GENWISE: Thematic Discovery from Textual Data

Minnie Kabra*
American Express AI Labs
minnie.kabra@aexp.com

Abhinav Nagpal*
American Express AI Labs
abhinav.nagpal1@aexp.com

Aayush Sacheti*†
Asato.ai
sachetiaayush@gmail.com

Mohit Kumar
American Express AI Labs
mohit.kumar30@aexp.com

Salil Rajeev Joshi
American Express AI Labs
salilrajeev.joshi@aexp.com

Abstract

In this work, we introduce GENWISE - a generative AI-based framework designed to streamline extracting and organizing key information from textual data. Focusing on the prevalent issue in business where significant time is spent on manual data analysis, our framework employs cutting-edge generative AI, embedding, and clustering techniques towards a thematic discovery. We further deliver hierarchical thematic representations, enhancing the ease of understanding for users at different levels. Our methodology includes precise issue extraction through generative AI, utilization of the Retrieval-Augmented Generation framework for improved accuracy, and a 20% improvement in cluster coherency using the Enhanced Community Detection algorithm. This comprehensive pipeline is optimized explicitly for industrial settings, offering a significant leap in efficiency and thematic representation for complex data sets.

1 Introduction

Banks and other financial institutions have been accumulating unstructured data for decades, including customer complaints, emails, chats, and call transcripts. Despite sophisticated processes to organize this data, about 80% of the analysis[1] remains descriptive. The sheer volume and complexity of this unstructured data pose significant challenges when combined with structured data, which is heavily relied upon by financial institutions to gain a comprehensive understanding of their customers. Business teams spend hundreds of hours each month reading and summarizing this data based on customer interactions to extract actionable themes. Identifying key information and grouping it semantically is both time-consuming and laborious. For example, sample records from a finance industry dataset mentioned in Table 1 contain several key segments, each as highlighted. This problem is compounded when these records are clustered based on these segments, represented at varying levels of detail.

In this paper, we address the challenge of thematic discovery from textual data using AI and ML techniques. We utilize cutting-edge generative AI, embedding techniques, and clustering methods to automatically identify key segments, transform them into a semantic format, and organize them hierarchically to ease the cognitive burden involved in the process.

The problem of thematic discovery has drawn academic attention in the past. Approaches ranging from simple rule-based extraction to statistical topic modelling and, more recently, neural techniques have been explored. However, we observed that these techniques had shortfalls in industrial settings. Bertopic's (Grootendorst, 2022) document representation is inadequate, or if it is adequate, like in TopicGPT (Pham et al., 2023), the algorithm is not suitable for industrial applications due to its high execution time in an online setting. Towards this end, we have curated an end-to-end system to discover themes from textual data in an unsupervised manner. The novelty of our system draws from the optimal use of underlying components and precise outcomes not feasible through existing systems.

Specifically, we make the following contributions:

- **Intuitive Representation:** The generated cluster representation presents the themes hierarchically at various levels of granularity, allowing senior leaders and business analysts to gain actionable themes.

- **Extracting important and distinct issues from**

*Equal contribution
†Work done as part of American Express AI Labs
[1] https://www.informationweek.com/machine-learning-ai/big-data-analytics-descriptive-vs-predictive-vs-prescriptive

Proceedings of the Joint Workshop of the 8th Financial Technology and Natural Language Processing (FinNLP) and the 1st Agent AI for Scenario Planning (AgentScen), Jeju, South Korea, August 3, 2024

Table 1: Sample CPFB Complaints, key information is highlighted

Abridged CPFB Note text
(ID 7317133) I filed a **dispute for incorrect information** on my credit report. I received an email from the credit bureau stating that they are assuming the disputes are coming from a 3rd party. They in fact did not come from a 3rd party. I even called them to verify it was me and they still **refused to process my disputes**. I wasted money on mailing my disputes out...
(ID 7317093) I am writing to **dispute the accuracy of the information** on my credit reports provided by XXXX, Experian, and XXXX. After reviewing the reports, I have identified several inconsistencies that I believe **require immediate attention and correction**. I kindly request that you **investigate and rectify the inaccuracies in accordance with the FCRA**...

the text: We use generative AI models with precise prompts to identify the key segments present in the text. The textual segments are further aggregated in a semantic space using state-of-the-art embedding techniques.

