Exploring the Application of Clustering on News Headlines with Pre-trained Language Model

Shiyu Sun
dept. School of Computer Science
Nanjing University of Posts and Telecommunications
1024040824@njupt.edu.cn

Abstract

In the real world, the volume of news data is enormous, making it impractical to rely solely on human effort for classification due to the incalculable workload involved. The advent of advanced text mining algorithms and pre-trained models has opened up new possibilities for rapid and efficient unlabelled text clustering. This experiment aims to investigate the feasibility of using clustering algorithms to extract meaningful information from news headlines, leveraging the transformative power of pre-trained models to enhance the process.

Introduction

In the current digital era, the real-time generation of massive news data has become a prominent characteristic. The rapid proliferation of online news platforms, social media, and digital content has led to an unprecedented volume of information being produced every second. Traditional manual classification methods, which rely on human annotators to categorize news articles, are not only inefficient and costly but also prone to bias due to subjective factors.

With the advancement of Natural Language Processing (NLP) technologies, particularly the development of pre-trained models such as BERT(Devlin 2018), and other transformer-based architectures, breakthrough solutions for unsupervised text clustering have emerged. These advanced techniques can effectively capture semantic features from unstructured text, enabling machines to understand the context and meaning of news articles without the need for labeled data. By leveraging the power of pre-trained models, it is possible to significantly enhance the accuracy and scalability of text clustering, thereby providing a practical technical pathway for large-scale news data analysis. This is particularly important in applications such as news recommendation systems, trend analysis, and real-time event detection.

This study integrates pre-trained language models such as BERT with clustering algorithms like K-Means(MacQueen et al. 1967) to construct an efficient news text clustering framework. The proposed framework is designed to address the challenges of handling high-dimensional text data and improving clustering performance in an unsupervised setting. Our methodology consists of three main stages:

- Utilizing pre-trained models to convert text data into coarse-grained document embeddings. This step involves transforming raw text into numerical vectors that capture the semantic meaning of the content, enabling downstream clustering algorithms to process the data effectively.
- Applying the K-means algorithm to perform clustering on the document embeddings. K-means is chosen for its simplicity and efficiency, although other clustering algorithms can also be explored depending on the specific requirements of the task.
- Evaluating the clustering results using metrics such as the Silhouette Coefficient(Rousseeuw 1987), Calinski-Harabasz Index(Caliński and Harabasz 1974), and Davies-Bouldin Index(Davies and Bouldin 1979). These metrics provide insights into the quality of the clusters, helping to assess the effectiveness of the proposed framework.

Related Works

The Transformer(Vaswani 2017) is a deep learning model architecture, primarily used for natural language processing (NLP) and sequence-to-sequence tasks. Its core innovation lies in the introduction of the self-attention mechanism, which enables the model to consider all positions in the input sequence simultaneously, thereby better capturing semantic relationships. The Transformer extends this mechanism through multi-head attention, allowing the model to process different information subspaces in parallel. The model typically consists of multiple stacked encoder and decoder layers, each incorporating residual connections and layer normalization techniques to mitigate gradient-related issues during training. Since the Transformer inherently lacks sequential position information, positional encoding is used to represent the order of words in the input sequence. This architecture has made it highly effective in sequence-to-sequence tasks such as machine translation.

BERT (Bidirectional Encoder Representations from Transformers) leverages a Transformer-based neural network to understand and generate human-like language. Unlike the original Transformer architecture, which includes both encoder and decoder modules, BERT adopts an encoder-only architecture, emphasizing the understanding of input sequences rather than generating output sequences. The BERT model undergoes a two-step process: pre-training on large amounts of unlabeled text to learn contextual embeddings, followed by fine-tuning on labeled data for specific NLP tasks. During pre-training, BERT learns contextual word representations by considering the surrounding context in sentences, utilizing unsupervised tasks such as Masked Language Modeling (MLM) to predict missing words or understanding relationships between sentences. After pre-training, the model is fine-tuned using labeled data tailored to specific NLP tasks like sentiment analysis, question answering, or named entity recognition. This fine-tuning process adapts BERT's general language understanding to the nuances of targeted applications, optimizing its performance for task-specific requirements.

