MailOrg

C.J.Sakshi, Mohammed Ismail, Pema Tshering Sherpa, Rakshitha K
School of Computer Science and Engineering, RV University, Bangalore, India
USN: 1RVU23CSE125, 1RVU23CSE280,1RVU23CSE333, 1RVU23CSE369

Abstract — **In today's digital landscape, managing a continuously growing volume of emails is a persistent challenge. Users often encounter cluttered inboxes filled with repetitive or similar content, which can reduce productivity and lead to inefficient storage usage. Traditional methods for filtering and organizing emails typically rely on manually defined rules or basic heuristics. These approaches often lack the sophistication needed to recognize deeper contextual or semantic similarities and fail to summarize content effectively for quick decision-making.**

To solve this problem, we propose MailOrg, a smart, machine learning-driven system built to automatically sort emails into meaningful groups based on their content, extract relevant details, and generate brief, informative summaries. The system integrates powerful NLP techniques, such as BERT-based embeddings, with clustering methods like K-Means to group emails that share similar themes. It also assists users in managing their inboxes by offering options to archive, delete, or prioritize emails based on the generated summaries. Testing on a self-curated dataset comprising more than 12,000 authentic emails demonstrates that MailOrg can significantly enhance inbox management and storage optimization by automating the clustering and summarization processes.

Index Terms — **Email clustering, summarization, NLP, BERT, K-Means, ROUGE, BART, inbox optimization, storage management, user decision support.**

I. Introduction

As digital communication grows exponentially, managing email storage efficiently has emerged as a necessity. Many people are struggling to look through the details present in every mail.

Inboxes are often overloaded with repetitive or semantically similar emails, causing disorganization and leading to wasted storage and decreased productivity. Traditional approaches to email classification are strongly based on static keyword filters or custom rules, and adhere to adaptability and scalability.

Recent advances in natural language processing (NLP) and unsupervised machine learning provide an opportunity to address these challenges by categorizing and summarizing the mails. In the project, we provide our model with unstructured raw email texts which the model goes through and clusters it into different groups and then summarizes every mail to minimum two lines and displays it to the users so that they can go through the details and decide whether they should delete , save or keep it starred.

Hence the user will be able to organize their mails in a better manner and also have it sorted according to their requirement.

II. Related Work

Email classification and summarization have been extensively studied within the domain of Natural Language Processing (NLP), with a primary focus on enhancing information retrieval and user efficiency. Carenini et al. introduced the Clue Word Summarizer (CWS), which utilizes significant terms to generate summaries of email threads. Their model was benchmarked against baseline systems such as MEAD and RIPPER using the Enron email dataset, demonstrating the role of domain-specific keywords in improving summary consistency and coherence.

In a more recent development, Mahira et al. proposed ShortMail, a comprehensive summarization system that leverages BERT for semantic representation, K-Means for clustering, and ranking algorithms to prioritize important emails. Although the dataset used in their study was not disclosed, the approach proved successful even with unstructured email data.

Additionally, Ghulam Mujtaba et al. presented a survey of email classification methods, analyzing models like Support Vector Machines (SVM), Naive Bayes, K-Nearest Neighbors (K-NN), and Decision Trees. Their review noted classification accuracies reaching up to 87% across various datasets, while also acknowledging challenges in model scalability and adaptability.

These studies establish the utility of combining machine learning with NLP for email management, yet they also emphasize the need for a unified system that performs both semantic grouping and summarization with actionable decision support in real time.

III. Methodology

A. Dataset

To train the model, We created a custom dataset called NewFinal. This contains 12,486 emails collected by three separate email accounts. Data records were split into standard ratios of 80:20.

Training Set: 9,988 emails
Testing Set: 2,498 emails
Classes: 10 major Semantic categories

This data record contains different types of emails sent from different types of sources, making it the best data record to work with.

B. *Natural Language Processing Architectures*

We evaluate the integration of transformer-based models and unsupervised clustering algorithms for textual data classification:

1. BERT (Bidirectional Encoder Representations from Transformers):

BERT, created by Devlin et al. Google AI Language, is a noteworthy development in the architecture of natural language processing. By using Transformer's bidirectional training methodology, BERT is able to understand the contextual relationships between textual words. With 12 transformer blocks, 768 hidden dimensions, and 12 self-attention heads, the pre-trained model has 110 million parameters. The ability of BERT to produce contextually nuanced vector representations that capture semantic links within textual data sets it apart from the competition. This makes it especially useful for document vectorization jobs where proper classification depends heavily on context.

