

UTRECHT UNIVERSITY  
Department of Information and Computing Science

---

Applied Data Science master thesis



**Comparing Unsupervised Learning Approaches for Topic  
Classification of Bank Complaints: An NLP Study**

**First examiner:**

Prof. dr. Antal van den Bosch

**Candidate:**

Kitharidis Sofoklis 8492816

**Second examiner:**

Yupei Du

July 11, 2023

## Abstract

This thesis investigates the application of unsupervised learning algorithms, namely KMeans, Latent Dirichlet Allocation (LDA), BERTopic, and Hierarchical clustering to analyze customer complaint data in the banking sector. The research aims to uncover patterns, topics, and insights from the complaints to enhance customer satisfaction strategies.

The problem statement revolves around understanding the impact of different natural language processing methods on the comprehension of financial complaint data and their comparative performance. The key research question addresses how various NLP methods influence the understanding of financial complaint data and how these methods can be compared.

To address this question, the study utilizes four unsupervised learning algorithms: KMeans, LDA, BERTopic, and Hierarchical clustering. KMeans is employed with Word2Vec, Doc2vec, TF-IDF and BERT embeddings, while LDA is applied using Bag of Words, TF-IDF, and Word2Vec representations. BERTopic with DBSCAN and hierarchical clustering algorithm is also explored with Word2Vec, Doc2vec, TF-IDF and BERT embeddings.

The analysis reveals significant findings, including the identification of key topics in the customer complaints dataset and the comparison of different clustering approaches. The results demonstrate that KMeans with Word2Vec embeddings achieves the highest cluster separation and density, indicating its superior performance. LDA highlights relevant topics related to loans, payments, communication, debt, and banking services. BERTopic with DBSCAN demonstrates improved cluster separation and provides precise and distinctive topics.

In summary, this research provides valuable insights into the understanding of financial complaint data using unsupervised learning algorithms. The findings contribute to the development of customer satisfaction improvement strategies in the banking industry. Last but not least, the study

---

addresses ethical considerations, such as privacy and data integrity, ensuring responsible research practices throughout the analysis.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Motivation and context . . . . .	5
1.2	Literature overview . . . . .	6
1.3	Research question . . . . .	7
<b>2</b>	<b>Data</b>	<b>8</b>
2.1	Consumer Complaint Database Overview . . . . .	8
2.2	Selected data exploration results . . . . .	9
2.3	Data preparation for analysis including motivation . . . . .	11
2.4	Ethical and legal considerations of the data . . . . .	13
<b>3</b>	<b>Method</b>	<b>15</b>
3.1	Translation of the research question to a data science question	15
3.2	Motivated selection of methods for analysis . . . . .	15
3.3	Motivated settings for selected methods . . . . .	18
<b>4</b>	<b>Results</b>	<b>32</b>
4.1	Selected analysis results . . . . .	32
<b>5</b>	<b>Conclusion</b>	<b>43</b>
5.1	Answering the data science question . . . . .	43
5.2	Answering the research question . . . . .	44
5.3	Describing implications for the proper domain setting . . . . .	45
5.4	Discussing ethical implications and consideration . . . . .	46
5.5	Limitations . . . . .	47
5.6	Future work . . . . .	49
<b>Appendix</b>		
<b>A</b>	<b>Appendix</b>	<b>51</b>
A.1	Annotated scripts and results of analyses and method settings	51

**Bibliography**

**67**

# 1. Introduction

## 1.1 Motivation and context

Customer complaints in the banking sector provide a huge repository of information about customer expectations, potential problems and possible areas for service improvement. This feedback plays a key role in identifying problems in banking services, enhancing customer satisfaction, ensuring smooth service operations and even potentially discovering opportunities for product innovation. By analysing these complaints, banks can gain a competitive advantage, provide superior quality of service and even prevent systemic risks.

However, the high-volume, high-velocity, and unstructured nature of complaint data makes it challenging to derive these insights using traditional analysis techniques. Herein lies the significance of applying unsupervised learning techniques, and for several reasons:

- **Handling Unstructured Data:** Complaint data is usually unstructured textual data. Unsupervised learning, especially Natural Language Processing (NLP) techniques, are ideal for analyzing such data. They can process and analyze large volumes of text, identify patterns, and even interpret the sentiment behind the complaints. [1]
- **Scalability:** Unsupervised learning methods are highly scalable, making them well suited to handle the large volumes of complaints that banks often receive. [2]
- **Detecting Hidden Patterns:** Unlike supervised learning, unsupervised learning does not require labeled data. It can detect hidden patterns and structures within the data that might not be immediately obvious. This is particularly valuable when analyzing complaints, as it allows

for the discovery of unexpected issues or themes that may not have been previously considered. [3]

- **Dimensionality Reduction:** Techniques such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) can help in reducing the dimensionality of the data, making it easier to visualize and understand.
- **Clustering for Customer Segmentation:** Unsupervised learning techniques like K-means clustering or Hierarchical clustering can help categorize complaints into different groups based on their similarities. This can reveal distinct categories of complaints, which can then be addressed more effectively. [4]
- **Topic Modeling:** Techniques such as Latent Dirichlet Allocation (LDA) can be used to identify the key topics that are being discussed within the complaints. This can provide a high-level overview of the main areas of concern for customers. [5]

Thus, by applying unsupervised learning methods, banks can harness a powerful weapon that can not only make sense of vast, complex customer complaint data sets, but also discover deep, actionable insights that can drive strategic decision-making and service improvement initiatives.

## 1.2 Literature overview

It becomes clear that unsupervised machine learning methods have been widely applied to textual data analysis, particularly in customer complaints, due to their ability to handle large, unlabeled datasets and extract actionable insights from them. Here are some examples of previous research in this area:

- **Customer Complaints Analysis Using Text Mining and Outcome-Driven Innovation Method for Market-Oriented Product Development, by Junegak Joung, Kiwook Jung, Sanghyun Ko and Kwang-soo Kim (2018)**

This paper appears to discuss the use of text mining techniques for

analyzing customer complaints. It is highlighted that text mining can help reveal patterns and trends in customer feedback that can be used to guide innovation and product development. Reference is made to various text mining techniques, such as natural language processing (NLP), and how these methods can be applied specifically to customer complaint data to guide the product development process in a way that better meets market needs and customer desires. [6]

- **Text mining for central banks, David Bholat, Stephen Hansen, Pedro Santos and Cheryl Schonhardt-Bailey (2015)**

This paper focuses on data mining in the context of central banking. It reports on the use of text mining to analyse various types of text data, such as reports, news articles and to a much lesser extent customer complaints, to extract meaningful information that can be used in decision-making processes within central banks. Finally, it concludes with specific challenges and opportunities related to the application of text mining techniques within central banking. [7]

- **Clustering with Deep Learning: Taxonomy and New Methods, by Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel and Daniel Cremers (2018)**

This paper provides a comprehensive classification of clustering methods based on deep learning. These unsupervised learning techniques are particularly effective in dealing with unstructured data such as text, audio and image data. It discusses methods such as self-encoders and other deep learning-based techniques for clustering data, a common task in text analytics. [8]

### 1.3 Research question

The research question can be formulated as follows:

*How do different NLP methods impact the understanding of financial complaint data, and how can these methods be compared?*



## 2. Data

### 2.1 Consumer Complaint Database Overview

The data utilized in this thesis is sourced from the Consumer Complaint Database provided by the Consumer Financial Protection Bureau (CFPB). This database contains consumer complaints about financial products and services, offering valuable insights into consumer experiences and aiding in the regulation of the consumer financial market. It is important to note that the published complaints in the database are those that have been sent to companies for response and are eligible for publication. Additionally, personal information is removed by the Bureau before publishing the consumer's narrative description, ensuring privacy.

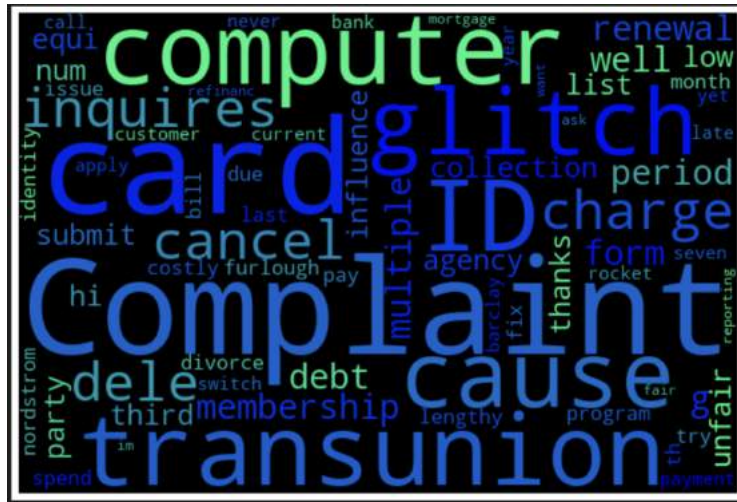
The complaint narratives provided by consumers within the database offer firsthand descriptions of their experiences. It is important to consider that the views expressed in these narratives are those of the consumers and do not necessarily align with the views or verification of the Consumer Financial Protection Bureau. Furthermore, the lack of complaints or a relatively low volume of complaints about a particular product, issue, or company does not necessarily imply the absence of consumer harm. Factors such as company size, market share, and the population in a specific geographic area should be taken into account when evaluating complaint volume.

By acknowledging the limitations and context of the data, the findings and recommendations presented in this thesis can be appropriately interpreted and applied in the banking industry, supporting informed decision-making and strategies to enhance customer satisfaction and regulatory practices. [9]

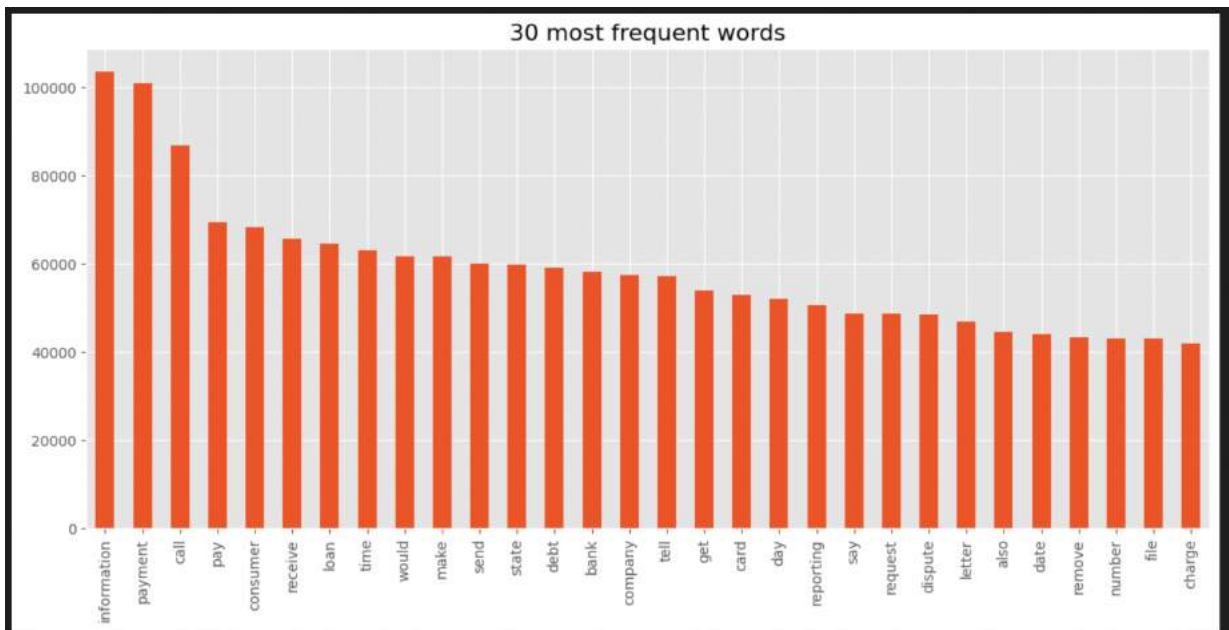
## 2.2 Selected data exploration results

Based on the data exploration and the creation of a word cloud and a bar plot, here are some potential key findings:

- **High Frequency of Specific Terms:** Words like "complaint", "transunion", "ID", "card", "information", "payment", and "call" appear most frequently in the complaint texts. This suggests that these are key topics or themes that are common across many of the customer complaints.
- **Identification of Potential Issues:** The prominent appearance of terms like "payment", "call", and "information" could indicate potential issues relating to payment processing, customer service interactions, and information handling or reporting.
- **Focus on Certain Products or Services:** The presence of "transunion" and "card" could signify that a significant portion of the complaints pertain to credit card services or issues related to credit reporting (TransUnion is a credit reporting agency).
- **Frequent Customer Actions or Feelings:** Words like "pay", "receive", "loan", "time", "would", "make" are also common, suggesting they are significant in the context of these complaints. These words might indicate common actions taken by customers (like making a payment or receiving a loan), or could express feelings or intentions ("would", "make").
- **Need for Deeper Analysis:** While these high-level observations provide some insights, further analysis could be beneficial to understand the context around these words and to uncover more complex themes or sentiment in the complaint texts. For instance, topic modeling or sentiment analysis could be applied to further explore the data.



**Figure 2.1:** WordCloud object from our data with the most frequently occurring words.



**Figure 2.2:** Bar plot of the 30 most frequently occurring words.

This type of analysis provides an overview of the data and helps to identify keywords and potential themes. However, for a deeper understanding of the topics, more advanced text mining techniques will be needed.

## 2.3 Data preparation for analysis including motivation

The following section will briefly explain the procedure followed for the preparation of the data:

### 1. Data Loading and Initial Pre-processing

- The raw data was loaded from a CSV file, with the 'complaint ID' column set as the index, and the 'Consumer complaint narrative' column was renamed as 'complaint'. This resulted in a dataset with 3,668,628 rows and 17 columns.

Complaint ID	Date received	Product	Sub-product	Issue	Sub-issue	complaint	Company public response	Company	State	ZIP code	Tags	Consumer consent provided?	Submitted via	Date sent to company	Company response to consumer	Timely response?	Consumer dispute
7008130	2023-05-22	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Account information incorrect	NaN	NaN	TRANSUNION INTERMEDIATE HOLDINGS, INC.	OH	44130	NaN	NaN	Web	2023-05-22	In progress	Yes	
7008144	2023-05-22	Debt collection	I do not know	Attempts to collect debt not owed	Debt was result of identity theft	NaN	NaN	Consumer Adjustment Company Incorporated	FL	33016	NaN	NaN	Web	2023-05-22	Closed with explanation	Yes	
7008145	2023-05-22	Credit card or prepaid card	General-purpose credit card or charge card	Problem with a purchase shown on your statement	Card was charged for something you did not pur...	NaN	NaN	BARCLAYS BANK DELAWARE	NJ	07047	NaN	NaN	Web	2023-05-22	In progress	Yes	
7006278	2023-05-22	Debt collection	I do not know	Attempts to collect debt not owed	Debt is not yours	NaN	Company has responded to the consumer and the ...	SUNRISE CREDIT SERVICES, INC	MO	64063	NaN	NaN	Web	2023-05-22	Closed with explanation	Yes	
7007141	2023-05-21	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Account status incorrect	NaN	NaN	TRANSUNION INTERMEDIATE HOLDINGS, INC.	MD	20743	NaN	NaN	Web	2023-05-21	In progress	Yes	

Figure 2.3: This is how data looks like before pre-processing step.

- To simplify and streamline the dataset, all the rows with missing data ('NaN' values) in the 'complaint' column were dropped and removed duplicates. This resulted in a cleaned dataset with 1,142,509 rows and 1 column.

- Due to resource constraints, 10% of the cleaned dataset was randomly sampled using a seed for reproducibility, reducing the dataset size to approximately 114,250 rows. [10] [11]

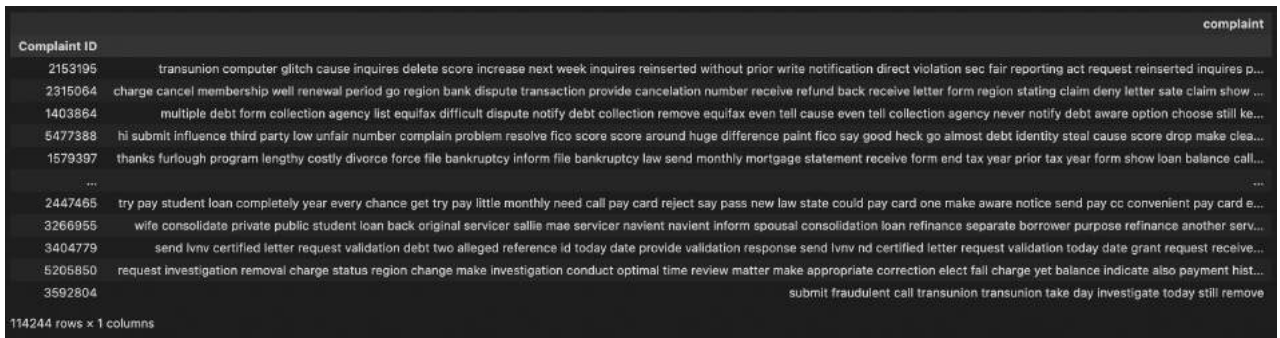
### 2. Text Preprocessing and Cleaning

This step involved the use of the "preprocess\_text" function, which

performs the following tasks:

- It first removes all special characters and numbers from the text, keeping only alphabetic characters.
- It tokenizes the text into individual words for further processing.
- It assigns part-of-speech tags to each token. This information is used in the next step.
- It lemmatizes each token based on its part of speech, converting words to their base form (e.g., 'running' to 'run'). The WordNetLemmatizer from NLTK is used for this task, which can enhance the accuracy of subsequent text analysis by reducing the dimensionality of the data and consolidating similar words.
- It removes stop words, which are common words like 'the', 'is', 'in', etc., that typically don't carry much meaningful information. Both NLTK's list of English stop words and a custom list of additional stop words including the top 3 words with a huge difference in the number of appearances compared to the rest of the words ('xx', 'xxxx', 'xxxxxx', 'xxxxxxxx', 'xxxxxxxxxxxx', 'account', 'report', 'credit') are used. This helps focus the analysis on the words that are likely to be most relevant to the complaints.
- It filters out any tokens that are less than 2 characters long, further refining the data.
- Finally, it rejoins the processed tokens into a single string of text, removes any extra spaces, and returns the cleaned and pre-processed text.