- **RAG (Retrieval-Augmented Generation) framework:** We use RAG to ingest industrial domain knowledge to reduce hallucination from generative models and make the segment extraction more precise.

- **Community detection algorithm enhancements:** After testing several non-parametric clustering approaches, we selected a community detection algorithm and further improved its cluster coherency by 20% on average.

The remainder of the paper is organized as follows: Section 2 presents a brief history of existing theme discovery systems focusing on recent advances. Section 3 provides a high-level architecture for GENWISE and explains the role of RAG and other generative AI paradigms. Section 4 provides a comparison of our system's performance on benchmarks as well as performance from state-of-the-art. Finally, we provide lessons learned in Section 5 that led to our ensemble architecture and note the conclusion in Section 6.

2 Related Works

Topic modelling is an information extraction technique that aims to extract a document's intrinsic "themes/topics" from a collection of documents. There have been multiple methods proposed over the years for topic modelling, including statistical methods (Deerwester et al., 1990; Hofmann, 1999; Blei et al., 2003; Févotte and Idier, 2011), deep learning-based methods using neural word embeddings (Moody, 2016; Dieng et al., 2020) and large language model (LLM) based approaches (Pham et al., 2023; Wang et al., 2023b) leveraging the zero-shot capabilities of the SOTA LLMs.

The earliest statistical approaches (Deerwester et al., 1990; Hofmann, 1999; Blei et al., 2003; Févotte and Idier, 2011) to model topics considered each document as a collection of words and modelled each document as a combination of underlying topics. LSI (Deerwester et al., 1990) decomposes a document term matrix using singular value decomposition (SVD) to identify the most prominent topics in each document. However, LSI has limited interpretability. Subsequently, Probabilistic latent semantic indexing (pLSI) (Hofmann, 1999) overcame this limitation by representing topics as multinomial random variables. Latent Dirichlet Allocation (LDA) (Blei et al., 2003), a hierarchical probabilistic model, generalized pLSI by incorporating Dirichlet conjugate priors for the word multinomial distributions over a topic and topic multinomial distributions over a document. However, these models have a limitation in that their document representation is inadequate as they do not consider the word context for creating document representation. They only consider the bag-of-words representations of the documents, ignoring the semantic relationships between words.

Of late, neural topic models (NTM) (Moody, 2016; Dieng et al., 2020) were proposed as computational power increased, and better text representation techniques such as (Mikolov et al., 2013) using neural models were discovered. One of the first proposed models was lda2vec (Moody, 2016) that employed word2vec (Mikolov et al., 2013) along with LDA. Lda2vec leverages the meaning encoded in a document to learn a better word representation by adding the document representation to the hub word representation to predict the context word representation. Here by hub, we refer to the key information or the cluster label. In the ETM model (Dieng et al., 2020), the topics contained in a document are represented in the same embedding space as the words. It produces a topic mixture from a logistic normal distribution and generates words for a topic by projecting the topic vector onto the vocabulary vector. ETM provides interpretable topics and achieves state-of-the-art results.

