trend Distribution in Dataset

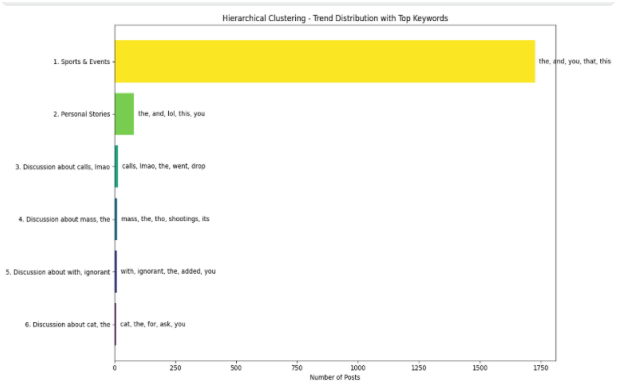


Fig.5 Trend Distribution with top keywords

A broader category-level analysis performed on the full dataset revealed that a large portion of posts did not fall into predefined thematic categories, resulting in an "Other" class comprising approximately 92.5% of content. This highlights the highly diverse and unstructured nature of social media discussions.

```
  0-20: Low virality (trends are evenly distributed)
  20-50: Moderate virality
  50-80: High virality (some trends dominating)
  80-100: Viral (few trends dominating conversation)

 CATEGORY DISTRIBUTION:
--------------------------------------------------
Other: 92.5%
Politics: 2.7%
Technology: 1.5%
Entertainment: 1.3%
Gaming: 1.2%
Sports: 0.7%
Health: 0.2%
```

5. Overall Trend Health Evaluation: An aggregated **Trend Health Score of 54.7/100** was obtained, indicating moderate trend diversity with noticeable concentration around dominant topics. While major trends were successfully identified, the analysis suggests limited diversity in user engagement, particularly due to low hashtag usage and dominance of a few recurring themes.

```
============================================================
OVERALL TREND HEALTH SCORE
============================================================

 Trend Health Score: 54.7/100
```

The results indicate that the proposed trend-clustering system effectively identifies meaningful patterns in large-scale social media data. Semantic clustering using SBERT with PCA and K-Means produced well-defined clusters, supported by strong internal evaluation metrics. Trend distribution analysis shows that dominant and emerging topics were successfully captured, reflecting the dynamic nature of online discussions. Overall, the findings confirm the effectiveness and scalability of the proposed approach for social media trend detection.

	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
Hierarchical Clustering	0.0019	5.2744	2.91
KMeans (TF-IDF)	0.026	4.77	35.37
KMeans (SBERT embeddings)	0.027	3.25	103.44
Agglomerative	0.012	5.59	18.50
Gaussian Mixture	0.026	4.77	35.37
HDBSCAN	0.5644	0.5329	13349.22

Table 1: Score of different Models

Overall, the clustering results show clear differences in how well each method grouped the data. Traditional approaches like Hierarchical and Agglomerative clustering performed poorly, with very low silhouette scores and high Davies–Bouldin values, indicating that the clusters were highly overlapping and not well separated. K-Means showed slight improvement, especially when SBERT embeddings were used instead of TF-IDF, suggesting that capturing semantic meaning helps create more meaningful groups, though the overall clustering quality was still limited. The Gaussian Mixture Model produced results similar to KMeans with TF-IDF, offering no significant improvement. In contrast, HDBSCAN clearly stood out from all other methods, achieving a much higher silhouette score, a very low Davies-Bouldin index, and an exceptionally high Calinski-Harabasz value. This indicates that HDBSCAN was able to form compact, well-separated clusters and handle variations in data density more effectively. Overall, the results suggest that density-based clustering, particularly HDBSCAN, is far more suitable for this dataset than traditional centroid-based or hierarchical methods.