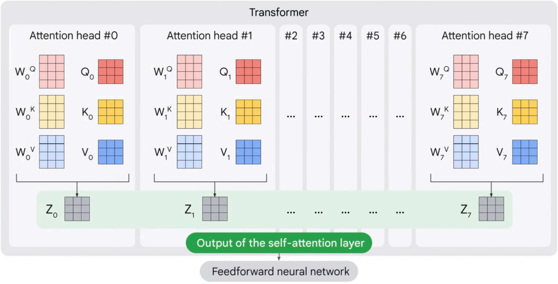


Fig. 1. Visual representation of BERT architecture

2. Document Vectorization Process:

Textual documents may be converted into high-dimensional numerical representations using BERT, which is used in the document embedding process. After being tokenized, each mail is converted into a 768-dimensional vector space representation and processed via BERT's attention mechanism. This conversion successfully captures contextual connections and semantic properties of documents. Subsequent clustering algorithms can detect organic groups based on theme or topical similarity since the generated document vectors maintain the closeness between semantically related content inside the high-dimensional feature space.

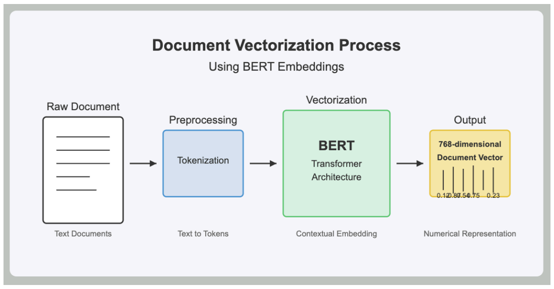


Fig. 2. Visual representation of MailOrg architecture

3. K-Means Clustering:

With the help of the unsupervised machine learning method K-Means, n observations are divided into k clusters, each of which is associated with the cluster that has the closest mean. Data points are assigned to the closest centroid via the iterative process, which then recalculates centroids using the cluster assignments as of right now. When applied to document vectors produced by BERT, K-Means clustering makes it easier to automatically classify texts according to their semantic similarity. Although the intended number of clusters must be pre-specified, the algorithm's computational efficiency makes it appropriate for large-scale document categorization applications.

4. BART Model:

An automatic summarization step aimed at reducing the complexity of email bodies for subsequent analysis was performed with the use of the Hugging Face transformers library. This involved the use of the *sshleifer/distilbart-cnn-12-6* model developed specifically for summarization tasks through the pipeline("summarization") API. Every email body was examined for length, and in order to preserve meaning, texts shorter than thirty words were not included in summaries. To make sure the model wouldn't go over the token limit, the first 1000 words of larger texts were included. Summarization parameters—maxlength and minlength—were adjusted to input size in order to cater to summarize-focused description requirements. This was performed on the Body column of the dataset with summarised text placed in a new column marked as "Summary." The obtained DataFrame was then formatted as a CSV file to enable further external processing for additional clustering and topic modeling tasks.

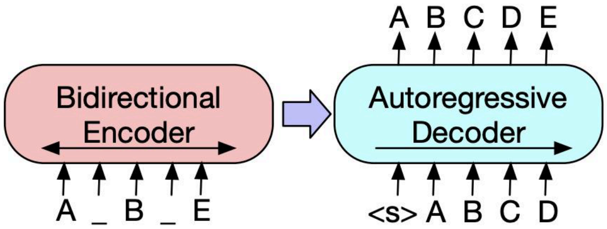


Fig. 3. Visual representation of AlexNet architecture

5. **Implementation Architecture:**

Preprocessing the documents, vectorizing them using BERT embeddings, and then clustering them using K-Means comprise the implementation architecture's sequential pipeline. Tokenization, normalization, and input sequence preparation that complies with BERT's specifications are all part of the preprocessing step. BERT's transformer design is used in the vectorization step to produce high-dimensional embeddings. In order to enable automated classification without supervised training, the clustering stage uses K-Means to find natural groupings within the document vector space.

