Semantic Exploration and Retrieval-Augmented Generation over Enron Data

Team01 Milestone II Project Final Report

(Anandita Bodas, Gary Schaumburg, Nathan Goldhardt)

Table of Contents

Introduction

Our goal was to explore machine learning techniques for identifying and suggesting actions based on email content, relevant to legal or customer support scenarios. Given the challenge of accessing real email and supporting manual datasets, we used Enron[1] documents as a proxy for a knowledge base and the email corpus as a stand-in for customer interactions. We aimed to evaluate and identify question topics that could come via any channel, including chatbots. Our unsupervised topic exploration and supervised email content identification simulate tasks data scientists might perform in legal or customer operations.
The SEC documents provide a basis for topic formation related to the Enron scandal, while the emails contain both general content and topics covered in the case documents, although imbalanced. Stakeholders include managers and customers in legal and support contexts.
Our supervised learning focuses on categorizing email content, while our unsupervised scope explores topic identification and modeling. We also consider retrieval augmented generation (RAG)[2] as a future extension for unsupervised learning, leaving detailed evaluation for subsequent projects.

Data Sources

From Milestone 1:

Name	Description	Size	Links
PDF Documents	Four SEC case documents detailing the charges against defendants that include Enron executives, attorneys, and audit firm lead. The documents are divided into section and paragraph groups that indicate the lower level paragraph topics. "comp18776": SEC Complaint against Lay, Skilling, and Causey "comp18435": SEC Complaint against Delainey "comp20441": SEC Complaint against Duncan "comp20058": SEC Complaint against Mintz and Rogers	comp18776: 954KB comp18435: 102KB comp20441: 5.98MB comp20058: 603KB	https://www.sec.gov/files/litigation/complaints/comp18776.pdf https://www.sec.gov/files/litigation/complaints/comp18435.pdf https://www.sec.gov/files/litigation/complaints/2008/comp20441.pdf https://www.sec.gov/files/litigation/complaints/2007/comp20058.pdf
Email Corpus	This dataset was collected and prepared by the CALO Project. It contains data from about 150 users, mostly senior management of Enron, organized into folders. The corpus contains a total of about 0.5M messages. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. The dataset does not include attachments, and some messages have been deleted as part of a redaction effort due to requests from affected employees.	Full tarred gzipped file: 1.7GB	https://www.cs.cmu.edu/~enron/enron_mail_20150507.tar.gz

Newly added for Milestone 2:

- Carnegie Mellon University labels on Enron emails for supervised learning scope
- Enron Annual Reports to test large scale ingestion for RAG
- Enron-focused accounting textbook and two case studies for RAG
    - Accounting/finance Lessons Of Enron: A Case Study by Harold Bierman, Jr. Hardcover | Barnes and Noble®
    - (PDF) Enron: The Good, The Bad, The Lessons
    - Enron Case Study (pdf)

Part A. Supervised Learning

For our supervised learning portion, we chose to use the Enron emails labeled by CMU students as our dataset. The use case for these labeled emails is to identify collaborative emails to narrow down the scope of legal discovery. Out of the 12 categories, we chose the 2 most relevant to our use case:
0: Company Business, Strategy, etc. [hereafter referred to as General Business]
1: Document editing/checking [hereafter referred to as Document Collaboration].

Pre-processing and Exploration

Our dataset is highly imbalanced, with a distribution of 855 emails belonging to Class 0 and 135 emails belonging to Class 1.

Text data is inherently noisy, and there is also scope for additional human error. Our pre-processing and data clean-up steps included:

1. Stop Words Removal: We eliminated common stop words using SpaCy's library, including the word 'Enron', as it added no value to the context of the emails.
2. Extra Spaces and New Line Removal
3. Name Replacement: We replaced all names in the emails with the word 'person' to reduce vocabulary size and avoid bias.

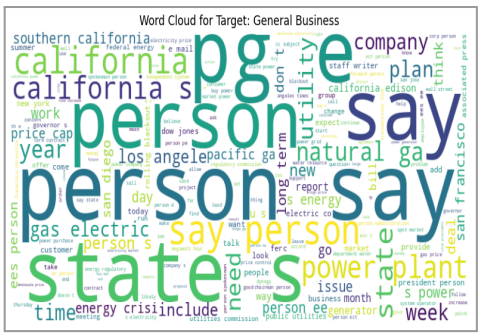

Fig A.1.1: Word Cloud for General Business

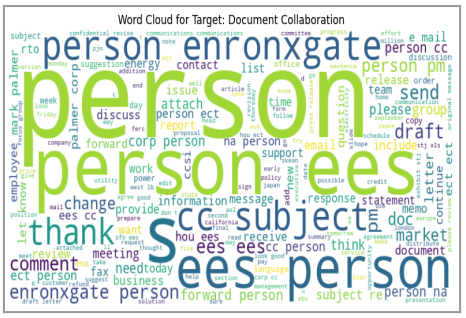

Fig A.1.2: Word Cloud for Collaborative Emails

Our initial exploration revealed a significant difference in the overall length of emails between the two categories. General Business emails were typically longer than Document Collaboration emails [Fig A.2]. The word clouds reveal that Document Collaboration emails contained a higher percentage of directive verbs like 'discuss', 'draft', 'schedule' and 'prepare' as compared to General Business emails [Figs. A.1.1 and A.1.2]. However, in terms of absolute numbers, these words weren't present in larger numbers within Document Collaboration emails.

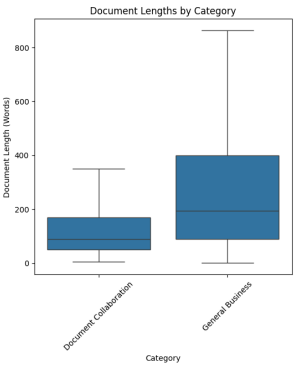

Fig A.2: Document lengths by Class

Building the Vectorizer and Embeddings

We used a TF-IDF vectorizer for our feature extraction due to its ability to highlight the importance of words within and across emails. While we also experimented with GloVe embeddings and SpaCy's vectorizer, we found that TF-IDF provided comparable results with faster computation times. Therefore, TF-IDF was finalized for its efficiency and effectiveness in our context. This vectorizer contains the vocabulary used to create the embeddings for our model. The size of this vocabulary is over 250k terms.