The 'preprocess\_text' function is a critical component of this analysis. It prepares the raw complaint texts for subsequent analysis by converting them into a more manageable and interpretable form. This process, often referred to as 'text normalization', enhances the effectiveness of the text mining techniques that will be applied later, by reducing noise, homogenizing the text, and highlighting the terms that are likely to be most informative. [12] [13] The code of data cleaning is detailed in Appendix, Chapter A.1.1



The image shows a screenshot of a data table with a dark background. The table has two columns: 'Complaint ID' and 'complaint'. The 'Complaint ID' column contains numerical values such as 2153195, 2315064, 1403864, 5477388, 1579397, 2447465, 3266955, 3404779, 5205850, and 3592804. The 'complaint' column contains detailed text descriptions of customer issues, such as 'transunion computer glitch cause inquires delete score increase next week inquires reinserted without prior write notification direct violation sec fair reporting act request reinserted inquires p...', 'charge cancel membership well renewal period go region bank dispute transaction provide cancelation number receive refund back receive letter form region stating claim deny letter sate claim show ...', and 'multiple debt form collection agency list equifax difficult dispute notify debt collection remove equifax even tell cause even tell collection agency never notify debt aware option choose still ke...'. The table is truncated at the bottom, showing '114244 rows x 1 columns'.

**Figure 2.4:** This is how data looks like after pre-processing step.

## 2.4 Ethical and legal considerations of the data

The dataset derived from customer complaints in the banking sector. Handling such data does come with its share of ethical and legal considerations.

- **Data Privacy and Anonymity:** One of the foremost ethical and legal concerns when dealing with customer complaints is respecting the privacy of the individuals involved. Even though personally identifiable information (PII) from the dataset is removed, it is crucial to ensure that none of the complaints can be traced back to the individuals who submitted them. Data anonymization is not only an ethical requirement but also a legal one under various data protection regulations such as the General Data Protection Regulation (GDPR) in the EU and the California Consumer Privacy Act (CCPA) in the United States. [14]
- **Data Quality and Integrity:** As with any dataset, there are ethical considerations related to data quality and integrity. Inaccurate data, whether due to errors in collection or intentional manipulation, can lead to faulty conclusions. The data should therefore be verified to the extent possible to ensure its accuracy and completeness. [15]

Regarding each potential concern outlined above, comprehensive measures have been undertaken to mitigate any associated risks in this study. Specifically, in terms of Data Privacy and Anonymity, this study has been carefully designed to utilize only the textual content of each complaint, without any direct or indirect reference to the individuals who lodged them. This ensures a high degree of anonymity and privacy in the data utilized,

which aligns with various data protection regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).

As for the concern related to Data Quality and Integrity, it is essential to note that the sheer volume of data involved, coupled with meticulous data cleaning processes, is likely to minimize any potential inaccuracies or discrepancies. During the data cleaning and preparation phase, particular attention was given to removing duplicate entries, handling missing values, and ensuring the overall reliability and completeness of the dataset. Thus, any potential threats to data integrity and subsequent analysis are greatly reduced, and the results derived from this data are likely to be accurate and reliable.

## 3. Method

### 3.1 Translation of the research question to a data science question

The vast complexity of financial complaint data presents a unique challenge for data scientists. The ability to classify and understand this data can yield significant insights and guide decision-making processes in financial institutions. The aim of this thesis is the transition from a broad research query to a specific data science question, trying to quantitatively address these challenges.

The key focus will be:

*How do the performances of different embeddings methods (TF-IDF, Word2Vec, Doc2Vec, and BERT) and topic modeling techniques (K-means, LDA, BERTopic, and Hierarchical Clustering) compare when applied to the classification of complaint data, based on various metric scores?*

This targeted data science question provides a concrete framework for the investigation, allowing to systematically explore, analyze, and compare the effectiveness of various state-of-the-art techniques in handling complaint data.

### 3.2 Motivated selection of methods for analysis

The choice to utilize unsupervised learning methods for the analysis of customer complaints in the banking sector was principally motivated by the nature of the data and the research question. Here's why these methods, namely KMeans, LDA, BERTopic and Hierarchical clustering are particularly apt:



- **Handling Unstructured Data:** Customer complaints are predominantly unstructured text data. Traditional methods that handle structured data are not well-equipped to analyze this type of data. Unsupervised learning methods, especially those involving Natural Language Processing (NLP) techniques, are designed to handle and derive insights from unstructured text data. They can process and analyze large volumes of text, identify patterns, and interpret the sentiment behind the complaints.
- **Scalability:** The large volume of customer complaints requires a scalable solution. KMeans, LDA, BERTopic and Hierarchical clustering are scalable unsupervised learning techniques that can handle large datasets and provide real-time insights. [16]
- **Discovering Hidden Patterns:** The research question entails identifying underlying patterns and themes in the complaints. Unsupervised learning methods, not relying on pre-labeled data, excel at detecting hidden patterns and structures within the data. KMeans and Hierarchical clustering can classify complaints into different groups based on similarities, while LDA and BERTopic can discover underlying topics in the data. This reveals common issues that customers face, providing actionable insights. [17]
- **Flexibility of Cluster Analysis:** Both Hierarchical clustering and DBSCAN provide additional flexibility in understanding data structure. Hierarchical clustering provides a dendrogram that gives an intuitive understanding of the data clusters and how they merge, offering insights at various levels of granularity. DBSCAN, on the other hand, can discover clusters of various shapes and sizes and is adept at handling noise in the data. [18] [19]
- **Advancements in NLP:** The evolution of NLP, especially in terms of transformer-based models like BERT, has led to more sophisticated unsupervised techniques like BERTopic. These advanced models can capture the context between words and provide more accurate and insightful topic extraction from the text. [20]

Given the nature of the data and the research question, unsupervised learning methods like KMeans, LDA, BERTopic and Hierarchical clustering are appropriate. They facilitate a comprehensive exploration of the data, ultimately revealing actionable insights for the banking sector.

- **KMeans:** Scalability and Efficiency. Considering the large size of customer complaint dataset, KMeans is an efficient method to quickly segment the complaints into various clusters, given its low computational cost. It could help identify major areas of customer dissatisfaction based on the clustering of similar complaint texts. [21]
- **LDA (Latent Dirichlet Allocation):** Topic Discovery and Assignment. Given that customer complaints could be on a variety of issues, LDA could be a useful tool to automatically discover underlying topics within the complaint dataset. The model could assign multiple topics to each complaint, thus providing a multifaceted view of the issues customers are facing. [5]
- **BERTopic:** Semantic Coherence. In a customer complaint dataset, it is essential to capture the contextual semantics of words as the same words could mean different things in different contexts. BERTopic uses BERT embeddings to achieve this, which could lead to the discovery of more interpretable and semantically coherent topics. [22]
- **Hierarchical Clustering:** Flexibility in Determining Number of Clusters. Given the lack of prior knowledge about the number of distinct categories of complaints in your dataset, hierarchical clustering could be a proper choice. This method would give the opportunity to visually inspect the dendrogram and decide on an appropriate number of clusters based on the dataset's structure. [23]
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Detection of Outliers and Handling Arbitrary Shapes. Customer complaint data can have outliers (for example, complaints that are not common) and clusters of arbitrary shapes. DBSCAN, being a density-based method, can effectively detect these outliers and handle clusters of various shapes and densities, providing a more nuanced

view of the data. [19]

### 3.3 Motivated settings for selected methods

The choice of parameters and settings for the selected methods was strongly influenced by the nature of the dataset and the goal of analysis. The primary objective was to categorize the bank's customer complaints into coherent clusters, thereby assisting the banking institution in recognizing and addressing common complaints.

Feature extraction methods, such as TF-IDF, BERT embeddings, Word2Vec, and Doc2Vec, were utilized to transform the text data into a numerical format, suitable for use by the algorithms.

#### **Feature Extraction:**

- **TF-IDF (Term Frequency - Inverse Document Frequency):** This statistic determines a word's importance within a document in a corpus. The TF-IDF value increases with a word's frequency in a document but decreases with its occurrence in the corpus, allowing for handling common words. It is beneficial for large text data as it assigns higher weight to document-specific terms, transforming the data into a matrix usable by algorithms like KMeans. Therefore, TF-IDF is effective for managing large textual datasets and identifying word significance within a document and across the corpus. This technique was applied using `TfidfVectorizer` from `sklearn's feature_extraction` module. The `min_df` parameter was set to 50, meaning that a word will be considered only if it appears in at least 50 documents. This method returns a matrix where each row represents a document and each column represents a term. The value in each cell represents the importance of the term in the document. [24]
- **BERT Embeddings:** BERT, an acronym for Bidirectional Encoder Representations from Transformers, is a pre-training technique used in natural language processing that relies on a transformer-based machine learning model. A key feature of BERT is its ability to generate

contextual embeddings, implying that the same word can have varying embeddings contingent on its context. This feature allows BERT to accurately capture the context in which words appear, making it particularly suitable for tasks involving complex language structures, such as customer complaints. By comprehending the semantic implications of words in different contexts, BERT facilitates the creation of embeddings that serve as meaningful input for clustering algorithms, thereby contributing to a more effective analysis of textual data. The BERT model was loaded using the SentenceTransformer function with the 'bert-base-nli-mean-tokens' and 'all-MiniLM-L6-v2' model. The embeddings were then generated using the encode function. These embeddings provide context-aware representations for each sentence. [20]

- **Word2Vec:** Word2Vec is a shallow, two-layer neural network model that is designed to produce word embeddings. Its functionality is based on the prediction of each word within a certain context window, and through this process, it learns to create vector representations for words. For our task, Word2Vec has been employed to comprehend and encode the meanings of words based on their context within customer complaints. This model's strength lies in its ability to encapsulate semantic and syntactic similarities between words, which allows it to generate meaningful input for clustering algorithms. Through this method, we can better understand the patterns and themes within the complaints. The Word2Vec model was trained using Gensim's Word2Vec function. The embeddings were generated by taking the average of the embeddings of all the words in each sentence. [25] [26]
- **Doc2Vec:** Doc2Vec is an extended model of Word2Vec, specifically designed to represent not only individual words, but entire documents. In addition to Word2Vec's word vectors, Doc2Vec incorporates an extra vector for paragraphs or documents. For our use-case, Doc2Vec has been applied to establish a numerical representation of the complete complaints, effectively capturing the associations among words and the collective semantic implication of each complaint. This method

is particularly beneficial as it regards the whole complaint as a singular context, providing a more comprehensive perspective for our dataset. Consequently, these generated document vectors offer significant insights for further processing. In our case, the documents were tokenized and each document was tagged with a unique identifier. The Doc2Vec model was then trained using these tagged documents. The `vector_size`, `min_count`, `window`, and `seed` parameters were set according to the requirements. The embeddings were then extracted using the `infer_vector` function. [27]

The code of feature extraction for each method is detailed in Appendix, Chapter A.1.2

### Evaluation Metrics:

- **Coherence Score:** Coherence Score is a metric used to evaluate the quality and interpretability of topics generated by topic modeling algorithms, such as LDA (Latent Dirichlet Allocation). It measures the degree of semantic similarity between words within a topic by considering their co-occurrence patterns. A higher coherence score indicates that the words in a topic are more closely related and provide a clearer and more meaningful representation of the underlying theme. Coherence Score helps in selecting the optimal number of topics and assessing the overall quality of topic models. The range of Coherence Score values typically falls between 0 and 1, with higher values indicating better coherence. However, it's important to note that the exact range can vary depending on the specific implementation and calculation method used. [28]
- **Silhouette Score:** The Silhouette Score is a measure of how well each sample in a cluster is assigned to its own cluster compared to other clusters. It ranges from -1 to 1, where a score close to +1 indicates that the samples are well-clustered, with clear separation between clusters, while a score close to 0 indicates overlapping clusters or ambiguous assignments. Lastly, a score close to -1 indicates that the samples may have been assigned to the wrong clusters.

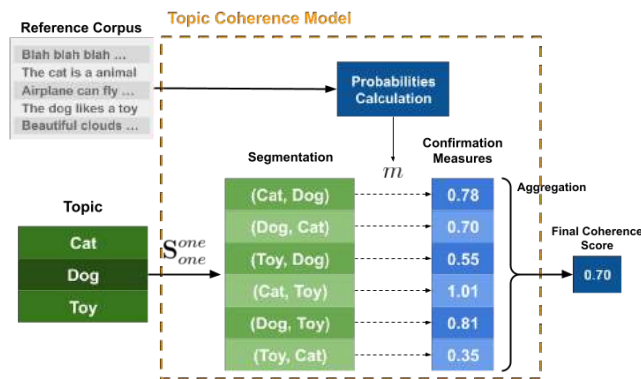


Figure 3.1: This is the formula of Coherence score metric.

The Silhouette Score can help evaluate the quality of clustering results, with higher scores indicating better-defined clusters. However, it is important to consider domain-specific knowledge and interpretability when assessing the practical significance of the Silhouette Score. Last but not least, it is generally used as a relative measure to compare different clustering algorithms, parameter settings, or datasets. [29] [30] [31]

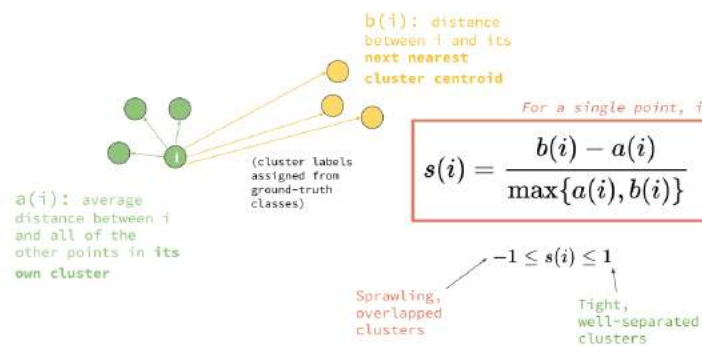


Figure 3.2: This is the formula of Silhouette score metric.

- **Calinski-Harabasz Score:** The Calinski-Harabasz Score, also known as the Variance Ratio Criterion, is a measure of the compactness and separation of clusters. It evaluates how well the data points are grouped into clusters by considering the ratio of the between-cluster dispersion to the within-cluster dispersion.

The Calinski-Harabasz Score ranges from zero to positive infinity, where:

- A higher score indicates better-defined and well-separated clusters. -

A lower score suggests overlapping clusters or insufficient separation between clusters.

Similar to the Silhouette Score, the Calinski-Harabasz Score is used as a relative measure for comparing clustering algorithms, parameter settings, or datasets. It is important to note that the interpretation of the score may vary depending on the specific problem domain and the characteristics of the dataset. [32] [31]

$$\begin{aligned}
 k &= \text{Number of clusters} \\
 n_q &= \text{Number of points in cluster } q \\
 c_q &= \text{Cluster center of cluster } q \\
 n_E &= \text{Number of data points} \\
 c_E &= \text{Cluster center of all points} \\
 \text{Between-cluster dispersion, } B &= \sum_{q \in k} n_q (c_q - c_E)(c_q - c_E)^T \\
 \text{Within-cluster dispersion, } W &= \sum_{q \in k} \sum_{x \in \text{cluster } q} (x - c_q)(x - c_q)^T \\
 \text{Calinski-Harabasz score (s)} &= \frac{B}{W} \times \frac{n_E - k}{k - 1}
 \end{aligned}$$

**Figure 3.3:** This is the formula of Calinski-Harabasz score metric.

### Methods:

- **KMeans Clustering:** KMeans is a centroid-based clustering algorithm that assigns data points to the nearest centroid iteratively until stable clusters are formed. The optimal number of clusters is determined using the elbow method and the KneeLocator function, which identifies the "elbow" point in the inertia plot. The quality of clustering and hyperparameter selection is evaluated using the Silhouette Score and the Calinski-Harabasz Index.

The same process is applied to different feature representations: TF-IDF, BERT embeddings, Word2Vec, and Doc2Vec. For each feature set, the elbow method is used to determine the number of clusters, followed by KMeans clustering with evaluation metrics calculated. The KMeans algorithm is initialized with 'k-means++' and 300 maximum iterations for all clustering tasks.