The K-means algorithm is an iterative clustering analysis algorithm whose core idea is to partition n objects in a dataset into K clusters, minimizing the sum of distances from each object to the center (centroid) of its assigned cluster. Typically, the Euclidean distance is used as the metric, though other distance measures can also be employed. The algorithm iteratively optimizes the clustering results, ensuring that objects within the same cluster are as compact as possible, while objects in different clusters are as separated as possible. This optimization process is usually based on an objective function, such as the Sum of Squared Errors (SSE), which measures the total sum of distances from all objects to their respective cluster centers, continuously adjusting the centroids to achieve an optimal partition.

Methodology

Corpus Processing

The process of BERT converting news headlines into embeddings can be described as follows:

- **Representation:** For each news headline, BERT first converts it into a format that the model can process:
  - Tokenization: The news headline is segmented into words or subword units using the WordPiece algorithm. For example, the headline ”Crude oil prices stalled as hedge funds sold: Kemp” might be split into ”Crude”, ”oil”, ”prices”, ”stalled”, ”as”, ”hedge”, ”funds”, ”sold”, ”:”, ”Kemp”.
  - Adding Special Tokens: A [CLS] token is added at the beginning of the headline (to represent aggregated information of the entire headline), and a [SEP] token is added at the end to indicate the conclusion of the headline.
  - Positional Encoding: Positional encoding is added to each token to preserve the order of words in the headline.
- **Embedding Layer:** After the above processing, each token is converted into a vector representation. BERT's embedding layer combines the token ID, positional encoding, and segment embedding to generate a comprehensive input embedding vector. The dimensionality of these embedding vectors is usually 768 (BERT-Base).
- **Transformer Encoder:** The input embedding vectors are fed into a multi-layer Transformer encoder for processing. The Transformer encoder consists of multiple identical layers, each containing a multi-head self-attention mechanism and a feed-forward neural network. The self-attention mechanism enables BERT to consider the contextual information of all tokens in the headline simultaneously, thereby generating context-dependent embedding representations. Through multiple layers of stacking, BERT can capture complex semantic and syntactic features in the headline.
- **Output Embedding:** After processing by the multi-layer Transformer encoder, the embedding vector of each token is updated to include contextual information. For the K-means clustering task, the embedding of the [CLS] token is typically used as the aggregated representation of the entire news headline. The [CLS] embedding contains the global semantic information of the headline, making it suitable for clustering analysis.

Clustering

Use the embedding vectors generated by BERT as input data. The execution of K-means is as follows:

- Set K Value: Set the number of clusters K in K-means to the number of categories in the original dataset. For example, if the original dataset has 5 categories, then K=5.
- Initialize Centroids: Randomly initialize K centroids, each representing the center of a cluster.
- Iterative Optimization:
  - Assignment Step: Assign each embedding vector to the cluster whose centroid is the closest.
  - Update Step: Recalculate the centroid of each cluster by taking the mean of all embedding vectors in that cluster.
  - Repeat the above steps until the centroids no longer change significantly or the maximum number of iterations is reached.

Evaluation

Datasets

GoogleNews[1]: is a processed dataset containing 38205 news headlines, covering 8 main categories such as business, entertainment, headlines, health, science, sports, technology, and world news.

TagMyNews[2]: is a processed dataset containing 18321 news headlines, mainly divided into 7 categories such as sports, world, US, sci-tech, entertainment, health, and business.

Evaluation of Indicators

Silhouette Score: The Silhouette Score is used to measure the compactness and separation of clustering results. It combines the intra-cluster distance (the average distance

[1] https://www.kaggle.com/datasets/crxxom/daily-google-news

[2] https://raw.githubusercontent.com/cskarthik93/TagMyNews-Classification

between a sample and other samples in the same cluster) and the inter-cluster distance (the average distance between a sample and the nearest cluster) to evaluate the quality of clustering.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance between sample and other samples in the same cluster (intra-cluster distance).$b(i)$ is the average distance between sample and all samples in the nearest cluster (inter-cluster distance).

Range from -1 to 1. A value closer to 1 indicates better clustering (samples are more tightly grouped within clusters and well-separated from other clusters).A value close to 0 suggests overlapping clusters.A negative value indicates that samples may have been assigned to the wrong clusters.

Calinski-Harabasz Index: The Calinski-Harabasz Index evaluates clustering quality by measuring the ratio of between-cluster dispersion to within-cluster dispersion. Higher between-cluster dispersion and lower within-cluster dispersion indicate better clustering.

$$CH = \frac{\text{SS}_B/(k-1)}{\text{SS}_W/(n-k)}$$

where SS_B is the between-cluster dispersion (the sum of squared distances between cluster centers and the global center).$SS_W$is the within-cluster dispersion (the sum of squared distances between samples and their cluster centers).k is the number of clusters.n is the total number of samples.