Calculating Metrics

Since we find it more important to identify all emails than to correctly identify them, we prioritize recall over precision. In all, we will be using accuracy, precision and recall to evaluate the performance of our model. To ensure a fair assessment of our metric scores, we will be using a stratified k fold cross validation strategy, where k=5. This ensures that each split at each iteration maintains the original 6:1 ratio of our data, providing a representative evaluation of our model's performance.

Building a Baseline Model

We built baseline models of the following types:

Model	Type	Accuracy	Precision	Recall
Logistic Regression	Probability Based	0.864646	0.527505	0.422222
Multinomial Naive Bayes	Probability Based	0.863636	0	0
XGBoost	Ensemble Based	0.871717	0.567146	0.266667
Decision Tree	Tree Based	0.846465	0.448941	0.377778
SVC	Instance Based	0.868686	0.616666	0.088888
Random Forest	Tree Based	0.865656	0.557142	0.096296
Bagging Using Logistic Regression Models	Ensemble Based	0.857575	0.48388	0.24444
Boosting Using Logistic Regression Models	Ensemble Based	0.80101	0.36054	0.58518

Observations

- We began with a good accuracy overall, owing to the high proportion of Class 0 data in the training set. However, our precision and recall metrics were generally low across the board, which is also attributable to this class imbalance.
- Of the models, Logistic Regression performs better. We think it is due to its inherent ability to handle binary classification tasks as well as high dimensional data like text.
- Naive Bayes may have performed poorly due to the assumption of independence between the features.
- Decision Tree and XGBoost performed similarly, and we believe it's due to their ability to handle non linear relationships.
- We also built an LSTM model but it wasn't much better than the other models due to the small size of the dataset.

After the baselining process, we chose to go ahead with Logistic Regression and Decision Tree Classifiers. Despite XGBoost's comparable performance, we favored Decision Tree for its interpretability and faster training and prediction times.

Improving the Model's Performance

- Synthetic Oversampling: We attempted SMOTE and K means SMOTE for various proportions using the library imblearn to address class imbalance. However, this only reduced the recall value further. This reduction might be because the synthetic samples generated by these methods failed to capture the semantic representation of the textual data accurately. Other unsuccessful attempts also included undersampling and random oversampling.

- Using Word Length as a Feature: We had initially observed a significant difference in word lengths between the two classes [See Fig A.2]. To leverage this observation, we horizontally concatenated the word length feature to the TF-IDF vectors of the emails. We chose not to scale the word length values to maintain the natural differences in length. Incorporating this feature resulted in a noticeable improvement in the recall value of our baseline models. We therefore chose to add this as a feature to our final model.
- We refrained from using the sender and receiver details as a feature in order to generalize the model and not introduce bias.

Hyperparameter Selection

The previous steps attempted to improve the model's performance by fine-tuning the dataset and feature selection. We now improved the performance by hyperparameter tuning.
To do so, we performed Grid Search to try a variety of hyperparameters on the Logistic Regression and Decision Tree models.
We finalized on the Logistic Regression model for its better recall value.

Model	Accuracy	Precision	Recall
Logistic Regression	0.744444	0.310916	0.711111

The model achieved an increase in recall (from 0.42 to 0.71), which was our primary goal. However, this improvement in recall came at the cost of reduced accuracy and precision.

Through various trials, we found that recall was most significantly influenced by the class_weight hyperparameter [Fig A.4]. Balanced weights lent to an improved accuracy.

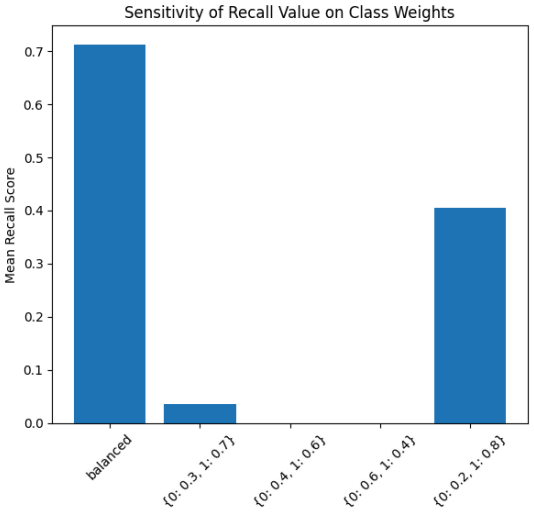

Fig A.4: Sensitivity of class weights w.r.t. recall

Feature Importance

Our feature set is the vocabulary in TF-IDF along with word length. The top 10 features are the words that contribute the most to the differentiation of the classes in the model [Fig A.3].