By following this approach, we were able to perform KMeans cluster-

ing on different feature representations and assess their quality using the Silhouette Score and Calinski-Harabasz Index as evaluation metrics. [33] [34]

The code of k-Means process is detailed in Appendix, Chapter A.1.3

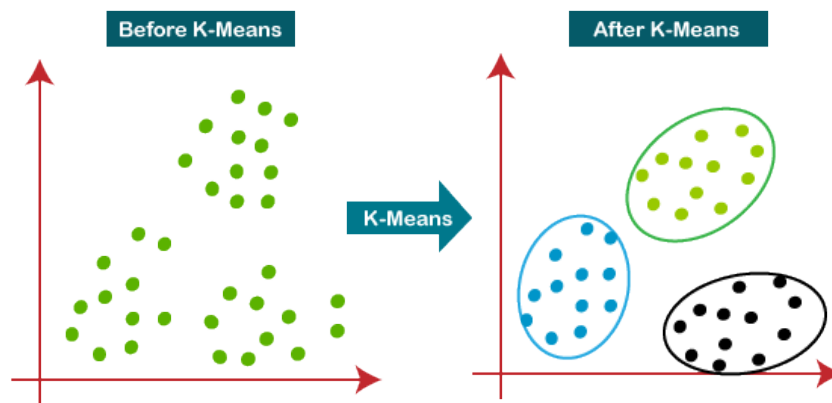


Figure 3.4: k-Means clustering.

- LDA (Latent Dirichlet Allocation):** LDA is a generative probabilistic model used primarily in Natural Language Processing to automatically classify documents into topics. It achieves this by identifying patterns of word occurrence and assuming that each document is a mixture of a certain number of topics, where a topic is characterized by a distribution over words.
  - LDA with Bag-of-words: The function `coherence_values` calculates coherence values of LDA models with different parameters, indicating topic quality. It trains various LDA models, compares their coherence scores using multiple cores for efficiency, and stores them in lists. Bag-of-Words inputs are utilized, and the best model is selected based on highest coherence, retrained, and its topics printed.
  - LDA with TF-IDF: The same function is reused, using TF-IDF representations this time. The process remains the same: train various models, select the optimal one with the highest coherence.
  - LDA with Word2Vec: Again, the same function is applied, using Word2Vec representations. Word2Vec is used to convert documents

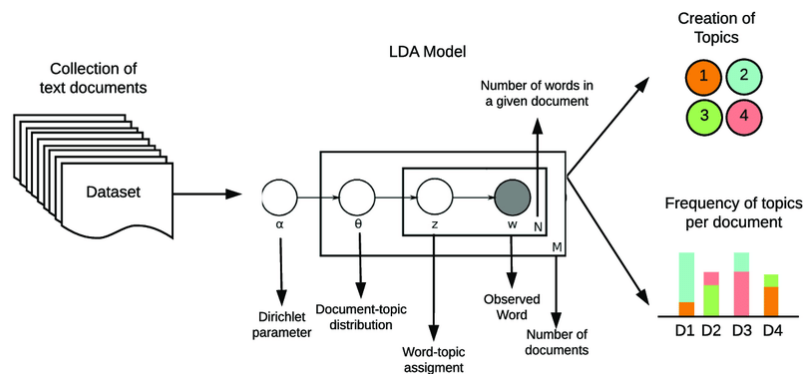


for Gensim compatibility. The process is repeated: train various models, select the optimal one with the highest coherence.

Note that in all of these cases, there is use of an approach called "grid search" to find the best hyperparameters (alpha, beta, and the number of topics). This involves training a model for each possible combination of hyperparameters within a specified range, and then selecting the combination that gives the best performance according to some metric (in this case, coherence).

Also, LDA model is not typically applied directly on BERT or Doc2Vec embeddings. These embeddings are high-dimensional and continuous, whereas LDA is designed for use with discrete count data (like word counts or TF-IDF scores). [35] [36]

The code of LDA process is detailed in Appendix, Chapter A.1.4



**Figure 3.5:** Topic Modeling and Latent Dirichlet Allocation (LDA) using Gensim.

- **BERTopic:** The code applies BERTopic for topic modeling, utilizing transformers, UMAP for dimensionality reduction, and HDBSCAN for clustering to identify topics in a text corpus. Steps, parameters, and settings include:
  - SentenceTransformer: First, a SentenceTransformer model is used to transform each sentence in the dataset into a high-dimensional vector. This is a type of word embedding that represents the semantic meaning of the sentence.
  - bert-base-nli-mean-tokens: This is a model trained on the task of

Natural Language Inference (NLI), where the model has to determine whether a given hypothesis is true, false, or undetermined based on a given premise. In this particular case, the original BERT model is used with base size (i.e., bert-base). It is trained on several large-scale NLI datasets and then fine-tuned to generate sentence embeddings. The "mean-tokens" part signifies that the embeddings of all tokens are averaged to obtain the final sentence embedding.

- all-MiniLM-L6-v2: MiniLM is a smaller and faster variant of the original BERT model. It is designed to provide comparable performance to larger models while being more computationally efficient. The "L6" refers to the fact that the model has six transformer layers, which is fewer than the 12 or 24 layers that are commonly found in BERT models. The smaller size of the model makes it faster to use and requires less memory. The "all" part signifies that the model has been trained on multiple languages, and "v2" indicates that this is the second version of the model.

- UMAP Model: The high-dimensional sentence embeddings are then fed into UMAP, a dimensionality reduction technique. UMAP reduces the dimensionality of the data, preserving the neighborhood relationships between data points, meaning that sentences with similar meanings stay close together in the reduced space.

- HDBSCAN: HDBSCAN is then used to cluster the reduced embeddings, effectively grouping sentences that have similar meanings together. Parameters `min_cluster_size` and `min_samples` dictate smallest cluster size and core point sample requirements.

- TfidfVectorizer: In addition to the embeddings, the sentences are also represented using TF-IDF, which transforms the text into a matrix of TF-IDF features. This allows the model to consider both the semantic meaning of the sentences (from the embeddings) and the actual words used in the sentences.

- BERTopic: All of the above components are used to create a BERTopic model. The model is then fitted to the data, which consists of a series

of complaints.

- Fit and Transform: BERTopic fits to the data and transforms it into topics and associated probabilities.

- Coherence Score Calculation: Coherence score from gensim’s CoherenceModel measures topic quality, higher scores indicating more coherent topics. [37] [38]

The code of BERTopic process is detailed in Appendix, Chapter A.1.5

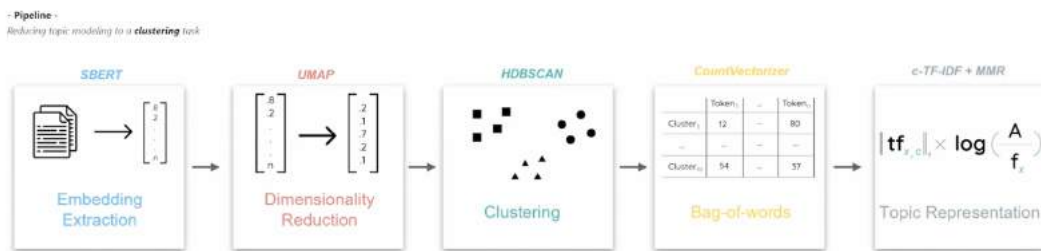


Figure 3.6: Topic Modeling with BERTopic.

- **Hierarchical Clustering:** Hierarchical clustering creates a hierarchy of clusters with a dendrogram, a tree-like diagram. The code implements it using BERT, Word2Vec, TF-IDF and Doc2Vec embeddings. Here are the key steps:

- Hierarchical Clustering: Leveraging the 'fastcluster' library, the code performs hierarchical clustering on the sentence embeddings using Ward’s method. This method minimizes within-cluster variance, thus ensuring that sentences within the same cluster are as similar as possible.

- Dendrogram Visualization: A dendrogram is created to visualize the hierarchical clustering results. The dendrogram is a tree-like diagram that showcases the nested grouping of sentences and the distance at which groupings merge, offering a clear insight into the cluster formation.

- Finding Optimal Clusters: The optimal number of clusters yield the highest silhouette score and it is based on the dendrogram.

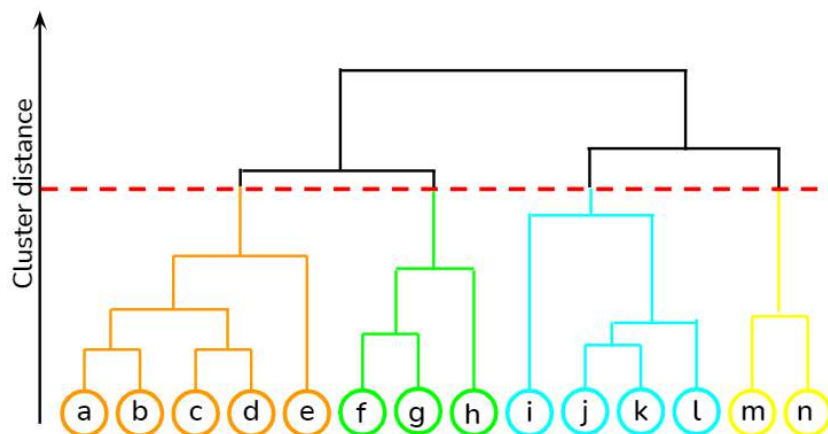
- Forming Clusters: From the hierarchical clustering, a predetermined

optimal number of clusters are extracted.

- Clustering Quality Assessment: To assess the quality of the clustering, Silhouette Score and Calinski-Harabasz Score are calculated.
- t-SNE Visualization: The high-dimensional BERT embeddings are reduced to two dimensions using t-SNE for visualization purposes. This reduction allows for the plotting of clusters in a two-dimensional space, where each cluster is represented with a unique color.
- Analyzing Cluster Content: For each formed cluster, the code tokenizes the sentences and counts the frequency of each token (word). By reporting the 15 most common words in each cluster, the analysis provides a means to interpret and understand the thematic content of each cluster.

The code executes the same steps for each of four embedding types: BERT (`bert_embeddings`), Word2Vec (`word2vec_embeddings`), TF-IDF (`X_tf_idf`) and Doc2Vec (`doc_vectors`). In each case, each row of the input matrix corresponds to the chosen representation of a sentence. While the overall clustering process remains the same, the resulting clusters differ due to the distinct features encapsulated by each type of embedding. [39] [40]

The code of Hierarchical clustering process is detailed in Appendix, Chapter A.1.6



**Figure 3.7:** Hierarchical clustering dendrogram.

### Hyperparameters

Hyperparameter tuning is guided by these scores: the coherence score guides the selection of the number of topics in LDA, while the silhouette score and Calinski-Harabasz score guide the selection of the number of clusters in the rest algorithms. The optimal parameters give the highest coherence/silhouette/Calinski-Harabasz score, indicating better topic clustering/representation.

The code runs on multiple representations of the text data and selects the configuration with the best evaluation metrics. This gives a comprehensive understanding of the text data by viewing it from different perspectives.

More specifically:

The alpha and beta parameters are hyperparameters that are used in the Latent Dirichlet Allocation (LDA) algorithm.

**Alpha** is a parameter of the Dirichlet prior on the per-document topic distributions. In the context of LDA, when alpha is higher, the documents are assumed to be made up of a mixture of most of the topics, and not any single topic specifically. A low value of alpha means that a document is likely to contain a mixture of just a few of the topics.

**Beta**, on the other hand, is a parameter of the Dirichlet prior on the per-topic word distribution. A high beta-value means that each topic is likely to contain a mixture of most of the words, and not any word specifically, while a low value means that a topic may contain a mixture of just a few of the words. [5]

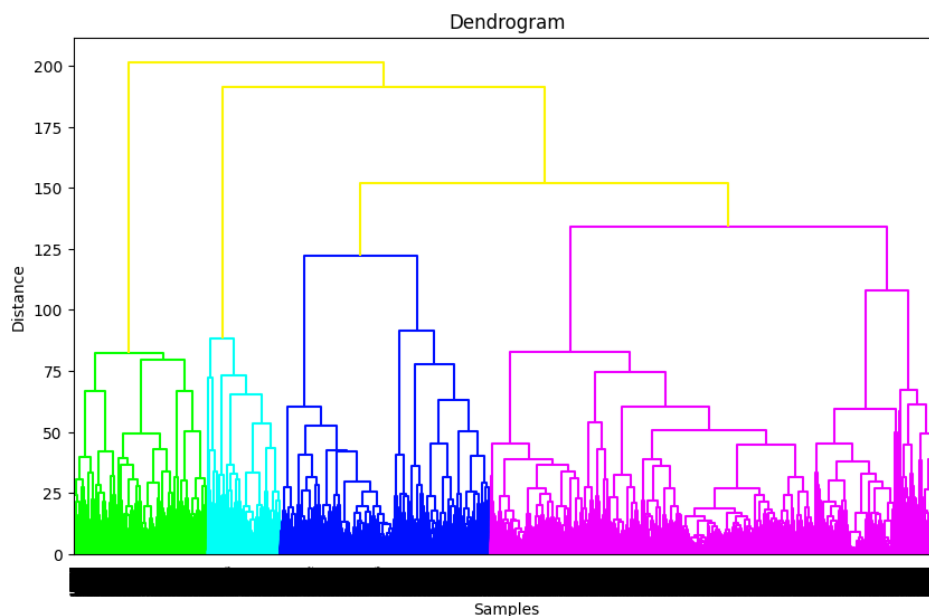
In the code, the alpha and beta parameters are used in the `coherence_`-values function to train the LDA model with different configurations. The aim is to find the combination of alpha, beta, and number of topics that results in the LDA model with the highest coherence score.

For each combination of alpha and beta values within the provided range (0.01 to 1 with a step of 0.3), an LDA model is trained and its coherence score is calculated. The LDA model configuration that gives the highest coherence score is considered optimal and is used for the final LDA model training.

This is part of the hyperparameter tuning process where various parameters are tested to find the most optimal model configuration. The alpha and beta parameters were adjusted to achieve better model performance in terms of topic coherence. By doing so, our aim is to find a model that generates more meaningful and interpretable topics.

Hierarchical clustering was performed on BERT embeddings, Word2Vec embeddings, Doc2Vec embeddings, and TF-IDF vectors. The following parameters and settings were used [25] [26]:

- The linkage method used was 'ward', which minimizes the variance of the distances between the clusters.
- The number of clusters was tuned by trying different values (from 4 to 8) and choosing the one that maximizes the silhouette score.
- The clusters were formed from the linkage matrix using the fcluster function. The number of clusters to form and the criterion 'maxclust' were given as inputs to this function.



**Figure 3.8:** The dendrogram visualizes the hierarchical clustering of documents.

#### Challenges in selecting the evaluation metrics

In the realm of unsupervised learning, and particularly in the domain of

clustering, the challenge of selecting an apt evaluation metric is considerably pronounced. Unlike supervised learning models, which hold the advantage of straightforward accuracy measures, unsupervised models lack a definite "ground truth" for comparison. This renders the process of model evaluation rather "foggy", obliging us to resort to certain proxy metrics to perform hyperparameter optimization and model comparison. For the task at hand, the selected proxy metrics are the Coherence Score, Silhouette Score, and Calinski-Harabasz Score. However, it's imperative to acknowledge the innate limitations each of these carries.

- **Coherence Score:** This metric has proven its effectiveness in topic modeling, predominantly evaluating the quality of the words in each topic cluster. It quantifies the degree of semantic similarity between high scoring words within a topic. Despite its popularity, one of the major drawbacks of the Coherence Score is its dependency on a good quality and appropriate word embedding model, as its performance can substantially deviate with poor or incompatible embeddings. Also, this score can sometimes lead to selecting over-specific or over-generic topics as the "best" ones. [41]
- **Silhouette Score:** The Silhouette Score offers a compact graphical representation of how well each object lies within its cluster. It's a measure of how similar an object is to its own cluster compared to other clusters. Despite its intuitiveness, the Silhouette Score carries some limitations. One of its significant drawbacks is its bias towards convex clusters, making it less effective with complex-shaped or density-based clusters. Moreover, this score can be computationally expensive for large datasets. [42] [43]
- **Calinski-Harabasz Score:** Also known as the Variance Ratio Criterion, this score is a ratio of the between-clusters dispersion mean and the within-cluster dispersion. Higher scores correspond to better-defined clusters. However, this score assumes the clusters to be convex and isotropic, which might not always hold true. It also expects clusters to be of similar sizes, hence can be biased towards larger clusters. [32]

[44]

In conclusion, while these metrics have their strengths, they are not without flaws. It is pivotal to consider these limitations when interpreting the results and to bear in mind that they are only heuristics, which may not perfectly align with the actual model's effectiveness in real-world applications.



## 4. Results

### 4.1 Selected analysis results

#### 4.1.1 K-Means

Comparing the results, it can be observed that K-means with Word2Vec embeddings achieved the highest Silhouette Score (0.10) and Calinski-Harabasz Index (18127.97), indicating better cluster separation and dense clusters. K-means with BERT embeddings also achieved relatively good results with a Silhouette Score of 0.071 and a Calinski-Harabasz Index of 7828.50.

On the other hand, K-means with TF-IDF and K-means with Doc2Vec embeddings yielded lower Silhouette Scores and Calinski-Harabasz Index values, indicating less distinct clusters and lower cluster density.