A higher value indicates better clustering.

Davies-Bouldin Index: The Davies-Bouldin Index evaluates clustering quality by measuring the ratio of within-cluster dispersion to between-cluster separation. A lower value indicates better clustering.

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{d(c_i, c_j)} \right)$$

where S_i is the average distance between samples in cluster and its center (within-cluster dispersion). $d(ci, cj)$ is the distance between the centers of clusters i and j (between-cluster separation).k is the number of clusters.

A lower value indicates better clustering.

Accuracy: First of all, we use Hungarian algorithm to solve the issue of label inconsistency by optimally matching the cluster labels to the true labels. The specific steps are as follows:

- Construct a Confusion Matrix: Calculate the confusion matrix between the clustering results and the true labels.Each element $C[i][j]$ in the confusion matrix represents the number of samples where the true label is i and the cluster label is j.
- Use the Confusion Matrix as a Cost Matrix: Input the confusion matrix into the Hungarian algorithm.The goal is to find a label alignment that maximizes the number of matched samples.
- Find the Optimal Matching: Use the Hungarian algorithm to find the optimal label alignment.
- Realign the Labels: Realign the cluster labels based on the results of the Hungarian algorithm.
- Calculate Evaluation Metrics: Use the realigned labels to calculate evaluation metrics

$$\text{Accuracy} = \frac{\text{Number of correctly predicted samples}}{\text{Total number of samples}}$$

The ”number of correctly predicted samples” in the formula refers to the number of samples where the aligned cluster labels match the true labels. The accuracy value ranges from [0,1], where a value closer to 1 indicates better agreement between the clustering results and the true labels.

Results

Table 1: Results of BERT+K-Means on TagMyNews and GoogleNews

datasets	Indicators			
	Silhouette	CH Index	DB Index	Accuracy
TagMyNews	0.032	515.041	3.844	0.345
GoogleNews	0.021	899.027	3.926	0.349

The K value is set to be the same as the original number of labels.The results can be seen on Table 1

Based on the Silhouette Score and Davies-Bouldin Index, the clustering performance of the TagMyNews dataset is slightly better than that of the GoogleNews dataset.

According to the Calinski-Harabasz Index and Accuracy, the clustering and classification performance of the GoogleNews dataset is marginally superior to that of the TagMyNews dataset.

Overall, the clustering performance for both datasets is suboptimal, suggesting that further optimization of the clustering algorithm or data preprocessing steps may be necessary to improve clustering quality.

Improvement

Dimensionality Reduction

PCA(Wold, Esbensen, and Geladi 1987) is a linear dimensionality reduction technique that projects high-dimensional data into a lower-dimensional space through orthogonal transformation, preserving the directions with the highest variance in the data.

It achieves dimensionality reduction by computing the covariance matrix and eigenvectors of the data. BERT-generated embeddings are typically high-dimensional (e.g., 768 dimensions), and PCA can reduce them to a lower dimension for analysis. PCA can remove noise and redundant information from the embeddings, retaining the most important features. PCA is computationally fast and suitable for large-scale datasets.

UMAP(McInnes, Healy, and Melville 2018) is a nonlinear dimensionality reduction method based on manifold learning theory, assuming that data lies on a low-dimensional manifold. It maps high-dimensional data to a lower-dimensional space by optimizing both local and global data structures.

UMAP can better preserve the local and global structures in BERT embeddings, making it suitable for visualizing high-dimensional embeddings (e.g., in 2D or 3D). UMAP can reveal potential clustering structures in the embeddings, helping to understand semantic similarities in text data. UMAP can be used to extract low-dimensional features for subsequent machine learning tasks.

PCA performs well in preserving linear structures but may not capture complex nonlinear relationships in BERT embeddings.UMAP excels in revealing underlying structures and clusters in embeddings but comes with higher computational costs.