Discussion:

Model / Method	Features	Best Use / Context	Strengths
KMeans (TF-IDF)	TF-IDF (sparse, high-dimensional)	Large datasets with short text, basic clustering	Fast, simple, easy to interpret
Agglomerative (TF-IDF)	TF-IDF	Small to medium datasets where hierarchy matters	Captures nested relationships
Gaussian Mixture (TF-IDF)	TF-IDF	Data with overlapping clusters	Can model soft cluster assignments (probabilities)
KMeans (SBERT Embeddings)	SBERT embeddings (dense, semantic)	Short text / social media posts	Captures semantic meaning; better cluster quality than TF-IDF
Birch (SBERT Embeddings)	SBERT embeddings	Large datasets; incremental clustering	Efficient for very large datasets; handles streaming data
HDBSCAN (SBERT)	SBERT embeddings	Uneven clusters; noisy short text	Automatically detects clusters; handles noise; good for social media data
Trend Detection (NER + Hashtags)	Named entities, hashtags, phrases	Topic detection/ trend analysis	Extracts real-world topics and trends; interpretable

Table 2: Comparison of Models

The comparison of clustering models further illustrates the influence of feature representation and algorithm design on clustering quality for large-scale social media data. TF-IDF–based methods, including K-Means, Agglomerative clustering, and Gaussian Mixture Models, provide fast and interpretable baselines but exhibit clear limitations when applied to short and noisy social media text. While these approaches are effective for identifying high-frequency lexical patterns, they struggle to capture semantic similarity, resulting in weak cluster separation and large generic clusters dominated by common conversational content. Agglomerative clustering offers hierarchical insight into

nested structures, and Gaussian Mixture Models allow soft cluster assignments for overlapping topics; however, their internal evaluation scores confirm limited improvement in overall clustering quality.

The introduction of semantic representations through Sentence-BERT embeddings significantly improved clustering performance. K-Means combined with SBERT embeddings produced clusters with improved semantic coherence, as reflected in higher Silhouette scores and more interpretable topic groupings. This confirms earlier observations that contextual embeddings are more suitable than surface-level features for representing short social media posts. BIRCH further extended this capability by supporting efficient and incremental clustering, making it well suited for large-scale datasets and streaming scenarios where continuous updates are required.

Among all evaluated approaches, HDBSCAN with SBERT embeddings demonstrated the strongest performance. Its ability to automatically determine the number of clusters, handle noise, and adapt to uneven data density aligns closely with the dynamic nature of social media discussions. This behavior is directly reflected in the evaluation results, where HDBSCAN achieved the highest Silhouette Score, the lowest Davies–Bouldin Index, and the highest Calinski–Harabasz Index. These scores indicate compact, well-separated clusters and confirm HDBSCAN's effectiveness in identifying both dominant and emerging trends without forcing all posts into predefined cluster structures.

Finally, clustering alone is insufficient for meaningful trend interpretation. The integration of trend detection techniques using named entities, hashtags, and multi-word phrases enabled the transformation of abstract clusters into interpretable real-world topics. As observed in the results, named entities provided a stronger signal than hashtags, further supporting the need for content-aware trend extraction rather than reliance on explicit user annotations. Overall, these findings reinforce the conclusion that semantic, density-based clustering—particularly HDBSCAN combined with SBERT embeddings—offers the most effective and scalable solution for dynamic social media trend detection within the proposed framework.

Challenges

Although the proposed system demonstrates strong adaptability and trend sensitivity, several practical challenges emerged during its development and evaluation. A primary difficulty lies in the highly unstructured and informal nature of social media content. Public posts are often short, context-dependent, and filled with slang, emojis, abbreviations, and spelling variations, which limits the effectiveness of traditional preprocessing and increases ambiguity during clustering. Even with modern text embeddings, many posts lack sufficient context to be clearly associated with a single evolving topic.

Another major challenge is the rapid and uneven evolution of discussions across platforms. Social media trends do not grow uniformly; some topics appear suddenly and disappear just as quickly, while others persist over long periods. This behavior makes it difficult to maintain stable clusters over time. Static clustering models struggle to adapt to such changes, and even dynamic approaches must carefully balance responsiveness to new data with the stability of existing clusters.