C. *Training*

K-Means is used to cluster document vectors produced by BERT as part of the training process:

1. **Vector Extraction:**

   The [CLS] token from BERT's final hidden state is used to transform each document into a 768-dimensional vector.

2. **BART Model (summary):**

- To make email bodies simpler, an automated summary phase was implemented using Hugging Face's *sshleifer/distilbart-cnn-12-6* model.
- Emails with less than 30 words were ignored.
- Texts that were too long were condensed to the first thousand words.
- For additional analysis, summaries were created and saved in a new Summary column.

3. **K-Means Instruction:**

- The predicted categories in the dataset are used to determine the number of clusters (K).
- K-means++ is used to initialize centroids.
- Centroids are regenerated until convergence, and documents are repeatedly allocated to the closest centroid.

4. **Assessment:**

   Clustering quality is confirmed by:
   - For coherence and separation, use the silhouette score.
   - The Davies-Bouldin Index measures how compact a cluster is.
   - Internal consistency by inertia.

5. **Application:**

   By embedding fresh documents with BERT and allocating them to the closest cluster, the trained model classifies them.

IV. Implementation

We use K-Means for clustering and BERT for vectorization in the implementation of the document categorization system. The following technical procedures are part of the methodology:

1. **Preprocessing:** WordPiece tokenizer is used to tokenize documents, and then specific tokens ([CLS] and [SEP]) that are needed by BERT are added. To provide the model with information on token significance, documents are padded or shortened to a consistent length and attention masks are created.
2. **Vectorization:** BERT's transformer architecture is applied to the preprocessed documents. As a complete representation of the full document in a 768-dimensional vector space, we extract the [CLS] token's final hidden state.
3. **Dimensionality Reduction:** To lower computing complexity while maintaining the crucial semantic linkages between texts, optional dimensional reduction techniques like Principal Component Analysis (PCA) or t-SNE may be used.
4. **Clustering:** The document vectors are subjected to the K-Means method with the following settings:
   - *Method of initialization:* k-means++
   - Euclidean distance is a measure of distance.
   - *The number (k) of clusters:* Arranged according to anticipated document categories
   - *Criteria for convergence:* movement of the centroid below the threshold or maximum number of iterations
5. **Evaluation:** To determine the quality of the clustering, the resulting clusters are assessed using internal validation metrics including the silhouette coefficient, Davies-Bouldin index, and inertia.

V. Results

In evaluating the performance of each clustering and summerisation of mails, we utilize a set of comprehensive metrics to assess various aspects of model effectiveness and efficiency.

Clustering:

1. **Silhouette Score:**

- The silhouette coefficient in k-means clustering measures the similarity of a data point within its cluster (cohesion) compared to other clusters (separation).
- The equation for calculating the silhouette coefficient for a particular data point:

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

S(i) is the silhouette coefficient of *i*th sample.

a(i) is the average distance between *i*th sample and all the other data points in the cluster to which *i* belongs.

b(i) is the average distance from *i*th sample to all clusters to which i does not belong.

2. **Davies-Bouldin Index (DBI):**

- *Goal:* Lower is better — smaller DBI indicates better clusters (i.e., compact and well-separated).

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

where **Rij:**

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

k: Number of clusters
Si: Average distance between each point in cluster ii and the centroid of cluster ii
Mij: Distance between the centroids of clusters ii and jj
For each cluster ii, find the ***most similar*** *(least separated) cluster jj, and compute the ratio above.*
DBI is the average of those "worst-case" similarities.

3. Calinski-Harabasz Score:

- The Calinski-Harabasz Score is a clustering validation metric that evaluates how well the clusters are separated and how close they are.

$$\text{Calinski – Harabasz Index} = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} * \frac{N - k}{k - 1}$$

n: Total number of samples
kk: Number of clusters
Tr(Bk): Trace of between-cluster dispersion matrix (how spread the cluster centroids are)
Tr(Wk): Trace of within-cluster dispersion matrix (how compact the clusters are)

Summarization:

1. ROUGE scores (Recall-Oriented Understudy for Gisting Evaluation) :

These are the standard metrics used to evaluate text summarization and text generation by comparing the output with reference texts.
ROUGE-1 → unigrams (single words)
ROUGE-2 → bigrams (two-word sequences)
ROUGE-L → longest common subsequence (LCS)