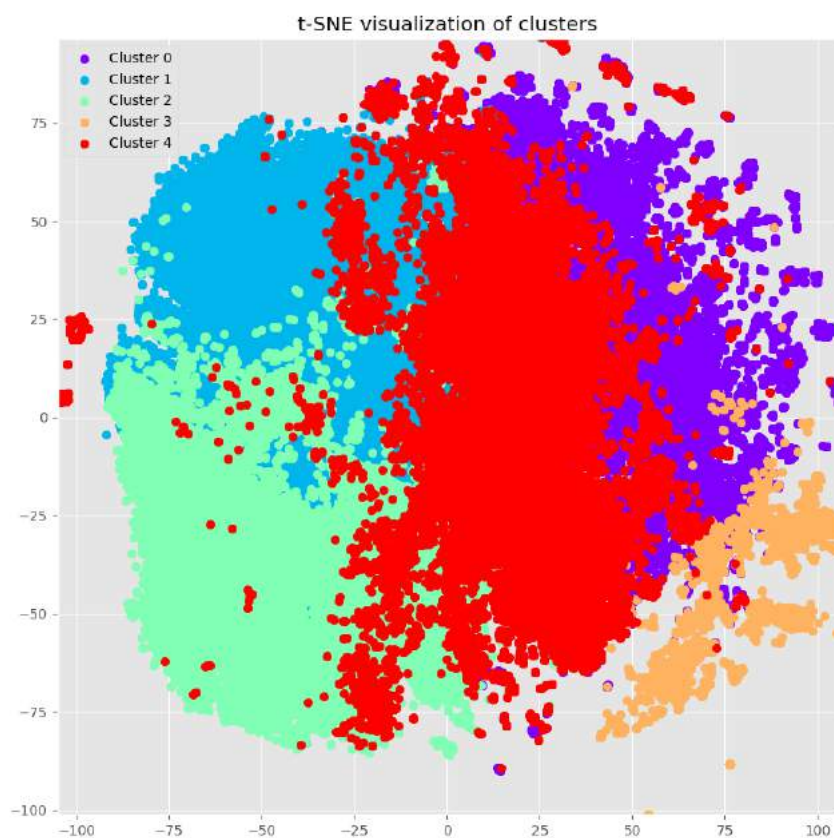
Overall, K-means with Word2Vec embeddings showed the most promising results, suggesting that the Word2Vec representations captured meaningful patterns and structures in the complaint data.

Based on the top 15 words for each cluster, we can try to identify possible topics or themes associated with each cluster in the K-means models with Word2Vec and BERT embeddings. Here's an overview of the possible topics for each cluster:

K-means with Word2Vec embeddings:

- **Cluster 0:** This cluster seems to be related to customer complaints about receiving or not receiving certain information, such as account balances, inquiries, or identity-related issues.
- **Cluster 1:** This cluster appears to be associated with banking transactions, including issues related to payments, addresses, and contacting the bank.

- **Cluster 2:** This cluster suggests complaints regarding inquiries, balances, and reporting issues, possibly related to credit cards or financial accounts.
- **Cluster 3:** This cluster might be related to debt-related complaints, including issues with debt collection, consumer rights, and disputes.
- **Cluster 4:** This cluster seems to involve complaints about banking services, including charges, customer service interactions, and related matters.

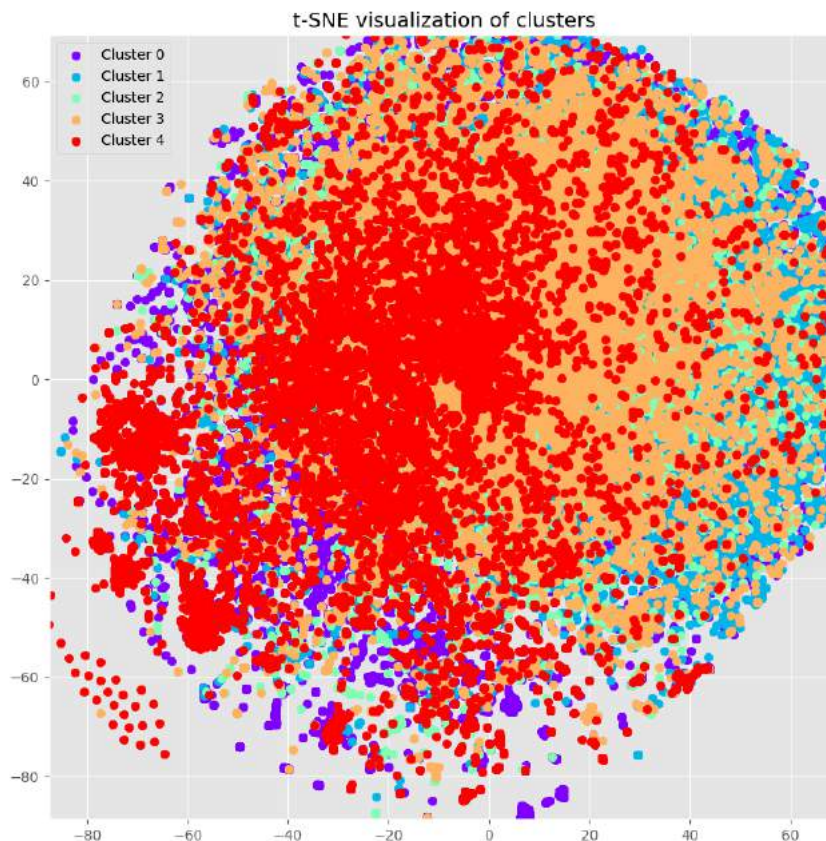


**Figure 4.1:** t-SNE visualization of clusters generated by k-means with Word2Vec embeddings.

K-means with BERT embeddings:

- **Cluster 0:** This cluster appears to involve complaints related to specific industries or entities, such as affiliations, branches, and chapters.
- **Cluster 1:** This cluster suggests complaints about allocated resources or arbitrary decisions made by entities.

- **Cluster 2:** This cluster seems to involve complaints about communication issues, possibly related to customer service or support.
- **Cluster 3:** This cluster might be associated with inquiries or disputes related to charges, credit, or debt collection.
- **Cluster 4:** This cluster appears to involve complaints related to identity theft, fraud, or unauthorized activities.



**Figure 4.2:** t-SNE visualization of clusters generated by k-means with BERT embeddings.

### 4.1.2 LDA

Latent Dirichlet Allocation (LDA) was applied to complaints data using Bag of Words (BoW), TF-IDF, and Word2Vec embeddings. The LDA model was evaluated with the coherence score to determine optimal parameters and topics.

BoW representation provided a coherence score of 0.49 with 7 topics. These topics appear related to loans and payments, with common words

like "loan," "payment," "bank," "call," and "information." Optimal alpha and beta parameters were both 0.31.

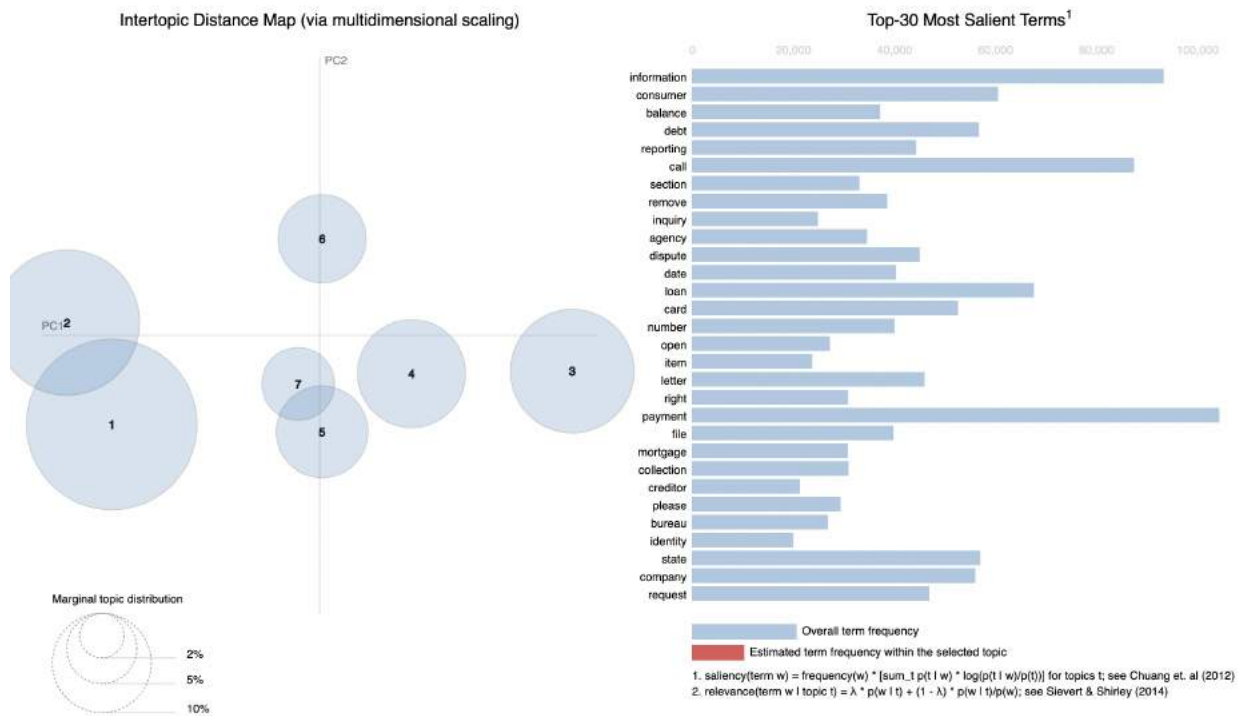
TF-IDF representation provided a coherence score of 0.53 with 12 topics. This representation produced a larger number of topics, with less frequent and specific words such as "mailbox," "unacceptable," and "capitalized."

Word2Vec provided a coherence score of 0.50 with 7 topics. This approach resulted in a more diverse topic range, including debt, payment, reporting, and consumer rights.

In conclusion, LDA model's optimal configuration and the interpretability of the topics depend on the text representation choice. BoW produced the highest coherence score, TF-IDF resulted in more nuanced topics, and Word2Vec captured a greater diversity of themes. These findings offer insight into the primary complaints found in the data, useful for guiding customer satisfaction improvement strategies.

### **Word2Vec model:**

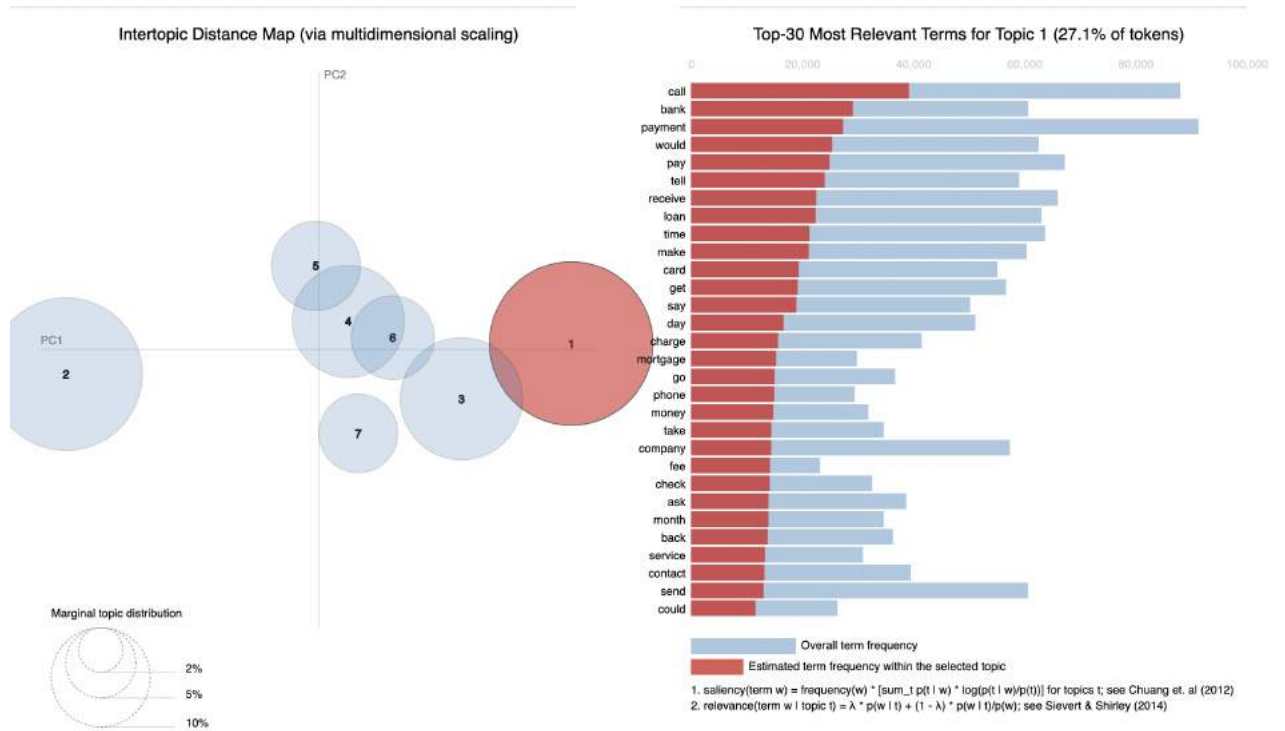
- Topic 0: Revolves around customer communication and disputes.
- Topic 1: Related to debt disputes.
- Topic 2: Deals with legal aspects of consumer rights.
- Topic 3: About issues with balances, payments, and disputes.
- Topic 4: Revolves around loans and mortgages, with communication aspects.
- Topic 5: About banking services, payment issues, and potential problems.
- Topic 6: Relates to debt collection, banking services, and potential legal issues.



**Figure 4.3:** Intertopic Distance Map visualizing the topics generated by LDA with word2vec.

**BoW model:**

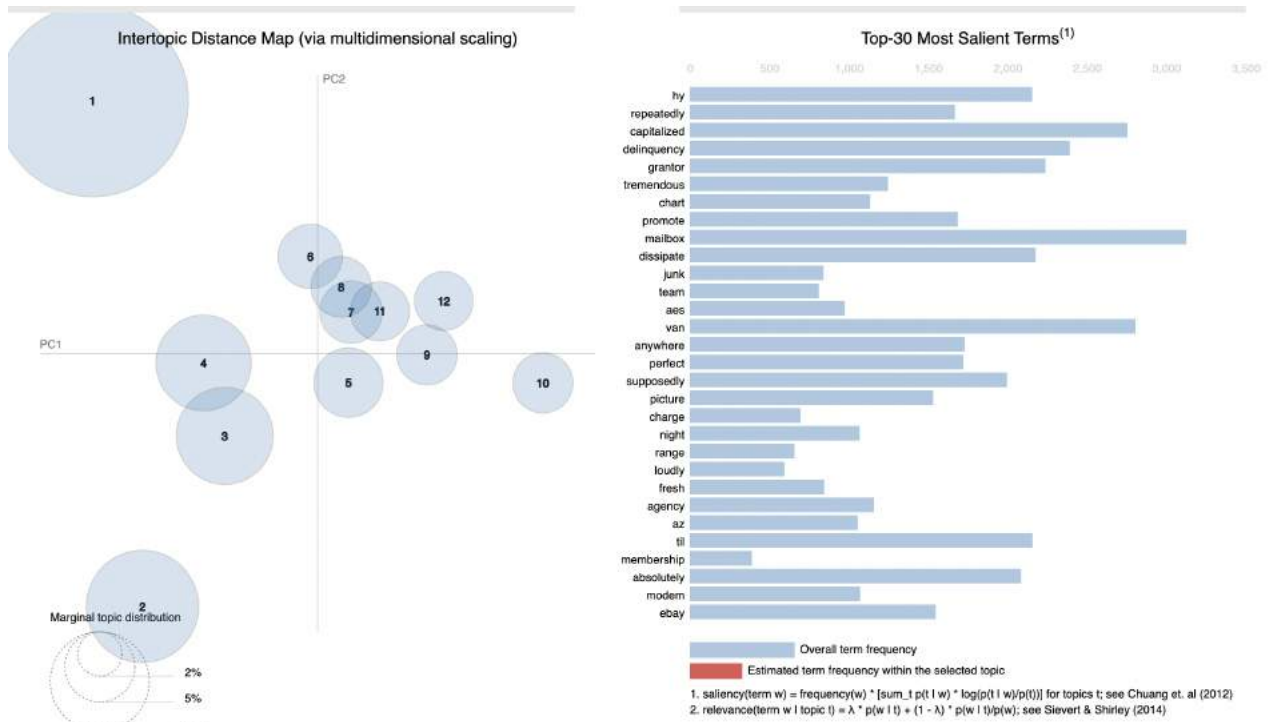
- Topic 0: Discusses filing claims with banks.
- Topic 1: Deals with loans, payments, and communication.
- Topic 2: Concerns late payment issues.
- Topic 3: Discusses communication and payment issues with banks.
- Topic 4: Relates to bank or credit card issues.
- Topic 5: Involves debt collection practices.
- Topic 6: Related to reporting issues or legal aspects in the financial sector.



**Figure 4.4:** Intertopic Distance Map visualizing the topics generated by LDA with Bag-of-Words.

**TF-IDF model:**

Topic 0-11: Topics seem disjointed while clear themes are not appearing possibly due to infrequent term weighting.



**Figure 4.5:** Intertopic Distance Map visualizing the topics generated by LDA with TF-IDF.

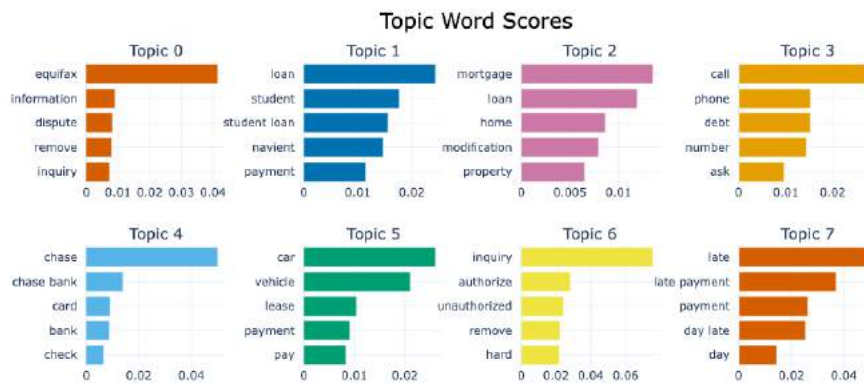
### 4.1.3 BERTopic with DBSCAN

Based on the Silhouette Scores alone, the experiment using BERT embeddings with the "all-MiniLM-L6-v2" model with Silhouette Score: -0.099 demonstrates a slightly better performance compared to the "bert-base-nli-mean-tokens" model with Silhouette Score: -0.155. The observed difference in Silhouette Scores suggests that the clusters generated with the "all-MiniLM-L6-v2" model exhibit improved separation and compactness, although it is important to note that both scores remain negative, indicating that the clusters may lack well-defined boundaries and clear separation.