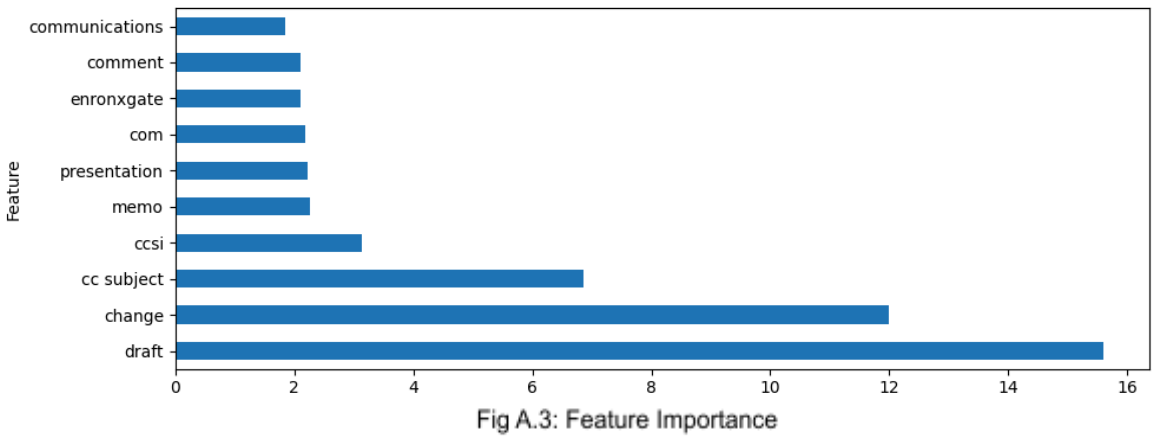

Fig A.3: Feature Importance

Additional Efforts to Improve the Metrics

In our attempt to improve the recall, we'd compromised significantly on precision. The major contributor to this was the imbalance in the dataset. Since SMOTE hadn't worked, we decided to use a library designed specifically for NLP augmentation called nlpaug. The library has a Synonym Augmenter that uses WordNet to replace a proportion of words in an input with its synonyms. We chose to perform minority oversampling. For every 1 sentence belonging to Class 0, we augmented 3 more, thereby reducing the imbalance in the dataset [Fig A.5].

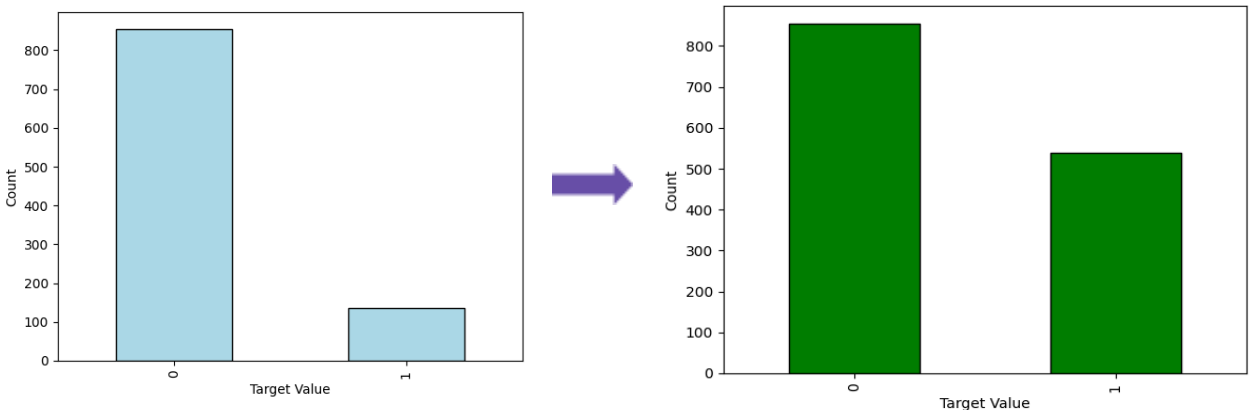

Fig A.5: Class distributions before & after using Synonym Augmenter

We concluded with more balanced precision-recall values, and a good improvement in the overall performance from the first baseline logistic regression model we'd built.

Model	Accuracy	Precision	Recall
Logistic Regression	**0.769176**	**0.686901**	**0.746296**

Common Sources of Error

- Email Length Dependency: The model heavily relied on the length of the emails, leading to misclassification of shorter General emails.
- Common Verbs Misclassification: Words frequently used in Document Collaboration emails (e.g., 'discuss', 'draft', 'schedule') caused misclassification of General Business emails into the collaboration category.
- Presence of Quoted Text: Sometimes the quoted text in general emails contained language similar to collaboration emails, thereby misclassifying class 0 as 1.

[Further breakdown with examples here]
Given additional time and resources, we might've liked to experiment with synthetic data generated by GPT-3 as well as using the vectors from these LLMs.

Key Learnings

- Data quality is crucial. No amount of model fine-tuning can fully compensate for data imbalance or scarcity, but data augmentation can mitigate this to some extent.
- We consistently faced a trade-off between precision and recall, ultimately prioritizing recall.
- Similarly, opting for interpretability and speedy experimentation meant sacrificing state-of-the-art vectorizers and advanced modeling techniques.

Dead Ends

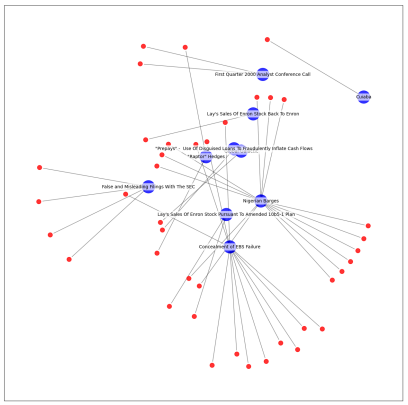

Fig A.6: Network Analysis of SEC data

Our initial use case was to match email content topics to GroupName topics from the SEC Case Complaint documents. Using TF-IDF and cosine similarity, we aimed to group emails based on similarity to these topics. This approach provided a rough estimate of email distribution across various threshold similarity values, visualized with a network graph [Fig A.6]. We then attempted to create synthetic data for supervised learning for two GroupNames: the Raptor Hedge scheme and "Improper Use of Reserves" using LLM and random email sections. We trained and compared Random Forest and Logistic Regression models. We encountered several issues with the training data used for this approach and ultimately opted to use the CMU labeled email data for our supervised learning scope.

Part B.1 Unsupervised Learning

For our unsupervised learning scope, we explored what we could learn about how the main topics of the charges against Enron by the Securities and Exchange Commission were described semantically. We were curious what unsupervised methods would show about word frequency, natural clustering, and ways of determining a topic from a set of documents (legal or technical) that might later be used in an email discovery. We used the complaint document (18776) that describes the charges against the main defendants, Skilling and Lay, as it appeared the most complete in describing the schemes to defraud which took place at Enron.