Recently, LLMs are increasingly used to analyze text automatically by prompting LLMs for tasks

Table 2: GENWISE compared to the closest works in the literature

Approach	GPTopic	TnT-LLM	TopicGPT	GENWISE
Input processing	Complete document	Queries an LLM to generate a summary of the document	Complete document	Use custom text cleansing modules including acronym expansion, spellchecker, etc., followed by querying an LLM to get the top-3 salient points in each document
Clustering	Uses HBDSCAN algorithm for clustering; Allows the user to specify a fixed number of topics; merges the topics using agglomerative clustering; Clustering occurs using either OpenAI / custom embeddings	Does not perform clustering explicitly	Does not perform clustering explicitly	Uses a hierarchical clustering approach generating themes at different granularities. Uses K-Means clustering in the 1st level followed by two levels of an Enhanced Community Detection. Clustering occurs using embeddings obtained from an LLM.
Topic generation	Names and description of themes are generated by prompting an LLM with the top-k words related to the theme	Prompts LLM multiple times in sequence to list the topics present in the document. Follows a topic generation, followed by a topic update and then topic review prompts.	Queries an LLM to list the topics present in the document given some sample topics	Use the hub element generated from clustering to label the topic/theme
Topic assignment	The document is assigned the label of the cluster to which it belongs	Uses a light-weight logistic regression model trained on the labels assigned by an LLM	Prompts an LLM to classify a given document to one or more topics generated during the topic generation phase	The document is assigned the labels of its constituent salient points

such as summarization (Liu and Healey, 2023; Laban et al., 2023), clustering (Hoyle et al., 2023; Zhang et al., 2023; Viswanathan et al., 2023), and topic modelling (Grootendorst, 2022; Pham et al., 2023; Wang et al., 2023b; Reuter et al., 2024; Wan et al., 2024). TopicGPT (Pham et al., 2023) aims to generate and label topics in an automated fashion using LLMs. It generates new topics by passing sample documents and some sample topics to an LLM. This resulting set is refined to avoid duplication. Another work, TnT-LLM (Wan et al., 2024), creates a label taxonomy using an LLM following topic generation, update and review steps. A lightweight classifier is then trained on the generated label taxonomy for classification. Similar to our work, (Reuter et al., 2024) first performs clustering on the documents using the HBDSCAN (Campello et al., 2013) algorithm, followed by labelling the clusters formed by prompting an LLM with the top 500 words related to each cluster. These words can come from different documents clustered together in the same cluster. Table 2 describes the approach of each of the above works. GENWISE offers a significant advancement over these previous works providing an efficient end-to-end pipeline. Instead of using multiple prompts throughout its pipeline, GENWISE streamlines the process by prompting only once, enabling quicker and more effective theme generation. Moreover, the hierarchical themes generated offers progressively finer themes providing a structured examination for users, starting from broader themes and moving to specific themes.

3 Solution Overview

Business analysts sift through large amounts of unstructured textual data to identify actionable themes. However, this data cannot be used in its raw form given that long texts, various ways of presenting the information, and domain information and jargon might be present as abbreviations. One needs to extract the key information to give it as an input to the clustering algorithm. We used generative large language models (LLM) to identify the key information of a raw text. In particular, we used open source LLMs such as Openchat[2] (Wang et al., 2023a) which is the best 7B parameter model at the time of writing this paper. As is widely known, a precise prompt is required for LLMs to extract the information suitably. We begin this section with details on prompt engineering for the financial text snippets. We subsequently explain the role of RAG, hierarchical representation, our enhancements to the clustering algorithm, and our high-level system architecture.

[2] https://huggingface.co/openchat/openchat-3.5-1210

3.1 Prompt engineering

We created the prompts using appropriate instruction placement, output format, multi-output responses, and negative instructions. Notably, the prompt engineering experiments were carried out with consideration for what different stakeholders expect the output to be.

1. **Instruction placement**: We noticed that providing the most important instructions at the beginning of the prompt was helpful to the LLM in carrying out the instruction. For example, '*This is a textual note from the customer. note_text For this note, carry out the following tasks. task_list*'.

2. **Output format**: This helps to parse and use the LLMs output easily. For example, '*For this note, provide the following information strictly in JSON format: output_format_example*'.