Table 2: Results of BERT+K-Means on TagMyNews and GoogleNews with dimensionality reduction

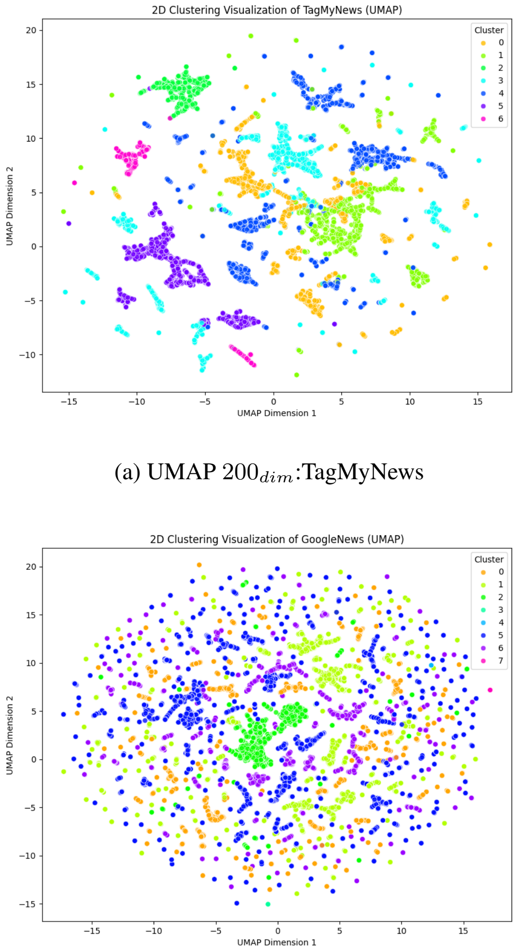
datasets	Silhouette	CH Index	DB Index	Accuracy
approach&dim		UMAP200		
TagMyNews	0.258	**4857.497**	**1.184**	0.428
GoogleNews	**0.305**	12990.679	1.106	0.397
approach&dim		UMAP500		
TagMyNews	0.250	4213.335	1.225	**0.449**
GoogleNews	0.304	**13993.360**	**1.169**	**0.453**
approach&dim		PCA200		
TagMyNews	-0.014	66.504	9.294	0.439
GoogleNews	-0.029	129.256	8.689	0.475
approach&dim		PCA500		
TagMyNews	-0.005	26.377	15.075	0.398
GoogleNews	-0.041	51.252	13.741	0.453
		Original		
TagMyNews	**0.032**	515.041	3.844	0.345
GoogleNews	0.021	899.027	3.926	0.349

The results can be seen on Table 2. The UMAP method outperforms the PCA method on most metrics, particularly excelling in the Silhouette Score and Calinski-Harabasz Index. The PCA method shows decent performance in certain cases, such as Accuracy, but falls short in cluster compactness and separation compared to UMAP. The clustering performance of the original data (without dimensionality reduction) is relatively poor, especially in terms of the Silhouette Score and Davies-Bouldin Index.

We visualized the clustering results after dimensionality reduction using a two-dimensional plot Fig 1, but the results were not very clear. This may be because the original data had a high dimensionality, and forcing it into a lower-dimensional space could disrupt the original structure. In other words, the data might exhibit good clustering performance in the original high-dimensional space, but this structure could be lost when projected into a lower-dimensional space.

Fine-tuning

Fine-tuning is a technique for further training a pre-trained model for a specific task or dataset. This method usually uses a smaller training dataset to ”fine-tune” the model, adapting it to the needs of a specific task after the pre-trained model has already learned a large number of common features from a broad and diverse corpus. Fine-tuning leverages the general knowledge embedded in the pre-trained model, allowing it to specialize in a particular domain or task without requiring extensive computational resources or time. This approach is particularly valuable in scenarios where labeled data is scarce or expensive to obtain, as it enables the model to achieve high performance even with limited task-specific data. Through fine-tuning, the model can quickly adapt to new tasks based on its pre-trained knowledge without the need to train from scratch. This not only saves significant computing resources and time but also ensures that the model retains the robust generalization capabilities acquired during pre-training. Fine-tuning allows the model to utilize the general linguistic patterns and semantic understanding learned during pre-training, while simultaneously adjusting its parameters to better align with the specific characteristics of the target task. This dual advantage makes fine-tuning an essential strategy for improving the performance of down-



(a) UMAP 200_{dim}:TagMyNews

(b) UMAP 200_{dim}:GoogleNews

Figure 1: UMAP Dimensionality Reduction

stream tasks, such as text classification, sentiment analysis, or named entity recognition.