Pre-processing Steps

- Conversion of text to lowercase, elimination of extra spacing and new lines.
- Feature Extraction: We chose the TF-IDF vectorizer again for its ability to highlight important terms within and across data examples. This method helps reduce the impact of common words that are less informative as well as rare words.
- Normalization: The resulting feature matrix was normalized using L2 normalization. This helped maintain consistent scaling across features and improved the performance of clustering algorithms by ensuring that no single feature dominated due to scale differences, leading to more balanced and meaningful clusters.

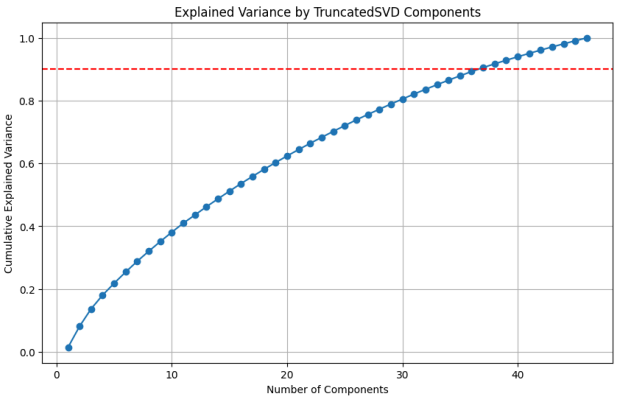

Fig B.1.1: Explained Variance using TruncatedSVD

- Dimensionality Reduction: We applied TruncatedSVD (Singular Value Decomposition) to reduce the feature space while preserving as much variance as possible. The final dimensions were reduced to 50 components, chosen based on the cumulative explained variance [Fig B.1.1]. Beyond 50, the incremental gain was minimal.

Clustering Algorithms

K Means

We applied K means to the SEC Complaint documents dataset to establish a baseline. We selected 7 clusters (K=7) based on the silhouette score method, which measures the quality of clustering [Fig B.1.2]. We relied on the results of K-Means to establish a baseline of the number of clusters.

To visualize the clusters formed, we reduce the vectors to 2 -dimensions using t-SNE [Fig B.1.3].

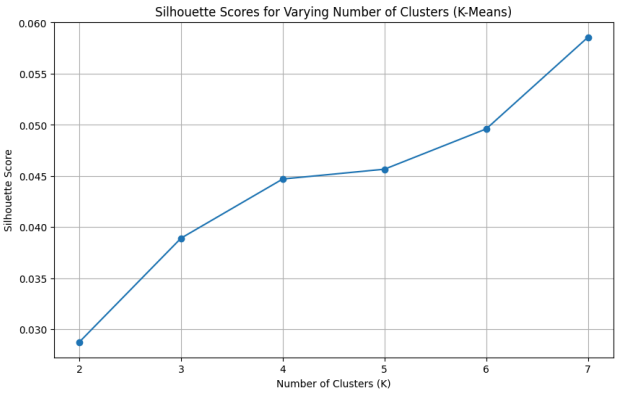

Fig B.1.2: Silhouette Score for K means clustering

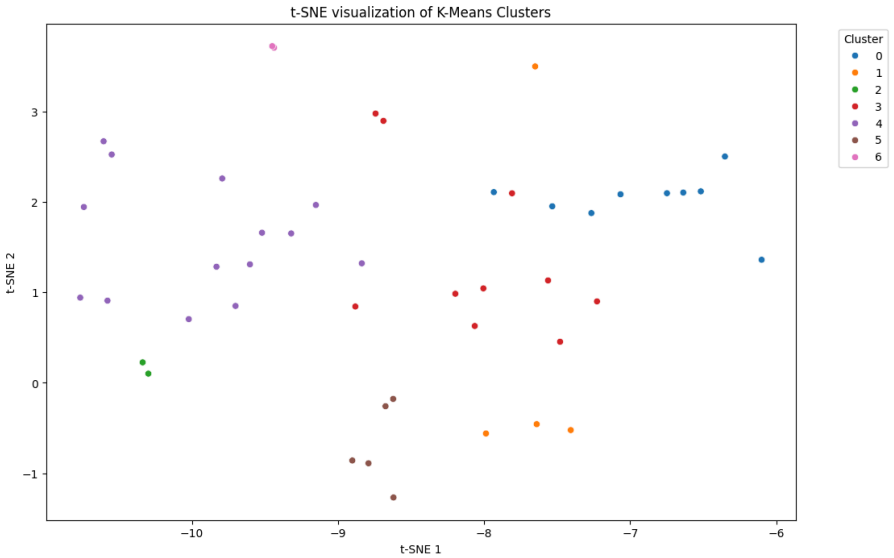

Fig B.1.3: K Means Cluster representation using t-SNE

Latent Dirichlet Allocation

We chose Latent Dirichlet Allocation as an alternative to K-Means. It is a probabilistic model that effectively discovers topics within the documents using term distributions across the data. For equivalent comparison, we set the number of topics the same as the k value in k means (i.e., 7). This allows us to directly compare the results from both methods and assess their effectiveness in clustering our data. Fig B.1.4 shows us the top 10 words for each topic.

Non-negative Matrix Factorization

Non-negative Matrix Factorization is a non-probabilistic technique that decomposes the original data matrix into two lower-dimensional matrices with non-negative elements, capturing parts-based representations. It is particularly useful for finding interpretable parts in the data. Similar to LDA, we maintained the number of topics to 7 for equivalent comparison. Fig B.1.5 shows us the top 10 words for each topic.