The enhanced performance can be attributed to the utilization of the "all-MiniLM-L6-v2" sentence embedding model, which captures more comprehensive semantic information and contextual understanding from the input text data.

Comparing the generated topics between the two cases, it is evident that the second experiment employing the "all-MiniLM-L6-v2" model yields

more precise and distinctive topics. These topics exhibit a stronger focus and delve into finer details concerning various financial aspects, including credit reporting, specific banking institutions, loans, and legal procedures such as bankruptcy. Moreover, the topics from the second experiment incorporate more representative terms that effectively describe the content encapsulated within each cluster.



**Figure 4.6:** Visualization of topics with top words generated by BERTopic with all-MiniLM-L6-v2.

It is worth noting that despite conducting multiple tests by adjusting the parameters of UMAP (`n_neighbors`, `n_components`, `min_dist`) and HDBSCAN (`min_cluster_size`, `min_samples`), further enhancements in the results could not be achieved. This suggests that the chosen parameter values already optimize the clustering performance given the characteristics of the dataset.

#### 4.1.4 Hierarchical clustering

By comparing the results of hierarchical clustering using different embedding models, we can evaluate their performance based on the Silhouette Score and Calinski-Harabasz Score. Here are the results:

- Hierarchical clustering with BERT embeddings:
  - Silhouette Score: 0.027
  - Calinski Harabasz Score: 4336.72



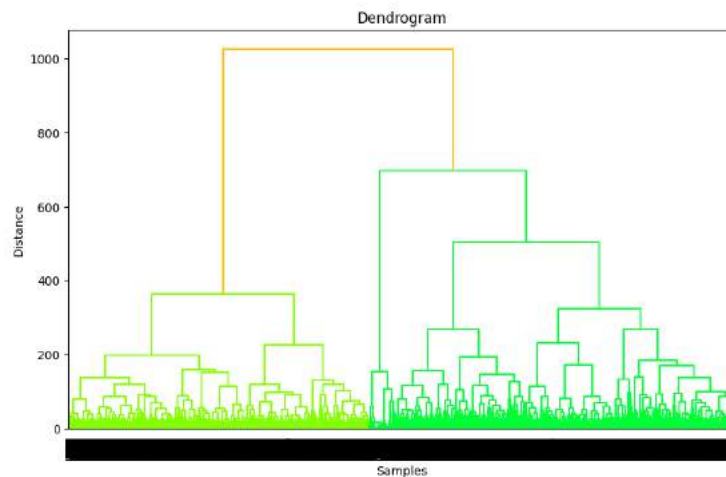
- Hierarchical clustering with TF-IDF:
  - Silhouette Score: 0.0047
  - Calinski Harabasz Score: 790.07
- Hierarchical clustering with Word2Vec:
  - Silhouette Score: 0.0704
  - Calinski Harabasz Score: 15511.66
- Hierarchical clustering with Doc2Vec:
  - Silhouette Score: -0.107
  - Calinski Harabasz Score: 1414.08

Based on the evaluation scores, we can observe the following:

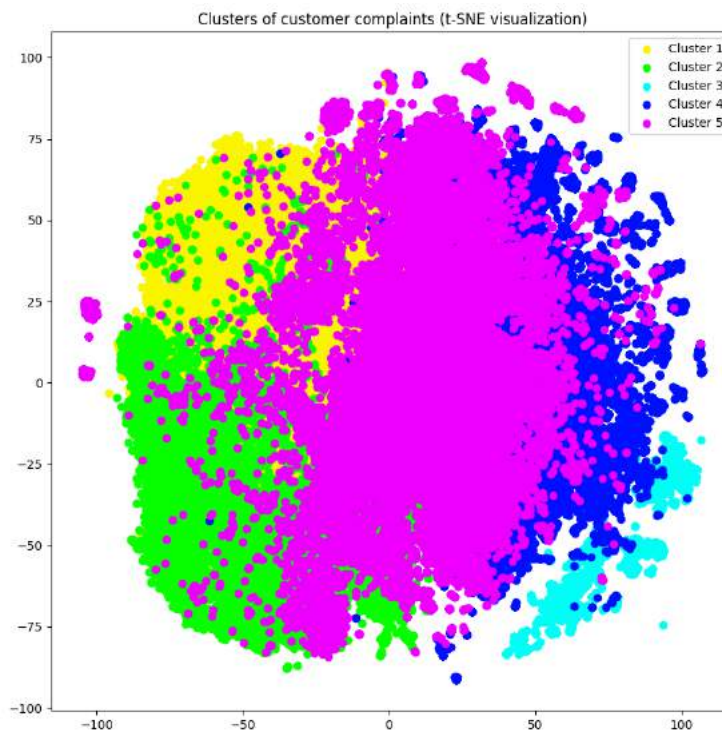
- Among the four models, hierarchical clustering with Word2Vec embeddings achieves the highest Silhouette Score of 0.0704. This indicates better-defined and well-separated clusters compared to the other models.

- The Calinski-Harabasz Score is also highest for hierarchical clustering with Word2Vec, with a score of 15511.66. This suggests that the clusters are more compact and well-separated.

- Hierarchical clustering with BERT embeddings has a moderate Silhouette Score of 0.0279 and a Calinski-Harabasz Score of 4336.72. It performs better than the TF-IDF and Doc2Vec models but is inferior to the Word2Vec model.



**Figure 4.7:** The dendrogram visualizes the hierarchical clustering of word2vec.



**Figure 4.8:** t-SNE visualization of clusters generated by hierarchical clustering with word2vec.

Referring to the generated topics, BERT embeddings seemed to highlight more specific transactional aspects, like payment, loans, and bank interactions, likely due to BERT's superior understanding of contextual language usage. Word2Vec, on the other hand, brought up themes that were slightly broader, covering aspects like consumer interactions and reporting along

with the financial transactions, indicating its more general semantic understanding. TF-IDF, known for its emphasis on identifying unique words, unveiled themes around rights violations, consumer protection, and identity theft, showing its effectiveness in spotlighting less common but important topics in the dataset.

In conclusion, based on the evaluation scores, hierarchical clustering with Word2Vec embeddings yields the best results in terms of cluster separation and compactness. It provides more distinct and well-defined clusters compared to the other models. Thus, the results suggest that models such as Word2Vec, which capture semantic relationships at the word level, are better suited for clustering textual data compared to models like TF-IDF and Doc2Vec.

## 5. Conclusion

### 5.1 Answering the data science question

This study presents a comprehensive analysis on the performance of various embedding methods (TF-IDF, Word2Vec, Doc2Vec, and BERT) and topic modeling techniques (K-means, LDA, BERTopic, and Hierarchical Clustering) when applied to the classification of financial complaint data.

Notably, the integration of Word2Vec embeddings with K-means clustering methodology yielded the highest Silhouette Score and Calinski-Harabasz Index. This underlines that Word2Vec coupled with K-means clustering successfully demarcated well-defined and densely aggregated clusters, demonstrating superior performance compared to other tested combinations. Although K-means clustering paired with BERT embeddings also produced respectable results, the performance metrics were not as impressive as the combination with Word2Vec.

Meanwhile, K-means clustering implemented with TF-IDF and Doc2Vec embeddings presented lower metric scores, indicating that the resulting clusters were less distinctive and sparse. Similarly, during the application of Latent Dirichlet Allocation (LDA), it was discovered that the choice of representation (BoW, TF-IDF, Word2Vec) significantly influences the coherence scores, hence affecting the interpretability of topics.

In an interesting turn of events, the application of BERTopic with DBSCAN and the "all-MiniLM-L6-v2" sentence embedding model surpassed the performance of the "bert-base-nli-mean-tokens" model, as evident from the higher Silhouette Scores despite the fact that both models resulted in bad performance (negative Silhouette Scores). Hierarchical clustering combined with Word2Vec embeddings outperformed the other combinations in terms of Silhouette Score and Calinski-Harabasz Score, suggesting well-separated

and compact clusters.

## 5.2 Answering the research question

The application of various NLP methods allowed for a nuanced and diverse understanding of the financial complaint data. High-frequency terms identified included "complaint", "transunion", "ID", "card", "information", "payment", and "call". This finding underscores key themes persistently present across customer complaints.

Furthermore, potential issues related to payment processing, customer service interactions, and information handling were extrapolated based on these recurring terms. Utilizing Word2Vec and BERT embeddings in conjunction with K-means clustering permitted the identification of specific themes tied to each cluster, facilitating a deeper comprehension of the nature of customer grievances.

Moreover, the implementation of LDA using different text representations (BoW, TF-IDF, Word2Vec) unveiled a variety of topics. These encompassed areas like loans, payments, communication with banks, and debt collection practices, illustrating the broad range of issues faced by customers. Remarkably, BERTopic coupled with DBSCAN brought to light more precise and unique topics, which investigated finer details relating to various financial aspects.

Hierarchical clustering offered insights into more specific transactional aspects, such as payment, loans, and bank interactions when used with BERT embeddings, or broader themes covering consumer issues when used with Word2Vec embeddings.

Ultimately, the selection of the NLP method significantly impacts the understanding and interpretation of financial complaint data. While certain methods provided a macroscopic overview, others unearthed more detailed and specific themes. Specifically, Word2Vec with K-means or Hierarchical Clustering, and BERTopic with the "all-MiniLM-L6-v2" model, surfaced more useful and distinct insights into the nature of the customer complaints,

demonstrating the utility of these approaches in the analysis of complaint data.

### **5.3 Describing implications for the proper domain setting**

The analysis results offer valuable insights for the banking industry, providing the following strategies for improving customer satisfaction:

- Enhancing communication and transparency: Prioritize clear and effective communication with customers, ensuring timely and accurate information about accounts, loans, and services. Improve communication channels and responsiveness to customer queries or concerns.

- Credit card services and credit reporting: Pay attention to credit card operations and accurate credit reporting. Implement proactive measures to address customer concerns and provide reliable credit card services.

- Focus on customer experience: Utilize topic modeling insights to understand underlying complaint topics and issues. Develop strategies to enhance the overall customer experience, including targeted employee training, process improvements, and new service offerings based on identified customer needs.

- Proactive fraud prevention: Invest in robust fraud detection and prevention systems to protect customer accounts and personal information. Take proactive measures to prevent fraud and enhance customer trust and satisfaction.

- Continuous analysis and monitoring: Regularly analyze customer complaints and monitor emerging trends or patterns. Implement feedback mechanisms, conduct customer surveys, and utilize advanced NLP techniques to monitor customer sentiment and identify potential issues proactively.

By implementing these strategies, the banking industry can prioritize customer-centric initiatives, streamline processes, and proactively address customer concerns. This will ultimately lead to improved customer satis-

faction, loyalty, and overall business success. [45]

## **5.4 Discussing ethical implications and consideration**

The analysis of customer complaints in the banking sector through natural language processing techniques presents significant opportunities to gain insights into customer experiences and improve overall customer satisfaction. However, it is essential to acknowledge and address the ethical considerations associated with such research.

The following three key ethical issues referring to the research and the findings are examined in detail: bias and fairness, transparency and accountability, and responsible use of findings. Each of these issues plays a crucial role in ensuring the ethical conduct of the research and the responsible utilization of the results. By exploring these concerns and implementing appropriate measures, this study strives to promote ethical practices and uphold the rights and well-being of individuals involved in the dataset.

### **Bias and Fairness**

In addressing the ethical concern of bias and fairness in this study, extensive measures were taken to mitigate potential biases in the analysis of customer complaints. The selection of appropriate NLP methods, such as careful consideration of embeddings and topic modeling techniques, aimed to reduce biases and promote fairness in the analysis of the complaint data. To further enhance fairness and transparency, the evaluation metrics used in the study were chosen to provide a comprehensive assessment of different models' performances, considering various aspects of clustering and topic coherence. [46] [47]

### **Transparency and Accountability**

Transparency and accountability were key considerations throughout the research process. Efforts were made to document and describe the methodologies, techniques, and tools employed in the analysis of customer com-

plaints. This includes providing clear explanations of the chosen NLP methods, such as K-means, LDA, BERTopic and hierarchical clustering, and their respective evaluations. Furthermore, the research findings were presented in a manner that allows for easy interpretation and understanding by different stakeholders, such as banking industry professionals and regulators. By providing detailed insights into the clustering results, topic models, and associated metrics, this study aims to foster transparency and facilitate discussions on the findings and their implications. [48] [49]

### **Responsible Use of Findings**

The responsible use of the research findings was of paramount importance in this study. The insights gained from analyzing customer complaints in the banking sector should be used to drive positive change and improvements in customer satisfaction. These findings can serve as a valuable resource for banking institutions to identify pain points, address recurring issues, and enhance their services and processes. However, it is crucial to consider the ethical and legal implications of implementing any changes based on these findings. Decisions and actions taken by the banking industry should prioritize the well-being and rights of their customers, while adhering to applicable data protection and privacy regulations. Responsible data governance practices, including obtaining informed consent, ensuring data security, and maintaining transparency, should be upheld when utilizing the research findings for customer satisfaction improvement strategies. [50]

Overall, this study has endeavored to address the ethical considerations associated with customer complaint analysis in the banking sector, paving the way for meaningful research outcomes that prioritize ethical practices and respect the rights and privacy of individuals.

## **5.5 Limitations**

Every research study has its limitations, and it is important to acknowledge them to provide a comprehensive understanding of the scope and potential



constraints of the findings. In the context of this thesis, several limitations should be considered, including the processing time required for the analysis and the challenges associated with selecting appropriate evaluation metrics for unsupervised learning.

One important limitation to consider is the significant processing time required for the analysis. Working with a sizable dataset, even after reducing it to a 10% sample, led to considerable computational overhead. The substantial computational demands of the analysis should be taken into account when considering the practicality of applying these models to larger datasets or real-time systems. The lengthy processing time required for calculations and modeling can hinder the scalability and efficiency of the proposed methods. Balancing accuracy and processing time becomes a critical trade-off, requiring careful consideration. Efforts should be made to optimize the algorithms, leverage computational resources, and explore parallel processing techniques to reduce the computational burden. By addressing these challenges, the models can become more feasible and applicable in real-life scenarios.

Furthermore, another limitation lies in the selection of appropriate evaluation metrics for unsupervised learning, specifically in the context of clustering. Unlike supervised learning models, unsupervised models lack a definitive "ground truth" for comparison. Therefore, proxy metrics are utilized for hyperparameter optimization and model comparison. In this thesis, the Coherence Score, Silhouette Score, and Calinski-Harabasz Score were chosen as evaluation metrics. However, it is important to recognize the limitations associated with these metrics. For instance, the Coherence Score heavily depends on the quality and suitability of the word embedding model, which can lead to misleading results with poor or incompatible embeddings. The Silhouette Score may exhibit limitations when dealing with complex-shaped or density-based clusters, and it can be computationally expensive for large datasets. The Calinski-Harabasz Score assumes convex and isotropic clusters, which may not always hold true in real-world data. Awareness of these limitations is crucial when interpreting the results and considering the practical implications of the clustering outcomes.

While the limitations related to processing time and evaluation metrics should be acknowledged, they do not invalidate the overall findings and recommendations of this thesis. It is essential to be aware of these limitations and approach the results with a critical mindset. By leveraging the insights and strategies presented in this thesis while accounting for these limitations, the banking industry can make significant strides towards improving customer satisfaction and enhancing overall business success.

## 5.6 Future work

The limitations identified in this thesis open up avenues for future research and improvement in the field of customer complaint analysis in the banking industry. Here are some potential areas for further investigation:

- Evaluation Metrics Enhancement: As unsupervised learning models lack a definitive ground truth, there is a need for the development of more robust and reliable evaluation metrics for clustering tasks. Future research could explore the design of metrics that capture the intrinsic characteristics of different types of clusters, including non-convex or density-based clusters. Additionally, investigating ensemble-based or consensus-based evaluation methods could provide a more comprehensive and reliable assessment of clustering performance. It is essential to continue the exploration and refinement of evaluation metrics to ensure accurate and meaningful evaluation of clustering algorithms.

- Integration of Real-time Data: While the analysis in this thesis focused on a static dataset of customer complaints, future work could explore the integration of real-time data sources, such as social media feeds or online customer interactions. By incorporating real-time data, researchers and practitioners can gain more timely insights into customer sentiments and emerging issues. This could enable banks to proactively address customer concerns, enhance service offerings, and improve overall customer satisfaction in a dynamic and rapidly evolving environment.

- Contextual Analysis: This thesis primarily focused on the analysis of

customer complaints in a general sense. Future research could delve deeper into the contextual analysis of complaints, considering factors such as customer demographics, geographical location, or specific product or service categories. Understanding the nuances and variations in customer complaints across different contexts can provide more targeted and tailored strategies for improving customer satisfaction and addressing specific pain points.

By exploring these avenues for future work, researchers can overcome the limitations identified in this thesis and further enhance the effectiveness and applicability of customer complaint analysis in the banking industry. These advancements can contribute to the development of more accurate, efficient, and actionable insights for banks to better serve their customers and optimize their operations.