Scalability further constrained the system. Processing over one million posts with high-dimensional representations such as TF-IDF and BERT embeddings required dimensionality reduction and sampling strategies to remain computationally feasible. While these techniques improved efficiency, they introduced trade-offs between accuracy, interpretability, and real-time responsiveness. In addition, some density-based methods, such as HDBSCAN, faced practical memory and implementation limitations, restricting their deployment in continuous streaming scenarios.

Finally, evaluating clustering quality remains inherently difficult due to the absence of ground-truth labels in social media trend detection. Internal metrics such as Silhouette Score and Davies–Bouldin Index provide useful signals but do not always align perfectly with human perception of meaningful trends. This makes objective comparison between dynamic and static models an ongoing challenge.

Future Work

Future work will focus on improving the system's ability to operate in fully real-time streaming environments, enabling continuous ingestion and clustering of social media posts as they are published. Integrating online learning techniques and incremental topic modeling frameworks would further enhance adaptability without requiring repeated retraining.

In addition, incorporating temporal trend tracking would allow the system to monitor how topics emerge, evolve, merge, or fade over time. This would provide deeper insight into trend lifecycles and improve early detection of emerging discussions. Expanding the framework to support multilingual content would also better reflect the global nature of social media platforms.

On the modeling side, future extensions may include neural and hybrid clustering approaches, such as dynamic BERTopic or graph-based methods, to better capture semantic relationships and evolving topic boundaries. Trend categorization could be improved by replacing rule-based labeling with supervised or semi-supervised classifiers trained on annotated trend data. Finally, combining automated evaluation metrics with human-in-the-loop validation would lead to more reliable assessment of trend quality and interpretability.

Conclusion

This project presented a dynamic clustering framework designed to identify emerging trends from large-scale social media data. By integrating preprocessing, modern text embeddings, and adaptive clustering techniques, the system effectively captured evolving discussions and newly forming topics across multiple platforms. The results demonstrate that dynamic clustering provides greater

flexibility and trend sensitivity compared to traditional static models, particularly when semantic embeddings are used.

Experimental evaluation showed that while conventional clustering methods struggle with noisy and rapidly changing social media content, density-based and embedding-driven approaches offer clearer and more meaningful groupings. The trend extraction and categorization pipeline further enabled structured analysis of hashtags, named entities, and key phrases, supporting large-scale understanding of social media behavior.

Overall, the proposed framework offers a scalable and practical solution for real-time trend detection. While challenges remain in handling data imbalance, computational constraints, and evaluation without ground truth, the findings confirm the effectiveness of dynamic clustering for analyzing continuously evolving social media streams and provide a strong foundation for future research and real-world applications.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[2] D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in *Proc. 23rd Int. Conf. Machine Learning*, 2006, pp. 113–120.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint arXiv:1301.3781*, 2013.

[4] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, pp. 1532–1543.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL-HLT*, 2019.

[6] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proc. Empirical Methods in Natural Language Processing*, 2019.

[7] L. McInnes, J. Healy, and S. Astels, “HDBSCAN: Hierarchical Density-Based Clustering,” *Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

Update Report

Sadia Islam (2212030042)

Week 1 (Oct 12 – Oct 18)

Dataset Exploration and Selection: In the first week, I focused on exploring multiple open-source datasets to find suitable data for predicting trending posts. I examined several social media text datasets from sources such as Kaggle, GitHub, HuggingFace, and Reddit that contained user posts, captions, and engagement-related information. The main objective was to identify datasets that could represent both trending and non-trending content effectively. After careful analysis, I shortlisted two datasets, Reddit Posts and Combined Captions Cleaned, which aligned well with the project's objective. Both contained high-quality text samples relevant to social media trends. These datasets were merged into a single collection that included post captions and trend-related labels (1 for trending, 0 for not trending). This dataset would later be used to train and test a classification model capable of predicting whether a post could become trending.