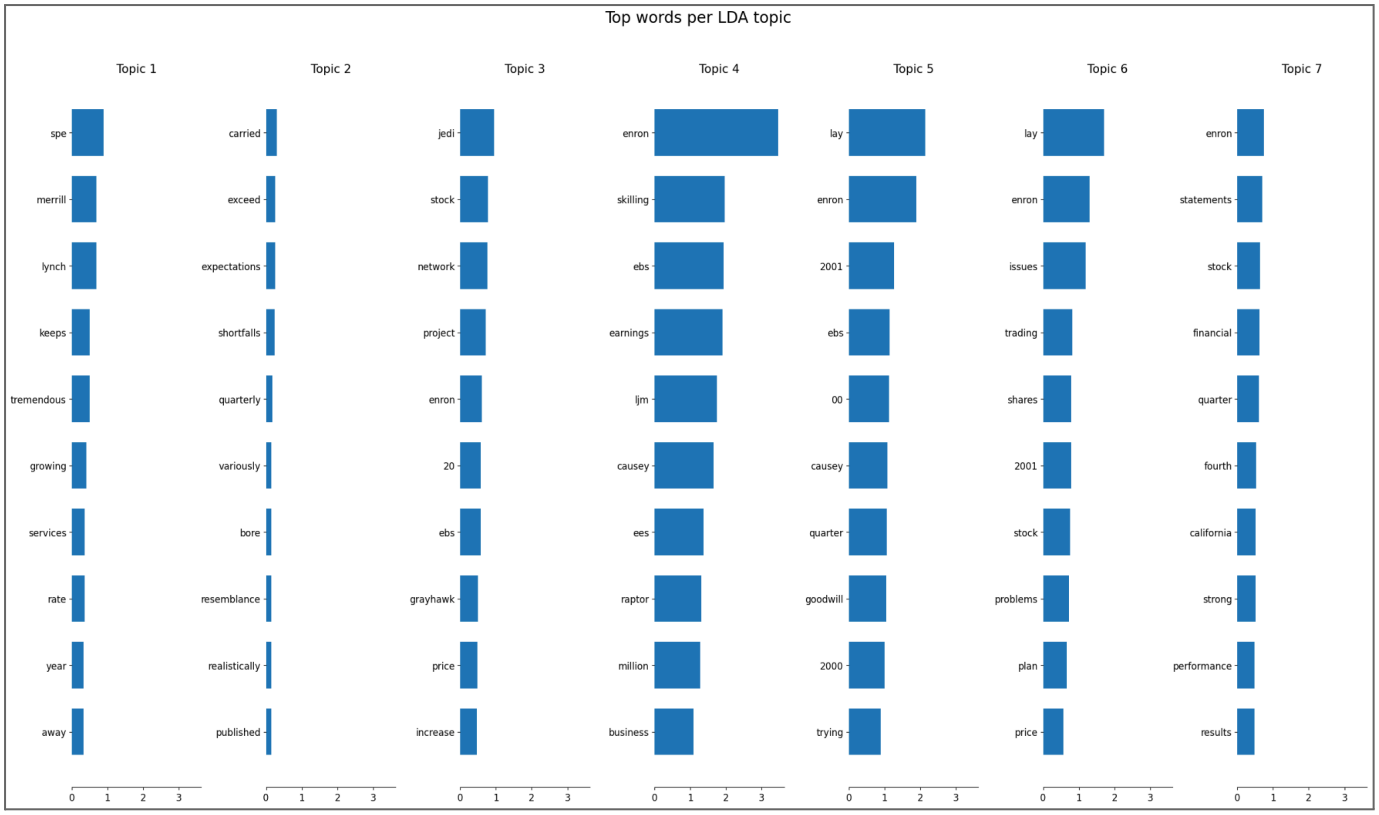

Fig B.1.4: Top 10 words in each topic clustered using LDA

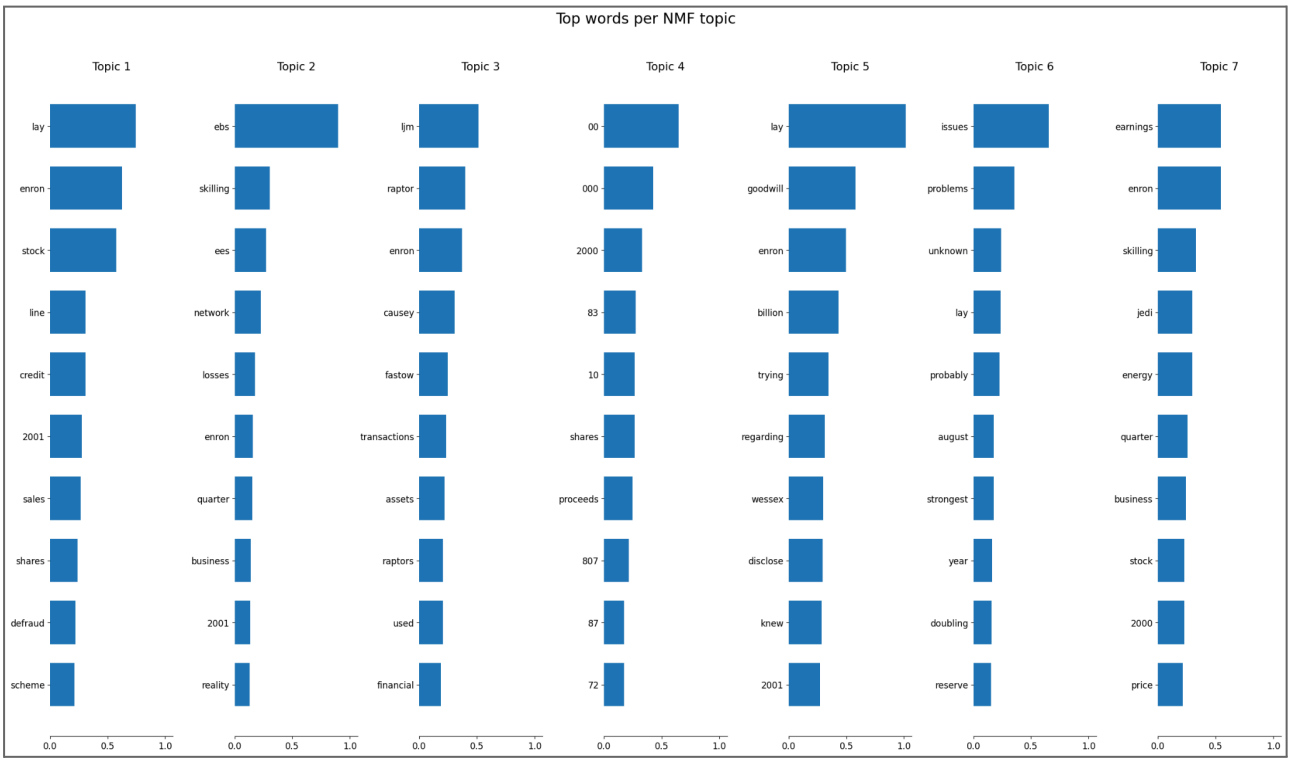

Fig B.1.5: Top 10 words in each cluster formed by NMF

A Comparison of the Topics Clustered Using NMF and LDA

LDA Topics	
Cluster	**Topic Name**
0	Corporate Financial Entities
1	Performance and Expectations
2	Stock and Network Operations
3	Corporate Governance and Earnings
4	Financial Statements and Goodwill
5	Trading and Shares
6	Financial Performance and Results

NMF Topics	
Cluster	**Topic Name**
0	Corporate Misconduct
1	Business Operations
2	Financial Transactions
3	Securities and Shares
4	Financial Statements and Goodwill
5	Corporate Issues
6	Earnings and Energy Market

A Comparison Between the Algorithms

Perplexity Values:

Looking at the perplexity values of NMF and LDA, we observed that Increasing the number of topics generally improved perplexity but the improvements diminished beyond a certain point, indicating an optimal number of topics around 7 [Fig B.1.6]. Further increases in the number of topics led to fragmented and less coherent topics.

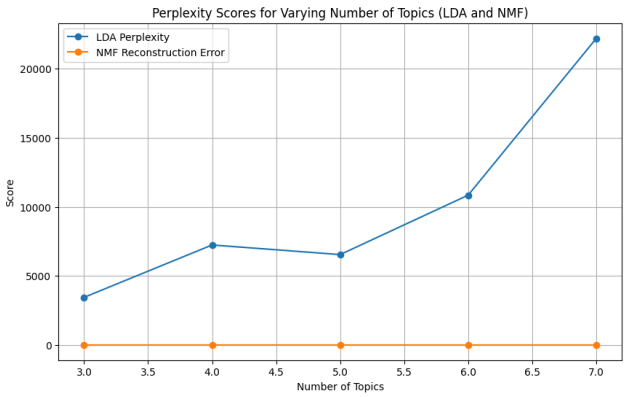

Fig B.1.6: Perplexity score comparison b/w NMF and LDA

Themes and Topics:

Across all algorithms, recurring themes related to financial misconduct, business operations, and regulatory issues were consistently identified. This convergence underscores the dominant issues within the SEC Complaint documents.

Common terms such as "enron," "financial," "company," "market," and "business" were prevalent in the top terms across K-Means, LDA, and NMF, reflecting the central focus of the dataset on corporate and financial matters.

- K-Means clustering tends to capture broader, more generalized themes due to its reliance on distance-based clustering. For example, Cluster 1 in K-Means broadly encompasses themes around "financial transactions" and "reporting," without delving into specific entities or detailed aspects.
- LDA offers more granularity and specificity in topic modeling by analyzing word co-occurrence patterns across documents. It captures detailed aspects of financial and corporate activities, such as in Topic 1 (Corporate Financial Entities), which includes specific entities like "merrill" and "lynch." This allows for a more precise understanding of particular subjects within the documents. LDA's probabilistic nature helps in understanding the distribution of topics within the dataset.