# A. Appendix

## A.1 Annotated scripts and results of analyses and method settings

### A.1.1 Data cleaning

```
# Remove NaNs and duplicates
data = data[['complaint']].dropna()
data = data.drop_duplicates()

# Keep only a sample of the dataset since the whole one equals to 1+ million rows.
sampled_df = data.sample(frac=0.1, random_state=42)

# Initialize a WordNetLemmatizer object to lemmatize words
lemmatizer = WordNetLemmatizer()

# Define the list of stopwords by NLTK
english_stopwords = set(nltk.corpus.stopwords.words('english'))

# Define a list of additional stopwords that are not part of NLTK's list (Xs coming from dates and 3 most common words)
additional_stopwords = ['xx', 'xxxx', 'xxxxxx', 'xxxxxxxx', 'xxxxxxxxxxxx', 'account', 'report', 'credit']

# Function to map NLTK's part-of-speech tags to WordNet's part-of-speech tags
def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN # Default to noun if no match is found

# Function to preprocess text
def preprocess_text(text):
    # Remove special characters and numbers using regular expressions
    text = re.sub('[^a-zA-Z]', ' ', text)

    # Tokenize the text into individual words
    tokens = nltk.word_tokenize(text)

    # Tag each token with its part of speech (e.g. noun, verb, adjective, etc.)
    tagged_tokens = nltk.pos_tag(tokens)

    # Lemmatize tokens using their part-of-speech tag
    lemmatized_tokens = [lemmatizer.lemmatize(token.lower(), get_wordnet_pos(tag)) for token, tag in tagged_tokens]

    # Remove stopwords from the lemmatized tokens
    filtered_tokens = [token for token in lemmatized_tokens if token.lower() not in english_stopwords and token.lower() not in additional_stopwords]

    # Filter out any tokens that are less than 2 characters long
    preprocessed_tokens = [token for token in filtered_tokens if len(token) > 1]

    # Join the preprocessed tokens back together into a single string
    preprocessed_text = ' '.join(preprocessed_tokens)

    # Remove any extra spaces
    preprocessed_text = ' '.join(preprocessed_text.split())

    # Return the preprocessed text
    return preprocessed_text

# Apply the text preprocessing function to the 'complaint' column of the DataFrame
sampled_df['complaint'] = sampled_df['complaint'].apply(preprocess_text)
```

Figure A.1: Pre-processing step.

## A.1.2 Feature Extraction

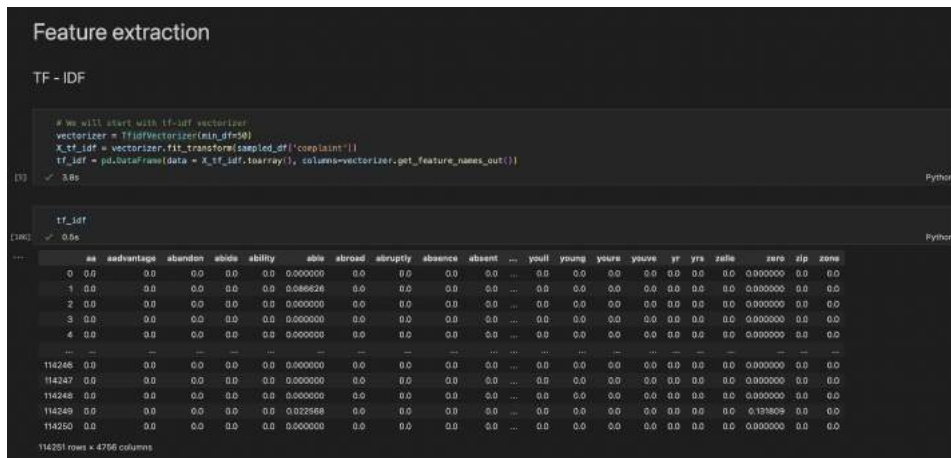


Figure A.2: Feature Extraction of TF-IDF vector.

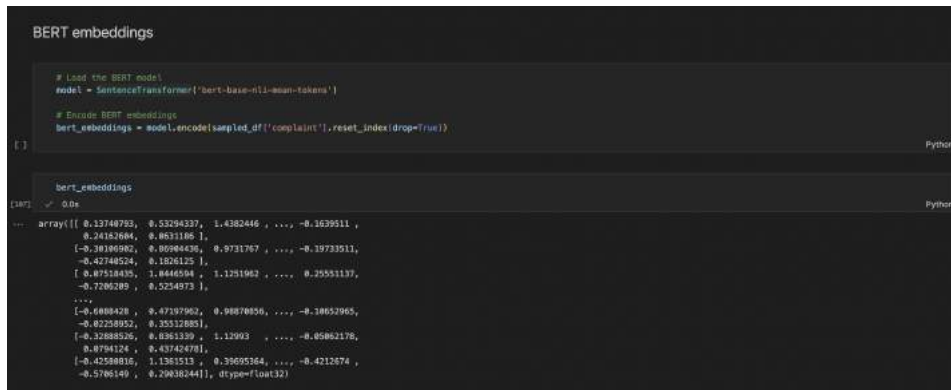


Figure A.3: Feature Extraction of BERT embeddings.

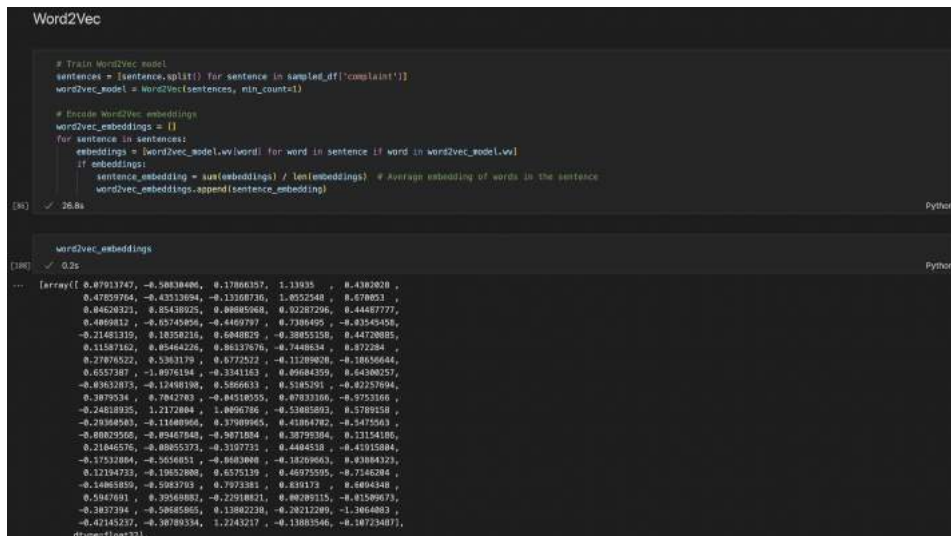


Figure A.4: Feature Extraction of Word2vec vector.

```

Doc2Vec

# Tokenize your preprocessed 'complaint' column into words
data_doc2vec = sampled_df['complaint'].apply(nltk.word_tokenize)

# Generate tagged documents
documents = [TaggedDocument(doc, [1]) for i, doc in enumerate(data_doc2vec)]

# Define a Doc2Vec model
model = Doc2Vec(vector_size=100, min_count=1, window=2, seed=42, workers=4)

# Build the vocabulary
model.build_vocab(documents)

# Train the model on your data
model.train(documents, total_examples=model.corpus_count, epoch=model.epochs)

# Extract vectors from the model
doc_vectors = [model.infer_vector(doc.words) for doc in documents]
    
```

Python

```

Python
doc_vectors
array([-2.87760881e-01,  7.27586578e-02, -1.59807445e-01,  4.77418572e-01,
        2.8238844e-01,  1.4939681e-01, -8.62383246e-02,  1.14876051e-01,
       -4.72377636e-02,  3.84283679e-01,  4.37627862e-02,  2.97844443e-02,
        7.68718165e-02,  9.49692958e-02,  4.32215212e-03,  3.23743228e-01,
       -1.23385482e-01, -1.5895954e-01,  1.28768739e-01,  6.5825289e-03,
       -2.28842382e-01,  5.89901851e-02, -3.27223711e-01,  4.69955169e-02,
        1.68798474e-01,  1.48627355e-01, -3.28837942e-01, -2.2238882e-01,
       -4.56238811e-01,  4.21723888e-01,  2.26871476e-01,  2.88018099e-01,
        1.98653886e-01,  3.63643765e-02, -9.24473485e-02,  4.69577983e-02,
       -1.28831746e-01, -1.46911582e-01, -1.79484891e-01, -1.82888695e-01,
        2.50815326e-01, -9.7783865e-02,  2.81388878e-01,  2.3933398e-01,
        4.99282311e-01, -1.89842337e-01,  1.81687345e-01,  1.63982682e-01,
       -1.31978895e-01,  1.33387521e-01,  2.24767789e-01,  3.38625743e-01,
        3.35576832e-02,  5.96561693e-01,  1.55637771e-01, -2.55478122e-01,
    
```

Figure A.5: Feature Extraction of Doc2vec vector.

### A.1.3 k-Means

The following procedure focuses on k-Means for TF-IDF vector. The rest representations (BERT embeddings, Word2Vec and Doc2Vec) follow the same pattern.

```

# Perform the elbow method to find the optimal number of clusters for KMeans clustering on the tf-idf features
inertia = []
K = range(1, 20)
for k in K:
    kmeansModel = KMeans(n_clusters=k, random_state=0)
    kmeansModel.fit(tf_idf)
    inertia.append(kmeansModel.inertia_)

kn = Kneelocator(K, inertia, curve='convex', direction='decreasing')
print(kn.knee)
    
```

Python

Figure A.6: Elbow method to find the optimal number of clusters.

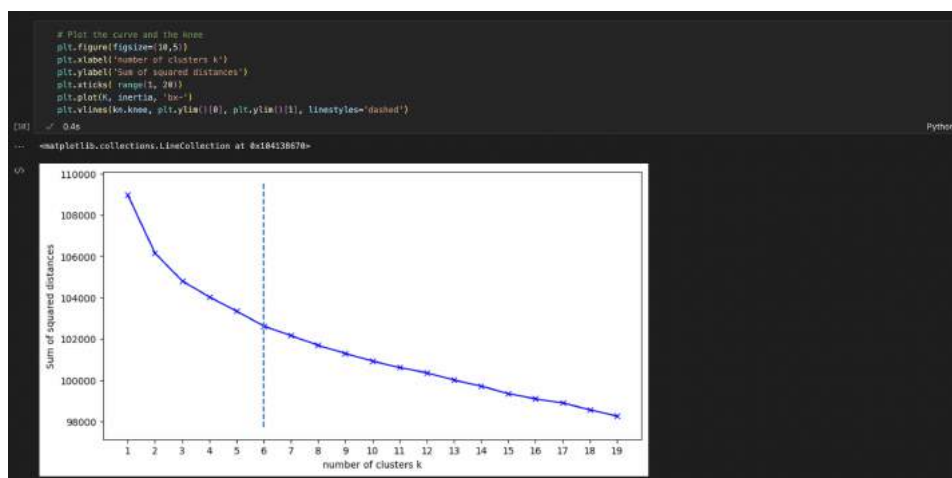


Figure A.7: Plot of the curve and the knee.

```

# Perform K-means clustering
clusters = kn.kmeans
kmeansModel = KMeans(n_clusters=clusters, init='k-means++', max_iter=300, random_state=0)
nod = kmeansModel.fit_transform(tf_idf)
sampled_idf['tf_idf'] = kmeansModel.predict(tf_idf)

labels = kmeansModel.labels_

# Calculate evaluation metrics
silhouette_avg = silhouette_score(tf_idf, labels)
calinski_score = calinski_harabasz_score(tf_idf, labels)

# Print evaluation metrics
print("Silhouette Score:", silhouette_avg)
print("Calinski-Harabasz Index:", calinski_score)

Silhouette Score: 0.8231779344836864
Calinski-Harabasz Index: 1413.8751517492735

```

Figure A.8: Perform of the model and calculation of the evaluation metrics.

```

# Top 15 words per cluster
# Get centroids = kmeansModel.cluster_centers_.argsort()[:, 1:-1]
terms = vectorizer.get_feature_names_out()
dict = {}
for i in range(clusters):
    print("Cluster %i, terms:" % i)
    for ind in order_centroids[i, 1:-1]:
        print(terms[ind], sep=" ", end=" ")
    print("")

0, information, receive, consumer, item, dispute, reporting, bureau, identity, file, balance, theft, equifax, please, inaccurate, delete,
1, payment, loan, late, pay, mortgage, make, month, call, time, would, due, interest, tell, get, day,
2, debt, collection, company, owe, letter, send, validation, collect, receive, agency, pay, call, dispute, request, information,
3, section, right, reporting, consumer, state, privacy, furnish, violate, agency, instruction, accordance, fair, act, write, without,
4, card, call, bank, charge, get, tell, say, receive, money, pay, would, check, time, company, send,
5, inquiry, hard, authorize, receive, unauthorized, date, company, fraudulent, equifax, transmission, please, information, following, pull, dispute,

# Summary of the distribution of words across the clusters
sampled_idf.groupby('tf_idf').count().reset_index()['tf_idf', 'complaint']

tf_idf  complaint
0       0      27674
1       1      19810
2       2      12591
3       3       4865
4       4      44888
5       5       4423

```

Figure A.9: Top 15 words per cluster and summary of the distribution of words across the clusters

```

# Fit and transform with t-SNE
t_sne = TSNE()
t_sne_embeddings = t_sne.fit_transform(tf_idf)

# Assign cluster colors
colors = plt.cm.rainbow(np.linspace(0, 1, clusters))

# Create a new figure
plt.figure(figsize=(10, 10))

# Plot points for each cluster with unique color and add it to the legend
for cluster_num in range(clusters):
    # Logical mask for the points belong to the current cluster
    mask = labels == cluster_num

    # Plot points belong to the current cluster only, with a unique color, and add it to the legend
    plt.scatter(t_sne_embeddings[mask, 0], t_sne_embeddings[mask, 1], color=colors[cluster_num], label=f'Cluster {cluster_num}')

# Calculate percentiles for x and y axes
x_low, x_high = np.percentile(t_sne_embeddings[:, 0], [0.01, 99.99])
y_low, y_high = np.percentile(t_sne_embeddings[:, 1], [0.01, 99.99])

# Show the plot
plt.xlim(x_low, x_high)
plt.ylim(y_low, y_high)
plt.legend()
plt.title('t-SNE visualization of clusters')
plt.show()

```

Figure A.10: Declaration of the t-SNE visualization.

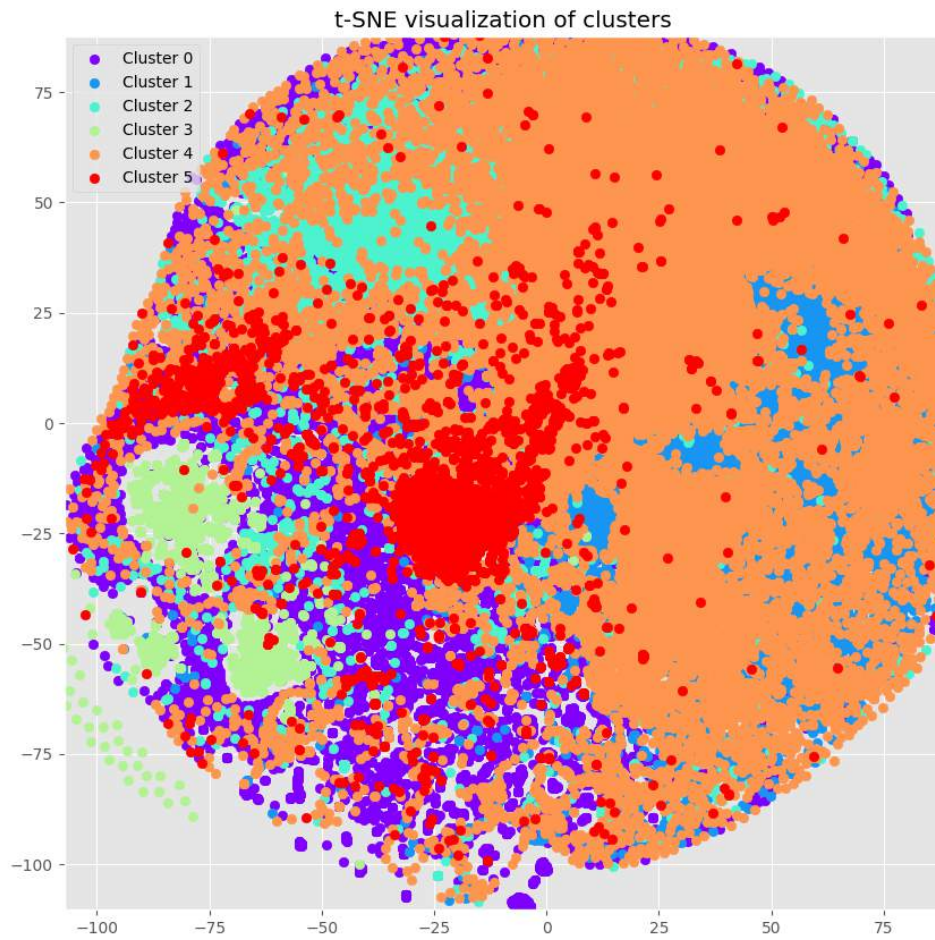


Figure A.11: t-SNE plot visualization.