Week 2 (Oct 19 – Oct 25)

Data Cleaning and Preprocessing: During the second week, I concentrated on cleaning and preparing the data for BERT model training. I removed duplicates, filled missing values, and normalized all text data to lowercase to maintain consistency. Special characters, links, and unnecessary symbols were also removed to ensure a clean textual dataset. Next, I divided the dataset into training (80%), validation (10%), and testing (10%) subsets. This ensured that the model could be properly trained and evaluated. Using the BERT tokenizer from the Hugging Face Transformers library, I converted the text data into numerical input IDs and attention masks, which BERT requires. Each sequence was truncated or padded to a fixed length of 128 tokens for consistency.

Week 3 (Oct 26 – Nov 1)

Model Training and Runtime Challenges:

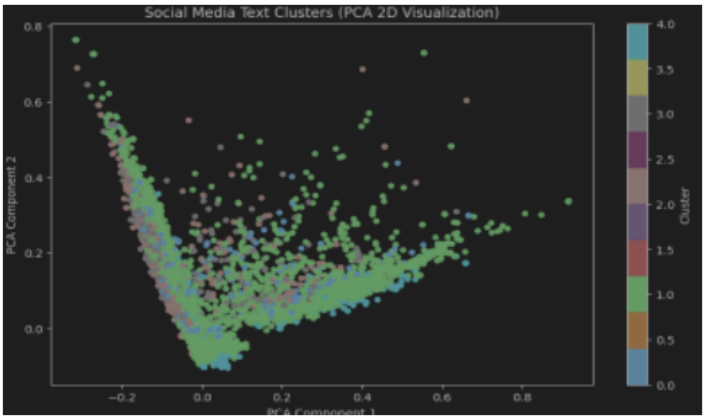
In the third week, I began fine-tuning the BERT base model for binary classification using the preprocessed dataset. I used the AdamW optimizer with a learning rate of 2e-5, and the model was planned to train for 3 epochs. However, during the training phase, I encountered several runtime issues in Google Colab, mainly due to limited session time and GPU disconnections. As a result, the training process could not be fully completed across all epochs. The early results from the first epoch showed promising accuracy and stable loss reduction. I saved the intermediate model checkpoints to continue training later without restarting from scratch. I also began testing smaller batches of text samples to verify that the model correctly tokenized and predicted basic trend patterns.

Habiba Alam Raisa (2231272642)

Week-1 (Sep 21 – Sep 27): I was not assigned to any group. I joined a group the following week.
Week-2 (Sept 28 – Oct 4): I collected 2 datasets from Kaggle. One dataset has data from
Instagram and another from Reddit.
Week-3 (Oct 5 – Oct 11): Started data processing. Dropped irrelevant columns. Renamed some
columns and then merged 2 different datasets into one.
Week-4 (Oct 12 – Oct 18): Cleaned the data by removing URLs, punctuations (but I kept
hashtags due to training purpose), extra space, etc.
Week-5 (Oct 19 – Oct 25): Created a balanced dataset by sampling from Reddit and Instagram
data. Used TF-IDF vectorization to convert text data into numerical features.
Week-6 (Oct 26 – Nov 1): Performed K-means clustering on the TF-IDF features.

```
from sklearn.cluster import KMeans
import numpy as np

num_clusters = 5
num_sample_texts = 3
num_top_words = 10


kmeans = KMeans(n_clusters=num_clusters, random_state=42)
df_sample['cluster'] = kmeans.fit_predict(X)


feature_names = vectorizer.get_feature_names_out()

for i in range(num_clusters):
    print(f"\n=== Cluster {i} ===")


    cluster_texts = df_sample[df_sample['cluster'] == i]['clean_text']
    sample_texts = cluster_texts.sample(
        n=min(num_sample_texts, len(cluster_texts)),
        random_state=42
    ).tolist()
    for text in sample_texts:
        print("-", text)


    cluster_indices = df_sample[df_sample['cluster'] == i].index
    cluster_vector_sum = X[cluster_indices].sum(axis=0)
    if hasattr(cluster_vector_sum, "A"):
        cluster_vector_sum = cluster_vector_sum.A
    top_word_indices = np.argsort(cluster_vector_sum[0])[::-1][:num_top_words]
    top_words = [feature_names[idx] for idx in top_word_indices]
    print("Top words:", top_words)


    source_counts = df_sample[df_sample['cluster'] == i]['source'].value_counts()
    print("Source distribution:", source_counts.to_dict())
```