- NMF, through its matrix factorization approach, provides highly interpretable topics that focus on specific parts of the data. For instance, Topic 1 (Corporate Misconduct) in NMF highlights terms like "lay" and "scheme," indicating a clear focus on individual actions and misconduct

within the corporate structure. It seems to be particularly useful for isolating and understanding different facets of complex data.

Key Learnings and Challenges

- NMF and LDA were able to distinguish meaningful and distinct topics, even with a relatively small number of clusters. Additionally, we noticed that the topics remained consistent across different modeling techniques, which reinforced the robustness of our results.
- We faced a challenge dealing with the high dimensionality and sparsity of the text data. It was also challenging to identify key topics within the clusters, as they needed some knowledge of the case issues.
- With more time and resources, we could experiment with other clustering algorithms and topic modeling methods, like Hierarchical Dirichlet Process (HDP) or BERT-based topic modeling to compare and potentially improve our results. Additionally, incorporating 3-dimensional interactive visualization could help in better understanding topic distribution.

Part B.2 Unsupervised Learning- RAG

To conclude our experiments in semantic search and topic associations, we also explored RAG approaches and evaluation methods using the Enron documents as our knowledge base. Our results include a set of Jupyter Notebooks and a separate RAG demo application in a HuggingFace Gradio Space. Part of this effort was trying new document ingestion methods, including a package called Unstructured.io for assembly-line-like extraction and parsing of documents. The new document data included the Enron annual reports (10-K's) from 1993 through 2000, and three Enron historical case studies. We discarded two of the case studies for the RAG notebook analysis since the content overlapped significantly between sources. We felt having fewer overlapping chunks demonstrated retrieval more effectively. In production use cases, however, a larger number of sources may be most effective, so we used all case study sources in the RAG demo application and employed an encryption scheme to preserve academic fair use guidelines with the copyrighted materials.