ROUGE-N F1:

$$\textbf{F1:} \quad \frac{\textbf{2*(Precision*Recall)}}{\textbf{Precision+Recall}}$$

$$\textbf{Precision:} \quad \frac{\textbf{Number of overlapping unigrams}}{\textbf{total unigrams in generated summary}}$$

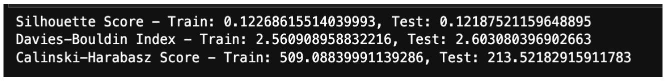
$$\textbf{Recall:} \quad \frac{\textbf{Number of overlapping unigrams}}{\textbf{total unigrams in reference summary}}$$

VI. Result Analysis

The experiments were conducted on Jupyter notebook with M3 10 Core GPU and Google Colab with T4 GPU acceleration where it is needed. The evaluations were done in the domain of unsupervised clustering and summarization quality. The results span across two major experimental designs:
1. Clustering Evaluation using BERT embedding.
2. Summary evaluation using word-embedding.

Clustering Evaluation using BERT Embeddings:
In the *mlpro* notebook we applied BERT embeddings on textual data followed by dimensionality reduction using PCA and clustering via KMeans Clustering. The quality of clustering was assured using well-known techniques such as :

1. Silhouette Score:
2. Davies-Bouldin Index:
3. Calinski-Harabasz Score:



```
Silhouette Score - Train: 0.12268615514039993, Test: 0.12187521159648895
Davies-Bouldin Index - Train: 2.560908958832216, Test: 2.603080396902663
Calinski-Harabasz Score - Train: 509.08839991139286, Test: 213.52182915911783
```

Fig. 4. Clustering Scores

From the results we received we determine that our model with a silhouette score near to 0 tells us that we have some overlapping clusters which is expected since our model uses emails which can be similar in nature.

For the *Davies-Bouldin Index* the scores we received suggest that the clusters are somewhat dispersed and not highly distinct. Typically a score near 0 indicates compact, well-separated clusters, so we still have room for improvement here.

The *Calinski-Harabasz Score* is also used to check the compactness and separation of the clusters. the higher the better the clustering is. Though the drop to 213 on the test set indicates that the model may not generalize as well to unseen data.

The clustering performance is modest as of now. While the BERT embeddings capture some structure the low silhouette score and high Davies-Bouldin index score suggest that the clusters are overlapping to some extent. the drop in Calinski-Harabasz score from the train to test further indicates that there is reduced generalization

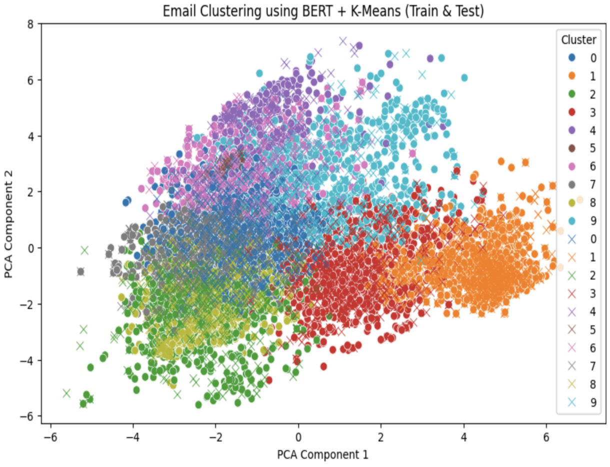


Fig. 5. Scatter plot for clusters using KMeans (k=10)

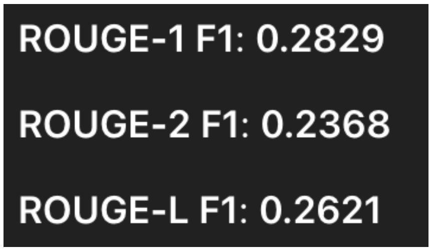
Summarization Evaluation:

In the second notebook we test performance based on custom metrics computed from word-overlap between original and summarized texts. these include:

1. Precision
2. Recall
3. F1 Score : 0.3086
4. Accuracy (Word overlap ratio) : 0.1896
5. Rouge Scores
    a. ROUGE-1 F1
    b. ROUGE-2 F1
    c. ROUGE-L F1

Taking the F1-score and accuracy values into account, the values indicate a moderate level of content retention where summaries capture some relevant information from the original text. However, the relatively low accuracy shows that a significant portion of original content is not reflected in the summaries which can be improved through the use of a better summarizer model.