#### A.1.4 LDA

The following procedure focuses on LDA for Bag-of-Words vector. The rest representations (BERT embeddings and Word2Vec) follow the same pattern.



```

# This function calculates coherence values for the LDA model with different numbers of topics, and different alpha and beta parameters.
# This is achieved by training multiple LDA models (using efficient multicore parallelism) with different configurations and comparing their coherence scores.

def coherence_values(dictionary, corpus, texts, limit, start, step, alpha, beta):
    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        for a in alpha:
            for b in beta:
                print('Calculating for', num_topics, 'topics, alpha=', a, ', beta=', b)
                # Train an LDA model with current parameters
                model = LdaMulticore(corpus=corpus, num_topics=num_topics, id2word=dictionary, alpha=a, eta=b, workers=min(6, multiprocessing.cpu_count() - 1))
                model_list.append(model)

                # Calculate coherence score and append to list
                coherence_model = CoherenceModel(model=model, texts=texts, dictionary=dictionary, coherence='c_v')
                coherence_values.append((coherence_model.get_coherence(), num_topics, a, b))
    return model_list, coherence_values

# List of alpha and beta values to try
alpha = list(np.arange(0.01, 1, 0.3))
beta = list(np.arange(0.01, 1, 0.3))

bow_complaints = sampled_df['complaint'].apply(nltk.word_tokenize)

# Create dictionary
id2word = corpora.Dictionary(bow_complaints)
id2word.compactify()

# Create corpus
corpus = [id2word.doc2bow(doc) for doc in bow_complaints]

# Define start, limit and step for range of topic numbers to try
start = 5
limit = 20
step = 1

# Calculate coherence values for different configurations
model_list, coherence_values = coherence_values(id2word, corpus=corpus, texts=bow_complaints, start=start, limit=limit, step=step, alpha=alpha, beta=beta)

# Find the configuration with the highest coherence score
max_coherencebow, optimal_num_topicsbow, optimal_alphabow, optimal_betabow = max(coherence_values, key=lambda item:item[0])

```

**Figure A.12:** This function calculates coherence values for the LDA model with different numbers of topics, and different alpha and beta parameters.

```

# Create a list to store the maximum coherence value for each number of topics
max_coherence_per_topic = []

# For each number of topics
for num_topics in range(start, limit, step):
    # Get the coherence values for the current number of topics
    current_coherence_values = [cv for cv, nt, a, b in coherence_values if nt == num_topics]

    # Find and store the maximum coherence value
    max_coherence_per_topic.append(max(current_coherence_values))

# Create the x data for the plot
x = range(start, limit, step)

plt.figure(figsize=(12, 12))
plt.plot(x, max_coherence_per_topic)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values()", "best"))
plt.vlines([np.argmax(max_coherence_per_topic)], plt.ylim()[0], plt.ylim()[1], linestyle='dashed')
plt.show()

```

**Figure A.13:** Create a list to store the maximum coherence value for each number of topics and then the plot.

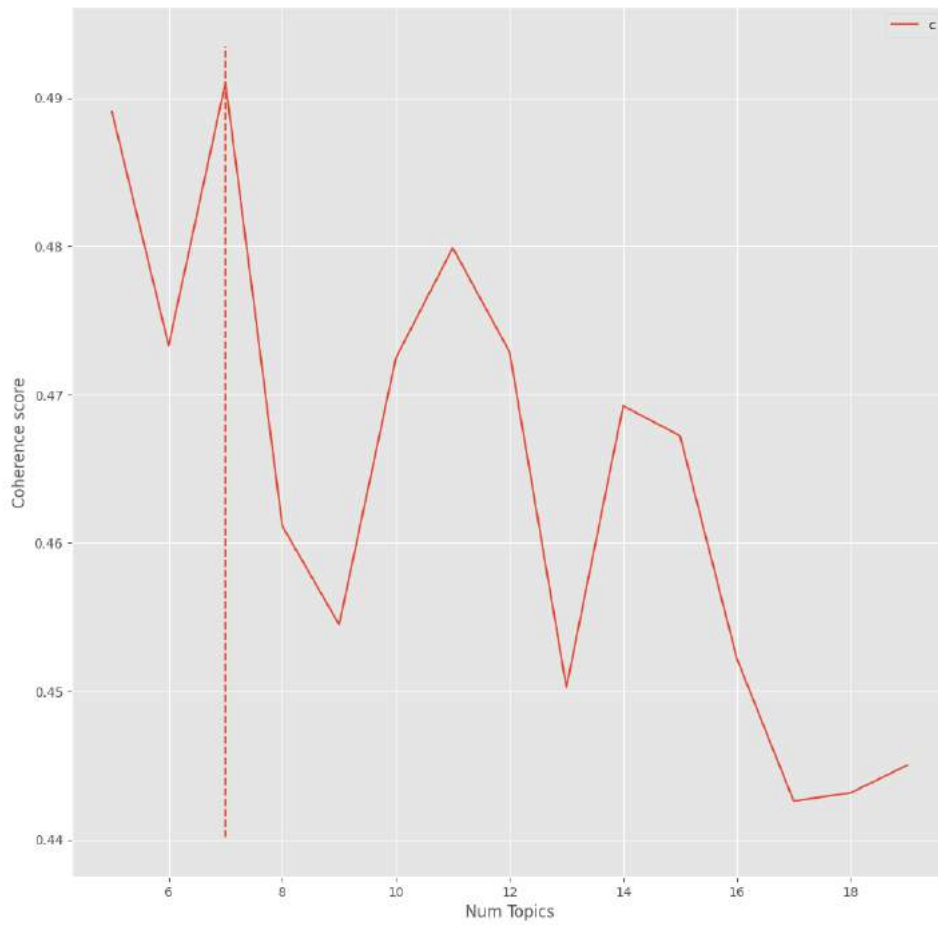


Figure A.14: The plot of coherence value for each number of topics.

```

# Train a final LDA model with optimal configuration and print the topics
sol = LDAMulticore(corpus=corpus, num_topics=optimal_num_topicsbow, id2word=id2word, alpha=optimal_alphabow, eta=optimal_betabow, workers=min(8, multiprocessing.cpu_count()) - 1)
model.print_topics()

[54] ✓ 98s Python
...
[[0,
  '0.099*send' + 0.089*loan' + 0.008*receive' + 0.007*bank' + 0.007*day' + 0.007*call' + 0.007*claim' + 0.007*file' + 0.005*request' + 0.005*complaint'],
 1],
 ['0.019*call' + 0.017*loan' + 0.034*payment' + 0.012*pay' + 0.011*would' + 0.010*tell' + 0.009*make' + 0.009*information' + 0.009*time' + 0.007*charge'],
 2],
 ['0.038*payment' + 0.019*balance' + 0.015*information' + 0.014*late' + 0.010*make' + 0.009*pay' + 0.009*remove' + 0.008*please' + 0.007*time' + 0.007*day'],
 3],
 ['0.016*call' + 0.012*bank' + 0.011*payment' + 0.010*would' + 0.010*pay' + 0.010*tell' + 0.009*receive' + 0.009*loan' + 0.009*time' + 0.008*make'],
 4],
 ['0.012*get' + 0.011*tell' + 0.011*call' + 0.011*card' + 0.010*time' + 0.010*would' + 0.008*day' + 0.008*send' + 0.007*loan' + 0.007*bank'],
 5],
 ['0.013*debt' + 0.012*collection' + 0.012*payment' + 0.012*pay' + 0.010*receive' + 0.009*send' + 0.009*never' + 0.009*company' + 0.008*information' + 0.008*letter'],
 6],
 ['0.024*information' + 0.023*consumer' + 0.015*reporting' + 0.013*section' + 0.012*debt' + 0.011*state' + 0.011*agency' + 0.010*inquiry' + 0.010*right' + 0.008*dispute']]

```

Figure A.15: Train the final LDA model with optimal configuration and the topics.

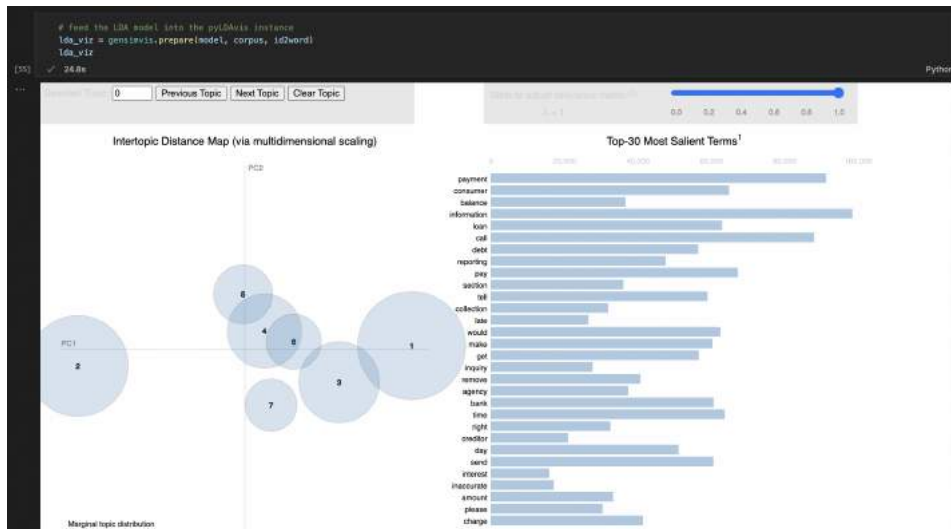


Figure A.16: The LDA model into the pyLDAvis instance.

### A.1.5 BERTopic

The following procedure focuses on BERTopic for SentenceTransformer('bert-base-nli-mean-tokens'). The other implementation of SentenceTransformer('all-MiniLM-L6-v2') follow the same pattern.

```

# sentence-transformers/all-MiniLM-L6-v2 is a popular high-performing model that creates 384-dimensional sentence embeddings.
embedding_model = SentenceTransformer('bert-base-nli-mean-tokens')

# UMAP is a dimensionality reduction technique which focuses on preserving the neighborhood relationships between data points,
# aiming to maintain the local structure rather than the global structure
umap_model = UMAP(n_neighbors=3, n_components=20, min_dist=0.1)

# HDBSCAN works by refining clusters based on density reachability. It considers dense regions of data points as clusters and identifies sparser regions as noise.
hdbscan_model = HDBSCAN(min_cluster_size=80, min_samples=30,
                        gen_min_span_tree=True,
                        prediction_data=True)

# TF-IDF representation
tfidf_vectorizer = TfidfVectorizer(ngram_range=(1, 2))

# Now we define our BERTopic model and fit to our data
model_tfidf = BERTopic(
    umap_model=umap_model,
    hdbscan_model=hdbscan_model,
    embedding_model=embedding_model,
    vectorizer_model=tfidf_vectorizer,
    top_n_words=5,
    language='english',
    calculate_probabilities=True,
    verbose=True)

topics_tfidf, probs_tfidf = model_tfidf.fit_transform(sampled_df['complaint'])

# Get embeddings for the test data
embeddings = embedding_model.encode(sampled_df['complaint'].tolist(), convert_to_tensor=True)

# Ensure topics are in a NumPy array and non-negative for silhouette_score
topics_np_tfidf = np.array(topics_tfidf)
topics_np_tfidf = topics_np_tfidf - np.min(topics_np_tfidf)

# Calculate silhouette Score
sil_score_tfidf = silhouette_score(embeddings.cpu(), topics_np_tfidf)

```

Figure A.17: Definition of the BERTopic model.

## A.1 Annotated scripts and results of analyses and method settings

```
freq = model.get_topic_info()
freq.head(15)
```

Topic	Count	Name	Representation	Representative_Docs
0	1 59186	1_Information_payment_debt_consumer	[information, payment, debt, consumer, company]	[intentionally review formal work compose declar...
1	0 5383	0_report_information_dispute_remove	[report, information, dispute, remove, inquiry]	[hear victim identify theft identity fraud loc...
2	1 3601	1_loan_student_student_loan_payment	[loan, student, student loan, payment]	[begin repay student loan graduate college obt...
3	2 2901	2_mortgage_loan_home_modification	[mortgage, loan, home, modification, property]	[move purchase home late mortgage cash deposit...
4	3 2912	3_call_phone_debt_number	[call, phone, debt, number, ask]	[past month receive phone call seem like diff...
5	4 2508	4_chase_chase_bank_card_bank	[chase, chase bank, card, bank, check]	[chase bank deposit customer chase card custom...
6	5 2142	5_car_vehicle_lease_payment	[car, vehicle, lease, payment, pay]	[date purchase deliver vin tag describe compl...
7	6 1643	6_inquiry_authorize_unauthorized_remove	[inquiry, authorize, unauthorized, remove, hard]	[inquiry unknown company please remove inquiry...
8	7 1476	7_late_late_payment_payment_day_late	[late, late payment, payment, day late, day]	[show late payment never late, day late payment...
9	8 1029	8_debt_validation_letter_send	[debt, validation, letter, send, request]	[unverified previously dispute never business ...
10	9 901	9_paypal_money_transaction_use_paypal	[paypal, money, transaction, use paypal, email]	[sway since synchrony bank take paypal reques...
11	10 959	10_fargo_well_fargo_well_check	[fargo, well fargo, well, check, bank]	[identity theft documentation summary unauthor...
12	11 942	11_bankruptcy_court_bankruptcy_court_verify	[bankruptcy, court, bankruptcy court, verify, ...]	[dispute chapter bankruptcy file error request...
13	12 819	12_consumer_reporting_item_information_consume...	[consumer reporting, item information, consume...	[accordance fair reporting act list violate fe...
14	13 788	13_section_right_violate_right_accordance_fair	[section, right, violate right, accordance fair...	[accordance fair reporting act violate right s...

Figure A.18: Generated topics along with representative words.

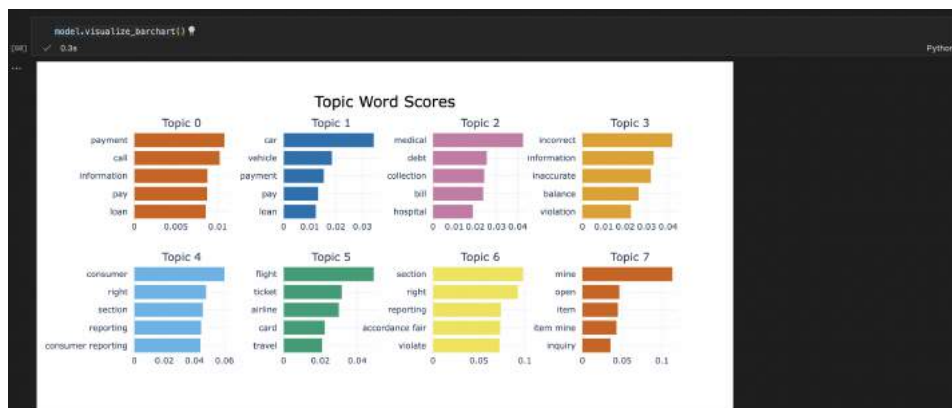


Figure A.19: Barchar for each topic along with top words.

### A.1.6 Hierarchical clustering

The following procedure focuses on Hierarchical clustering for BERT embeddings. The rest representations (TF-IDF, Word2Vec and Doc2Vec) follow the same pattern.

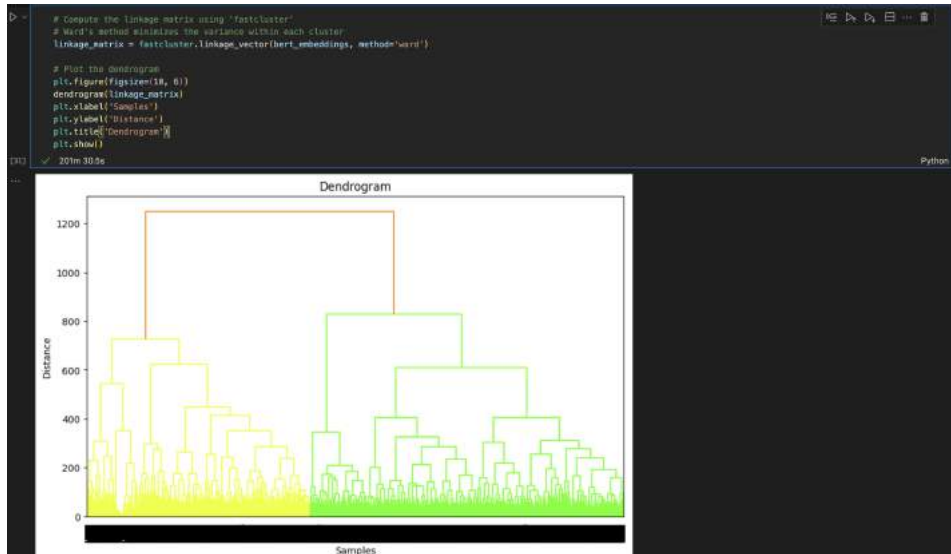


Figure A.20: The linkage matrix using 'fastcluster' and the dendrogram

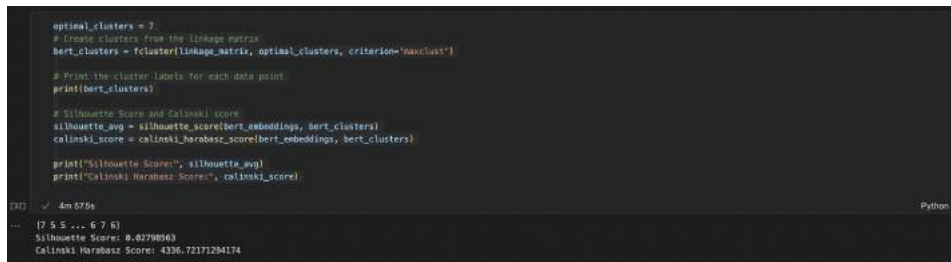


Figure A.21: Create clusters from the linkage matrix and calculate evaluation scores



Figure A.22: Definition of the t-SNE plot.