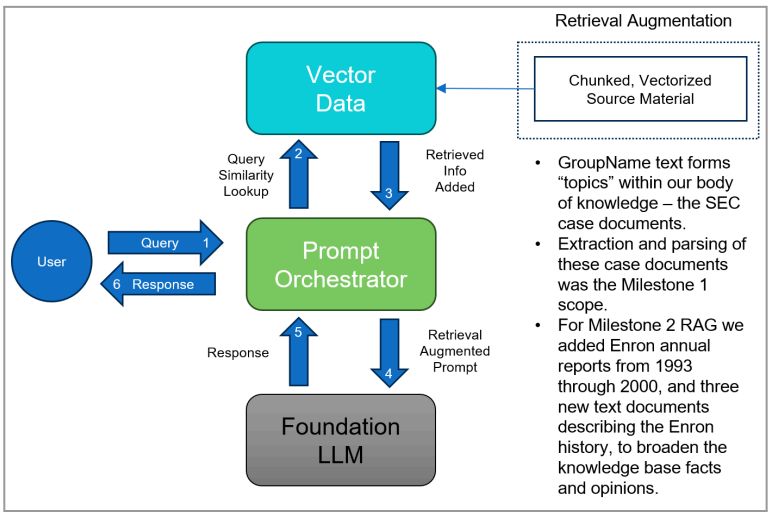

Fig B.2.1 RAG Architecture

The RAG efforts described here are intended as a brief introduction to approaches and tools that we might pursue in the future, not an exhaustive analysis. Surprises we encountered included the fact that some of the document chunks would not be returned even when queries were verbatim representations of their content. Challenges included realizing that evaluating a RAG framework would entail a huge effort as embeddings, chunkings, queries, and retrievals are varied to find the best combinations. Not surprising, but our demo also demonstrated that a well-known area of knowledge such as the Enron case shows modest RAG benefits with modern LLMs (3.5 and up) since they've been trained on all the articles, movies, books, etc. that exist on the history. In a future effort

we might include an evaluation tool such as TruLens and combine that with Weights and Biases integration for more extensive testing.

Technical Overview

We used the LangChain RecursiveCharacterTextSplitter function for our chunking, with 400 character chunks and 100 character overlap. Initially TF-IDF and then later SentenceTransformer was used for the embeddings. Cosine similarity provided our retrieval criteria.

We tried various queries and observed the results. While we didn't do a detailed check of each query-retrieval combination, we were satisfied that we had laid the foundation for a more serious RAG project in the future. The Weights and Biases integration points allow for details of experiment tracking and artifact capture on each run for future analysis (Each execution and related artifacts are logged to our authenticated WandB account).

Observing Similarity Distances with Dimensionality Reduction

For document retrieval similarity distances, we used UMAP and t-SNE. Our favorite was the UMAP 3D interactive we created [Fig B.2.2]. It shows the query, the retrieved documents, and un-retrieved document chunks in different colors, with pan, zoom, and dropdown selection.

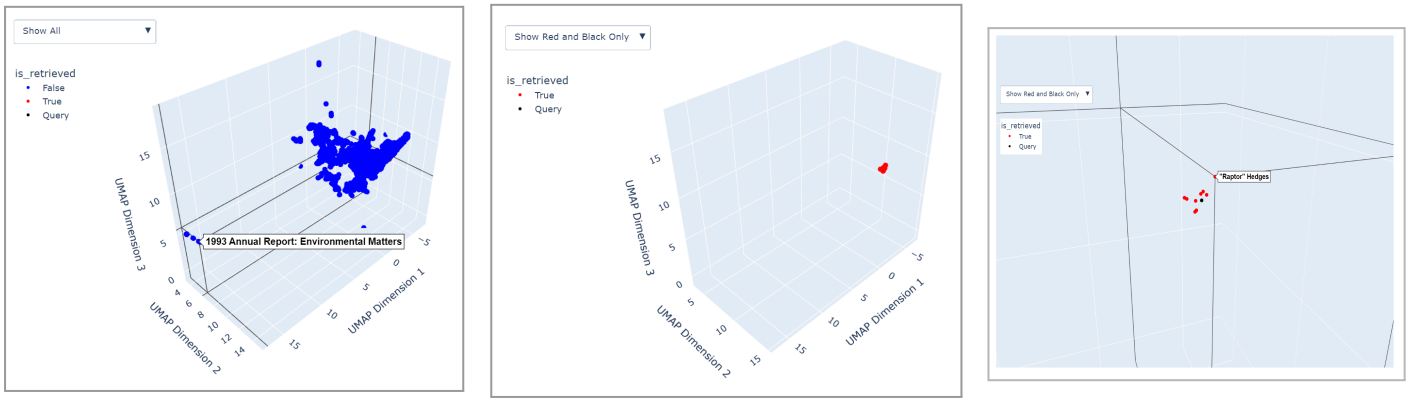

Fig B.2.2: UMAP 3-D interactive

UMAP and t-SNE are both unsupervised methods for dimensionality reduction. UMAP preserves global relationships better than t-SNE. See Figure B.2.3 of t-SNE for comparison to the UMAP in Fig B.2.2 above. The document embedding grouping is more bunched up in t-SNE, showing less focus on global relationship distances.

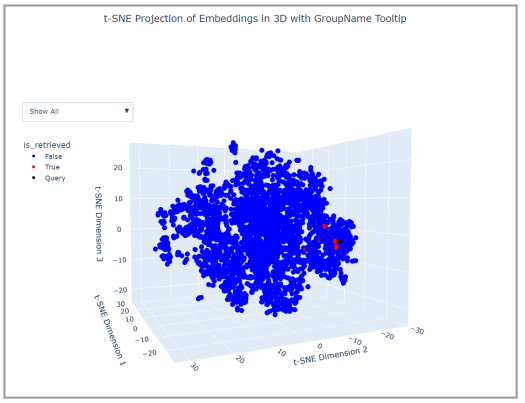

Fig B.2.3: t-SNE embeddings

An observed example:

In Fig B.2.2 with UMAP, the 1993 Annual Report documents are far from the other chunks, given that they contain chunked content from before most of the fraudulent activities began. The UMAP representation preserves these global distances better than the t-SNE version.