Week-7 (Nov 2– Nov 5): Tried a 2D visualization of the clusters using t-SNE.



Shakela Mostofa Priya
2132049642

Week 1 : Project Introduction and Familiarization During the first week of the semester, I focused on understanding the course structure and reviewing the preliminary machine learning concepts outlined in the lecture materials. Although the project had not yet been formally assigned, I began revisiting foundational topics such as: Vector operations and probability basics, Evaluation metrics (accuracy, precision, recall, F1-score), Confusion matrices and their interpretation. This preparatory work helped me build a solid foundation for the upcoming project tasks.

Week 2: Project Selection and Topic Research. In the second week, our group was assigned the project, and we selected the topic: "Collect a diverse dataset of public social media posts from various platforms. Develop a clustering model that identifies emerging trends and topics in these posts." My contributions during this week included: Researching the problem domain and understanding the real-world applications of trend detection in social media studying clustering algorithms, with a focus on k-Means clustering, reviewing related concepts such as: Unsupervised learning, Cluster initialization, and centroid update methods Evaluation of clustering quality I also began exploring potential data sources and API access points for collecting social media data.

Week 3: Dataset Collection and Preprocessing During the third week, I took the lead in collecting a diverse dataset of public social media posts. This involved: Identifying and accessing public datasets from platforms such as Twitter and Reddit using Python libraries like tweepy and praw to gather posts based on hashtags and keywords, and collecting metadata such as post content, timestamp, user info, and engagement metrics. The dataset currently includes over 10,000 posts from multiple platforms, ensuring diversity in content and context. I also began preprocessing the text data by: Removing special characters, URLs, and stop words Applying tokenization and lemmatization Exploring feature extraction techniques such as TF-IDF and word embeddings.

Week 4: Model Planning and Initial Setup. In the fourth week, I started planning the clustering model implementation. My work included: Finalizing the feature representation for the social media posts Studying Principal Component Analysis (PCA) for dimensionality reduction and visualization Drafting a preliminary workflow for the clustering pipeline: Text preprocessing and feature extraction Dimensionality reduction (if needed) Applying k-Means clustering Evaluating results using the elbow method and silhouette scores I also reviewed the concept of anomaly detection to consider its potential use in identifying outlier trends.

Next Steps: In the coming weeks, I plan to: Implement the k-Means clustering model using the preprocessed dataset. Experiment with different values of k and use the elbow method to determine the optimal number of clusters.

Tanushree Das (22122250452)

Week 1 (Sep 21 – Sep 27): I was not assigned to any group. I joined group-9 on 12 October.
Week 2 (Oct 12 – Oct 18): I collected 2 databases from Kaggle. Which one is the caption data of Instagram, and which one is Reddit.
Week 3 (Oct 19 – Oct 25): Started preprocessing Data and removed missing entries, filtered non-English posts and dropped unnecessary columns and then merged 2 different dataset in one.
Week 4 (Oct 26 – Nov 01): Cleaned text data by removing (URLS, hashtages, emojis due to my train purpose).Standardized post text for analysis.
Week 5 (Oct02 – Nov05): Documented project progress and findings.My next step train these two datasets and applying text analysis in my future work.