To receive the rouge score we have taken 3 csv files that we extracted after summarizing the body sections. Taking the *summarized_latest_data.csv* we got the following rouge scores:



ROUGE-1 F1: 0.2829

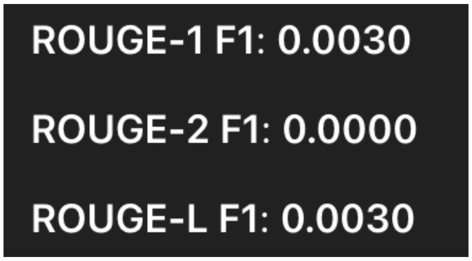
ROUGE-2 F1: 0.2368

ROUGE-L F1: 0.2621

Fig. 6. Summarizing Score -1

The following scores suggest that the generated summaries contain a reasonable overlap with the original texts at both the unigram and bigram levels. Rouge-L also confirms that longer sequence matches are present, although not strongly.

Now for two csv for which we extracted the data month wise and not daily data we got the following values:



ROUGE-1 F1: 0.0030

ROUGE-2 F1: 0.0000

ROUGE-L F1: 0.0030

Fig. 7. Summarizing Score -2

The values we got for these indicate that there is almost no meaningful overlap between the original and summarized content. This suggests a failure in the summarization process for these datasets. As currently our model is made to give daily summarized mails and not a month at a time we have diverted our focus to make the daily summarized scores better so we can get cak to this at a later date.

The summarization model performs moderately well on the latest dataset, retaining essential information, but **fails to generalize** to the monthly dataset. This discrepancy highlights the importance of **dataset quality and diversity** in training robust summarization systems. Further fine-tuning or preprocessing might improve performance consistency across datasets.

Overall Insights:

Overall our project explored two key tasks: unsupervised clustering using BERT embeddings and text summarization evaluation using BART. Both experiments provided valuable insights into the strength and limitations of the model we have used.

From the values we got from our models the following are the takeaways we have received:

- The BERT-based clustering was effective but needs more refinement to improve the separation and generalization
- Summarization models show promise on recent data but lack adaptability across datasets.
- Our project emphasizes the importance of data quality, model tuning, and cross-data evaluation for achieving reliable NLP performance.

VII. Conclusion

As digital communication continues to expand, the need for smart email management solutions becomes increasingly critical. *MailOrg* offers a practical response to this challenge by utilizing machine learning and natural language processing to automatically group emails with similar content and summarize key information. By incorporating models like BERT for semantic understanding and K-Means for clustering, the system efficiently organizes inboxes and supports informed user actions such as deleting, saving, or prioritizing emails. Our testing on a custom dataset of over 12,000 emails confirms that *MailOrg* can reduce clutter, improve user experience, and optimize email storage through automation. The ability to present essential content in a simplified form empowers users to manage their inboxes more effectively. While the current system performs well, further refinements—such as advanced sub-clustering and improved summarization—can enhance its capabilities. Overall, *MailOrg* demonstrates strong potential as a modern tool for efficient, scalable, and user-friendly email management.

VIII. Acknowledge

We would like to express our sincere gratitude to our esteemed professors, Dr. Shabber Basha S H, Prof.Achala Hegde from the School of Computer Science and Engineering at RV University.

IX. References

[1] G. Carenini, R. Ng, and X. Zhou, "Summarizing emails with conversational cohesion and subjectivity," *Proceedings of the 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2007. [Online]. Available: http://students.lti.cs.cmu.edu/11899/files/email-p91-carenini.pdf

[2] M. Bouazizi et al., "ShortMail: An End-to-End Email Summarization Framework Using Deep Learning," *AI Open*, vol. 5, pp. 12–23, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665963823000805

[3] A. Nenkova and K. McKeown, "Automatic Summarization," *IEEE Access*, vol. 5, pp. 160–177, 2017, doi: 10.1109/ACCESS.2017.2702187.

[4] Hugging Face, "Hugging Face – The AI Community Building the Future," [Online]. Available: https://huggingface.co/