[Note: By panning, dragging, and zooming on the visuals in the demo app one can get a feel for the distances of different document types.]

Visualizing Chunk Relationships with Hierarchical Clustering

After much experimentation we settled on dendrogram and tree map representations with hover-over tooltips. Both of these are hierarchical unsupervised learning methods for clustering. A dendrogram [Fig B.2.4] represents a bottoms-up, agglomerative clustering, and helps give a general idea of groupings, but requires a lot of interpretation. Dendrograms use branches to show the order and distance of clustering, while tree maps [Fig B.2.5] are good for revealing proportions.

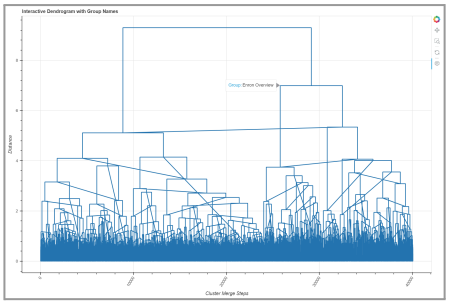

Fig B.2.4: Dendogram

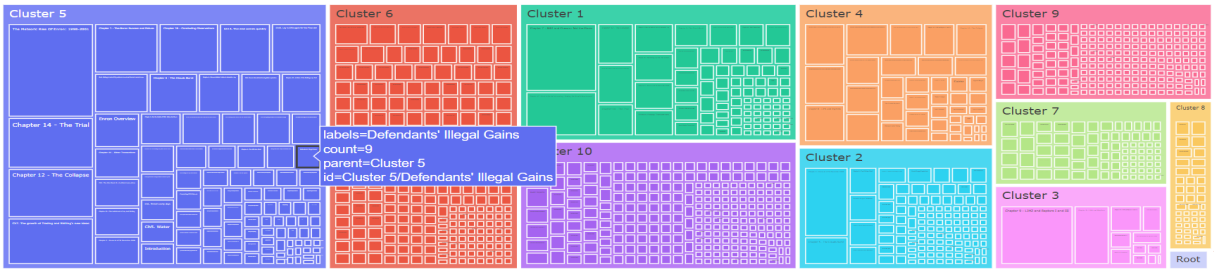

Fig B.2.5: Treemap

Ethical Considerations

- In the context of legal discovery, false positives (for example, misclassifying a General Business email as Document Collaboration) and false negatives (failing to identify a Document Collaboration email) have significant consequences. False positives can lead to unnecessary scrutiny, while false negatives can result in missed critical information.
- Transparency and accountability are key when it comes to such problem statements. It is vital that we use interpretable models as much as possible in order to justify the results.
- In the context of RAG, if the documents provided are biased, the 'facts' returned by the model will also contain bias. For example, the case complaints include the opinions of the government's prosecution team which are strongly inclined towards pressing charges.
- There is an increasing need of introducing a human in the loop to verify the results reported by any unsupervised learning, especially RAG. They should be someone with domain knowledge.

Related Work

- Our Milestone 2 here is only leveraging the preprocessed data from Milestone 1- no overlap in scope.
- Supervised learning project[3] on Medium [Machine Learning with Python on the Enron Dataset | by Will Koehrsen]
- Supervised learning project[4] on Medium [Data Cleaning for The Enron Fraud Machine Learning Case | by Elena Eidson | Medium]

- Recent MADS Capstone project that focused on retrieval-augmented generation, but used a very different data set (MADS program info). (Capstone Team's Github: https://github.com/psollars/capstone)

Statement of Work

- Use of LLMs: Tools such as ChatGPT, Gemini, and Copilot were used in our project for experimentation on tasks such as synthetic labeled data creation, document text extraction, code consultation, as well as searching on NLP approaches and concepts.

- Division of Tasks:

Team Member	Tasks
Anandita Bodas	Supervised Email Modeling Code and Analysis
	Final Report Content
	Final Report Compilation and Formatting
	Github Setup
Nathan Goldhardt	Unsupervised Topic Modeling Code and Analysis
	Final Report Content
Gary Schaumburg	Prototype Code Across Complete Scope
	Unsupervised RAG Code and Analysis
	HuggingFace Demo App for RAG
	Final Report Content
	Linking Data from Milestone 1
	Scrum Master

Appendix A- References

1.Wikipedia contributors. (2024, February 4). Enron: The Smartest Guys in the Room. In *Wikipedia, The Free Encyclopedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Enron:_The_Smartest_Guys_in_the_Room&oldid=1203293550

2.June, F. (2024, February 2). Advanced RAG 02: Unveiling PDF parsing. In *Towards AI* on *Medium*. Retrieved from https://pub.towardsai.net/advanced-rag-02-unveiling-pdf-parsing-b84ae866344e?sk=v2%2Fc7ce9ee2-1bae-46e1-b197-d6fafba2e76a

3. Koehrsen, W. (n.d.). Machine learning with Python on the Enron dataset. Medium. Retrieved from https://medium.com/@williamkoehrsen/machine-learning-with-python-on-the-enron-dataset-8d71015be26d

4. Eidson, E. (n.d.). Data cleaning for the Enron fraud machine learning case. Medium. Retrieved from https://medium.com/@egmp777/data-cleaning-for-the-enron-fraud-machine-learning-case-96b14b971d14

5. Aggarwal, C. C., and Reddy, C. K. (Eds.). (2013). *Data clustering: Algorithms and applications*. CRC Press.

Appendix B- Additional Links

- Github link
- Link to error analysis of final supervised learning model
- HuggingFace demo app for RAG