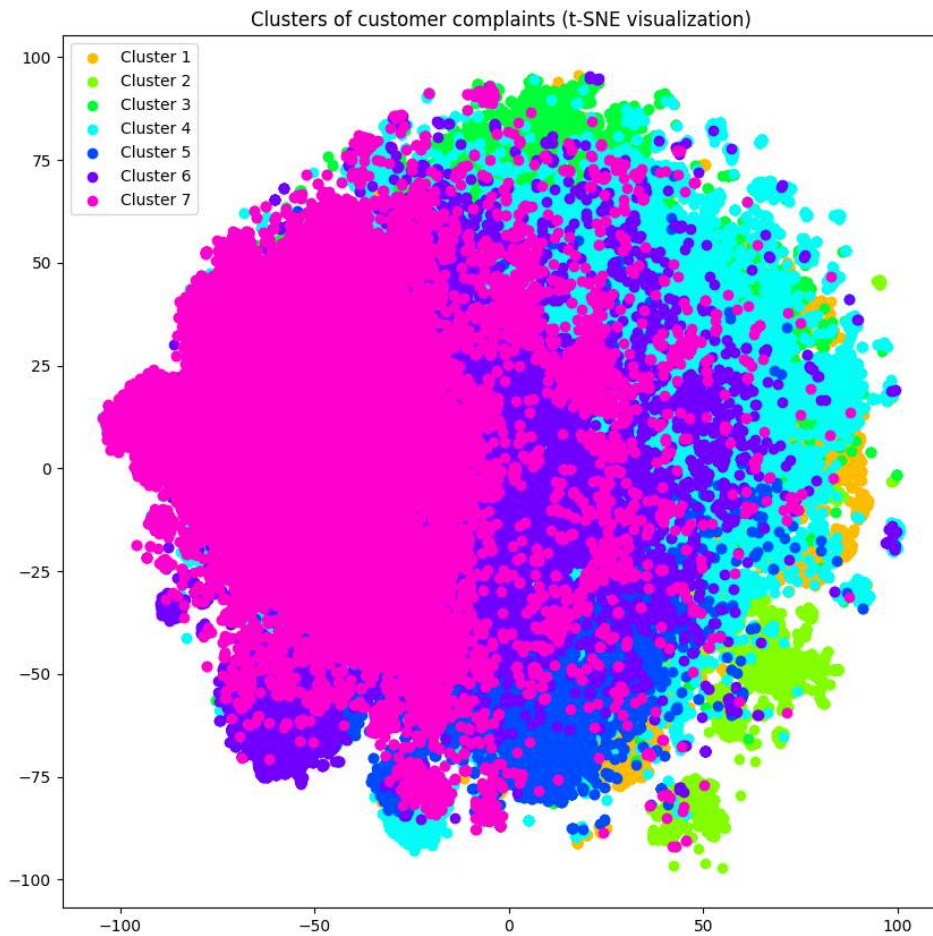


Figure A.23: t-SNE plot for generated clusters.

```

# Tokenizer to break text into words
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')

# Create a dictionary to store the cluster sentences
clusters = {}

# Assuming sampled_df[complaint] contains the original sentences
sentences = sampled_df['complaint'].tolist()

for i, label in enumerate(bert_clusters):
    if label not in clusters:
        clusters[label] = []
    if i < len(sentences): # Add this condition to check if the index is within the range of sentences
        clusters[label].append(sentences[i]) # Append the data point to the corresponding cluster

# For each cluster, combine all sentences, tokenize, and count word frequencies
for cluster_id, data_points in clusters.items():
    # Combine all sentences in the cluster into a single string
    combined_text = ' '.join(data_points)

    # Tokenize the text to get individual words
    words = tokenizer.tokenize(combined_text.lower())

    # Count the frequency of each word
    word_count = Counter(words)

    # Print the top 15 words for this cluster
    print(f"Cluster {cluster_id}:")
    for word, count in word_count.most_common(15):
        print(f"Word: {word}, Count: {count}")
    print("-----")
    
```

Figure A.24: Function which print the clusters along with sorted top words and each word's number of appearances

```
... Cluster 7:
payment: 52720
loan: 42816
call: 41193
pay: 35569
would: 28142
time: 28728
tell: 28891
receive: 26397
make: 26806
get: 24908
month: 24581
mortgage: 23498
bank: 22347
say: 21827
send: 20983
-----
Cluster 5:
never: 13483
debt: 18026
information: 8458
call: 8216
receive: 8736
company: 8546
send: 7516
...
time: 1394
send: 1303
verified: 1386
-----
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.
```

**Figure A.25:** Clusters along with sorted top words along with each word's number of appearances

# Bibliography

- [1] N. R. de Oliveira, P. S. Pisa, M. A. Lopez, D. S. V. de Medeiros, and D. M. F. Mattos, "Identifying fake news on social networks based on natural language processing: Trends and challenges," *Information*, vol. 12, no. 1, 2021. [Online]. Available: <https://www.mdpi.com/2078-2489/12/1/38>.
- [2] D. P. Acharjya and K. Ahmed, "A survey on big data analytics: Challenges, openresearch issues and tools," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, 2016. [Online]. Available: <https://www.studocu.com/row/document/universite-ibn-zohr/computer-science/9-a-survey-on-big-data-analytics-challenges-open-research-issues-and-tools-article-author-d-p-acharjya-kauser-ahmed-p/48804900>.
- [3] A. Malik and B. Tuckfield, *Applied Unsupervised Learning with R: Uncover Hidden Relationships and Patterns with K-Means Clustering, Hierarchical Clustering, and PCA*. Packt Publishing, Limited, 2019. [Online]. Available: [https://books.google.nl/books?hl=en&lr=&id=\\_jmPDwAAQBAJ&oi=fnd&pg=PR1&dq=Detecting+Hidden+Patterns+with+unsupervised+learning&ots=coC8qbkhaT&sig=g1YDmCdCesRUipThZniNMrF3bog#v=onepage&q=Detecting%20Hidden%20Patterns%20with%20unsupervised%20learning&f=false](https://books.google.nl/books?hl=en&lr=&id=_jmPDwAAQBAJ&oi=fnd&pg=PR1&dq=Detecting+Hidden+Patterns+with+unsupervised+learning&ots=coC8qbkhaT&sig=g1YDmCdCesRUipThZniNMrF3bog#v=onepage&q=Detecting%20Hidden%20Patterns%20with%20unsupervised%20learning&f=false).
- [4] P. Pappula and R. B. "An empirical comparison of clustering using hierarchical methods and k-means." (2016), [Online]. Available: [https://www.researchgate.net/publication/309305798\\_An\\_empirical\\_comparison\\_of\\_Clustering\\_using\\_hierarchical\\_methods\\_and\\_K-means](https://www.researchgate.net/publication/309305798_An_empirical_comparison_of_Clustering_using_hierarchical_methods_and_K-means).
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, 2003. [Online]. Available: <https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>.
- [6] J. Joung, K. Jung, S. Ko, and K. Kim, "Customer complaints analysis using text mining and outcome-driven innovation method for market-oriented product development," *Sustainability*, vol. 11, no. 1, 2018. [Online]. Available: <https://www.mdpi.com/2071-1050/11/1/40>.
- [7] D. Bholat, S. Hansen, P. Santos, and C. Schonhardt-Bailey. "Text mining for central banks," Bank of England. (2015), [Online]. Available: <https://www.bankofengland.co.uk/-/media/boe/files/ccbs/resources/text-mining-for-central-banks.pdf>.



- [8] V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers. "Clustering with deep learning: Taxonomy and new methods." (2018), [Online]. Available: <https://arxiv.org/abs/1801.07648>.
- [9] "Consumer complaint database." (2017), [Online]. Available: <https://www.consumerfinance.gov/data-research/consumer-complaints/>.
- [10] W. Mckinney. "Pandas: A foundational python library for data analysis and statistics." (2011), [Online]. Available: [https://www.researchgate.net/publication/265194455\\_pandas\\_a\\_Foundational\\_Python\\_Library\\_for\\_Data\\_Analysis\\_and\\_Statistics](https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics).
- [11] E. Rahm and H. Hai Do. "Data cleaning: Problems and current approaches." (2000), [Online]. Available: [https://www.researchgate.net/publication/220282831\\_Data\\_Cleaning\\_Problems\\_and\\_Current\\_Approaches](https://www.researchgate.net/publication/220282831_Data_Cleaning_Problems_and_Current_Approaches).
- [12] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, Incorporated, 2009.
- [13] M. J. Denny and A. Spirling, "Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it," *Political Analysis*, 2018. [Online]. Available: <https://www.cambridge.org/core/journals/political-analysis/article/abs/text-preprocessing-for-unsupervised-learning-why-it-matters-when-it-misleads-and-what-to-do-about-it/AA7D4DE0AA6AB208502515AE3EC6989E>.
- [14] C. Staunton, S. Slokenberga, and D. Mascalzoni, "The gdpr and the research exemption: Considerations on the necessary safeguards for research biobanks," *European Journal of Human Genetics*, vol. 27, pp. 1159–1167, 2019. [Online]. Available: <https://www.nature.com/articles/s41431-019-0386-5>.
- [15] B. Saha and D. Srivastava. "Data quality: The other face of big data." (2014), [Online]. Available: [https://www.researchgate.net/publication/271462251\\_Data\\_quality\\_The\\_other\\_face\\_of\\_Big\\_Data](https://www.researchgate.net/publication/271462251_Data_quality_The_other_face_of_Big_Data).
- [16] P. Gupta, A. Sharma, and R. Jindal, "Scalable machine-learning algorithms for big data analytics: A comprehensive review: Scalable machine-learning algorithms for big data analytics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2016. [Online]. Available: [https://www.researchgate.net/publication/308123199\\_Scalable\\_machine-learning\\_algorithms\\_for\\_big\\_data\\_analytics\\_a\\_comprehensive\\_review\\_Scalable\\_machine-learning\\_algorithms\\_for\\_big\\_data\\_analytics](https://www.researchgate.net/publication/308123199_Scalable_machine-learning_algorithms_for_big_data_analytics_a_comprehensive_review_Scalable_machine-learning_algorithms_for_big_data_analytics).
- [17] C.-C. Chen, H.-H. Juan, M.-Y. Tsai, and H. H.-S. Lu, "Unsupervised learning and pattern recognition of biological data structures with density functional theory and machine learning," *Scientific Reports*, vol. 8, 2018. [Online]. Available: <https://www.nature.com/articles/s41598-017-18931-5>.
- [18] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 2007.

- [19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Institute for Computer Science, University of Munich, Tech. Rep., 1996. [Online]. Available: <https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf>.
- [20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." (2018), [Online]. Available: <https://arxiv.org/abs/1810.04805>.
- [21] J. MacQueen. "Some methods for classification and analysis of multivariate observations." (1967), [Online]. Available: <https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Some-methods-for-classification-and-analysis-of-multivariate-observations/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp>.
- [22] M. Grootendorst. "BERTopic: Neural topic modeling with a class-based tf-idf procedure." (2022), [Online]. Available: <https://arxiv.org/abs/2203.05794>.
- [23] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, 1967. [Online]. Available: <https://link.springer.com/article/10.1007/BF02289588>.
- [24] M. Lan, S.-Y. Sung, H.-B. Low, and C. L. Tan. "A comparative study on term weighting schemes for text categorization." (2005), [Online]. Available: [https://www.researchgate.net/publication/4202318\\_A\\_comparative\\_study\\_on\\_term\\_weighting\\_schemes\\_for\\_text\\_categorization](https://www.researchgate.net/publication/4202318_A_comparative_study_on_term_weighting_schemes_for_text_categorization).
- [25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space." (2013), [Online]. Available: <https://arxiv.org/abs/1301.3781>.
- [26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality." (2013), [Online]. Available: <https://arxiv.org/abs/1310.4546>.
- [27] Q. V. Le and T. Mikolov. "Distributed representations of sentences and documents." (2014), [Online]. Available: <https://arxiv.org/abs/1405.4053>.
- [28] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. "Automatic evaluation of topic coherence." (2010), [Online]. Available: [https://www.researchgate.net/publication/220817098\\_Automatic\\_Evaluation\\_of\\_Topic\\_Coherence](https://www.researchgate.net/publication/220817098_Automatic_Evaluation_of_Topic_Coherence).
- [29] R. Baruri, A. Ghosh, S. Chanda, and R. Banerjee. "A comparative study on k-means clustering method and analysis." (2019), [Online]. Available: [https://www.researchgate.net/publication/333164868\\_A\\_Comparative\\_Study\\_on\\_k-means\\_Clustering\\_Method\\_and\\_Analysis](https://www.researchgate.net/publication/333164868_A_Comparative_Study_on_k-means_Clustering_Method_and_Analysis).

- [30] A. Bhardwaj. "Silhouette coefficient." (2023), [Online]. Available: <https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c>.
- [31] X. Wang and Y. Xu, "An improved index for clustering validation based on silhouette index and calinski-harabasz index," 2019. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/569/5/052024/meta>.
- [32] T. Caliński and J. A. Harabasz. "A dendrite method for cluster analysis." (1974), [Online]. Available: [https://www.researchgate.net/publication/233096619\\_A\\_Dendrite\\_Method\\_for\\_Cluster\\_Analysis](https://www.researchgate.net/publication/233096619_A_Dendrite_Method_for_Cluster_Analysis).
- [33] S. Nirmal, "Comparative study between k-means and k-medoids clustering algorithms," *International Research Journal of Engineering and Technology (IRJET)*, 2019. [Online]. Available: <https://www.irjet.net/archives/V6/i3/IRJET-V6I3154.pdf>.
- [34] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," 2010. [Online]. Available: [https://ieeexplore.ieee.org/abstract/document/5453745?casa\\_token=o5DM3QnCA7gAAAAA:qUrmls8pkpqtJoLUAX2dV3f46hV\\_DUo600DaY4055-UU8W0gSOYZ6hHtyL3J1zeMfOpCNH806rc](https://ieeexplore.ieee.org/abstract/document/5453745?casa_token=o5DM3QnCA7gAAAAA:qUrmls8pkpqtJoLUAX2dV3f46hV_DUo600DaY4055-UU8W0gSOYZ6hHtyL3J1zeMfOpCNH806rc).
- [35] K. Bastani, H. Namavari, and J. Shaffer, "Latent dirichlet allocation (lda) for topic modeling of the cfpb consumer complaints," *Expert Systems with Applications*, vol. 127, 2019. [Online]. Available: [https://www.sciencedirect.com/science/article/abs/pii/S095741741930154X?casa\\_token=4W1QvYdDAkoAAAAA:mDdb8WkWDVZ3WhTR3TQ7wXHOXXWe8FFW703f0aeTjCCBmQWQkR-g\\_ovKaNINmGOSH\\_tdwvIfegg0](https://www.sciencedirect.com/science/article/abs/pii/S095741741930154X?casa_token=4W1QvYdDAkoAAAAA:mDdb8WkWDVZ3WhTR3TQ7wXHOXXWe8FFW703f0aeTjCCBmQWQkR-g_ovKaNINmGOSH_tdwvIfegg0).
- [36] I. Putri and R. Kusumaningrum, "Latent dirichlet allocation (lda) for sentiment analysis toward tourism review in indonesia," 2017. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/801/1/012073>.
- [37] N. Reimers and I. Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." (2019), [Online]. Available: <https://arxiv.org/abs/1908.10084>.
- [38] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." (2018), [Online]. Available: <https://arxiv.org/abs/1802.03426>.
- [39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, 2008. [Online]. Available: <https://jmlr.csail.mit.edu/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>.
- [40] F. Murtagh and P. Contreras. "Methods of hierarchical clustering." (2011), [Online]. Available: <https://arxiv.org/abs/1105.0121>.
- [41] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," 2015. [Online]. Available: <https://dl.acm.org/doi/10.1145/2684822.2685324>.

- [42] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.
- [43] J. Oyelade, I. Isewon, F. Oladipupo, *et al.*, "Clustering algorithms: Their application to gene expression data," *Bioinform Biol Insights*, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5135122/>.
- [44] M. Meilă, "Comparing clusterings by the variation of information," in 1970. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-540-45167-9\\_14](https://link.springer.com/chapter/10.1007/978-3-540-45167-9_14).
- [45] A. Amoah-Mensah. "Customer satisfaction in the banking industry: A comparative study of ghana and spain." (2010), [Online]. Available: <https://core.ac.uk/download/pdf/132551562.pdf>.
- [46] M. Hardt, A. Narayanan, and S. Barocas, *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.
- [47] B. Institution. "Detecting and mitigating bias in natural language processing." (May 2021), [Online]. Available: <https://www.brookings.edu/articles/detecting-and-mitigating-bias-in-natural-language-processing/>.
- [48] H. Felzmann, E. Fosch-Villaronga, C. Lutz, and A. Tamò-Larrieux, "Towards transparency by design for artificial intelligence," *Science and Engineering Ethics*, 2020. [Online]. Available: <https://link.springer.com/article/10.1007/s11948-020-00276-4>.
- [49] B. I. for Innovation + Entrepreneurship. "Intro to ai for policymakers: Understanding the shift." (Mar. 2018), [Online]. Available: [https://brookfieldinstitute.ca/wp-content/uploads/AI\\_Intro-Policymakers\\_ONLINE.pdf](https://brookfieldinstitute.ca/wp-content/uploads/AI_Intro-Policymakers_ONLINE.pdf).
- [50] G. Fleming and P. C. Bruce, *Responsible Data Science*. Wiley, 2021.