Fine-tuning is particularly effective for tasks with limited labeled data. Even when the available dataset is small, the pre-trained model has already mastered a wide range of common features and patterns from its initial training on large-scale datasets. This prior knowledge enables the model to generalize well to new tasks, even with minimal additional training. For example, in the context of news text clustering, fine-tuning a pre-trained model like BERT on a small dataset of labeled news articles can significantly enhance its ability to capture domain-specific semantics and improve clustering accuracy. By fine-tuning, the model can better understand the nuances of the target domain, such as the specific terminology and contextual relationships unique to news data, leading to more accurate and meaningful clustering results.

Table 3: Comparisons between Bert and fine-tuning model on TagMyNews and GoogleNews

datasets	Silhouette	CH Index	DB Index	Accuracy
model		all-mpnet-base-v2		
TagMyNews	0.023	259.668	5.261	**0.566**
GoogleNews	0.031	465.608	5.211	**0.693**
model		Bert-base-uncased		
TagMyNews	0.032	515.041	3.844	0.345
GoogleNews	0.021	899.027	3.926	0.349

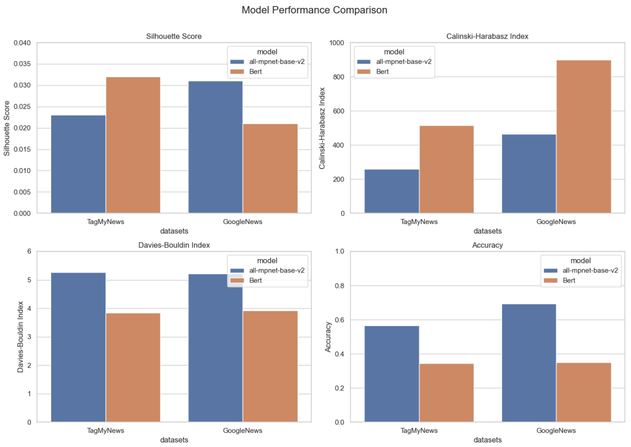


Figure 2: Comparisons between Bert and fine-tuning model on TagMyNews and GoogleNews

Results can be seen in Table 3 and Fig 2. Bert performs better in terms of clustering effectiveness, particularly in the Calinski-Harabasz Index and Davies-Bouldin Index metrics. all-mpnet-base-v2[3][4] demonstrates superior performance in classification accuracy, especially on the GoogleNews dataset. When selecting a model, the choice should be based on the specific task requirements to determine which model is more suitable. We also conducted comparisons on 200_{dim} and 500_{dim} using UMAP, and the results were identical, so they will not be reiterated here.

Conclusion

In this study, we explored the feasibility of leveraging pre-trained language models, specifically BERT, in conjunction with clustering algorithms like K-Means to automate the classification of unannotated news data. The proposed framework demonstrated the potential of combining advanced NLP techniques with traditional clustering methods to handle the challenges posed by the massive news data. By converting unstructured text into semantic-rich embeddings using BERT, we were able to effectively capture the contextual meaning of news headlines, enabling downstream clustering algorithms to process the data efficiently.

Our experiments revealed that the integration of BERT and K-Means provided a scalable and practical solution for unsupervised text clustering. However, the clustering performance varied across datasets, with the TagMyNews dataset showing slightly better results in terms of the Silhouette Score and Davies-Bouldin Index, while the GoogleNews dataset performed marginally better in the Calinski-Harabasz Index and accuracy. These results suggest that while the framework is effective, further optimization of clustering algorithms and data preprocessing steps may be necessary to improve overall clustering quality.

The application of dimensionality reduction techniques, such as UMAP and PCA, highlighted the importance of feature extraction in enhancing clustering performance. UMAP, in particular, outperformed PCA in preserving the local and global structures of the embeddings, leading to better clustering results. This underscores the value of selecting appropriate dimensionality reduction methods based on the specific characteristics of the data and the task at hand.

Overall, this study underscores the transformative potential of pre-trained models in unsupervised text clustering, offering a technical pathway for large-scale news data analysis. Future work could focus on fine-tuning pre-trained models for specific datasets, exploring alternative clustering algorithms, and incorporating additional contextual information to further enhance clustering accuracy and scalability. By addressing these areas, we can continue to advance the field of automated text classification and unlock new possibilities for real-time news analysis and information management.

References

Caliński, T.; and Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1): 1–27.

Davies, D. L.; and Bouldin, D. W. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2): 224–227.

[3] https://www.sbert.net/docs/sentence_transformer/pretrained_models.html#original-models

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20: 53–65.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3): 37–52.