3. **Multi Output responses**: Asking for the multiple outputs in a single prompt to the LLM instead of using it multiple times to get output for a text at the different levels. For example, '*The JSON object should list key 'segments' which summarize the text. For each segment, provide a 'succinct description' and a 'concise label'*'.

4. **Negative Instructions**: LLMs tend to hallucinate without precise instructions. Negative instructions help reduce hallucinations by bounding the tasks. For example, '*Generate only the requested output, do not include any other language before or after the requested output. Do not repeat any information. Remove dates, amounts, and names.*'

3.2 RAG Framework

We address the challenge of LLMs misinterpreting acronyms by implementing the RAG framework, thus enhancing LLMs with internal knowledge for accurate acronym expansion. For example, "NPSL" may translate to "No present spending limit" for a financial company, while it expands to "National Premier Soccer League" as a general expansion. LLMs may hallucinate without accurately expanding the term *NPSL*.

We prevent this through a vector database (LlamaIndex[3]) that utilizes different indexing methods like VectorStore Index for semantic information and KeywordIndex for syntactic information, which is pivotal for acronym expansion.

We also used these indexes for ambiguous acronyms, i.e., which can be used as an acronym or word. "AM" is one such acronym. It can be either used as an acronym whose expansion is account manager (e.g., *am called to inform us*) or as a verb (e.g., *I am calling to ask*). We used these indexes to determine when such acronyms should be expanded. On an internal dataset, we found that when ambiguous acronyms are used, these indexes can identify them as acronyms 50% of the time. Moreover, when they are not used as an acronym, these indexes do not identify them as acronyms 80% of the time.

3.3 Levelwise clustering

Once the long descriptions corresponding to the key segments are extracted from the texts, they need to be clustered semantically. To present the semantic themes at various granularity levels, we create a hierarchy of clusters so that relationships between the clusters at different levels can be analyzed effectively.

Clustering techniques are applied at different levels to get granularity of themes, which can be crucial to understanding the data more deeply. Given that we cannot predetermine the number of themes in the data, we tried several non-parametric clustering algorithms such as Hierarchical Agglomerative clustering (HAC) (Hastie et al., 2009), mean-shift clustering (Derpanis, 2005), DBSCAN (Khan et al., 2014), etc., and observed best results with another algorithm fast community detection[4]. We further modified this solution and referred to it as "Enhanced Community Detection" (ECD). The overall hierarchical process works as follows:

- **Level 0 clusters** (L_0): Gets broad-level clusters with a primary objective of quickly partitioning large volumes of data. Extremely large clusters are not as informative as themes, and by suitably assuming a threshold for the maximum size of the cluster, we arrive at a broad estimate for the number of clusters. For this step, we use k-means clustering (Hartigan and Wong, 1979) as a fast, parametric clustering technique. This step is optional and depends on the volume of the data received.

[3] https://pypi.org/project/llama-index/

[4] https://github.com/UKPLab/sentence-transformers/blob/master/examples/applications/clustering/fast_clustering.py

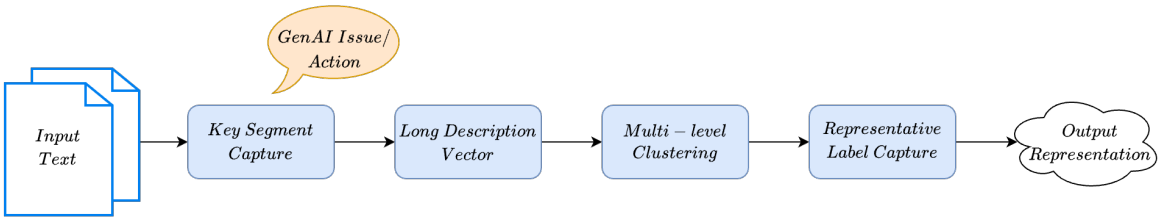

Figure 1: High-level system architecture for GENWISE

- **Level 1 clusters** (L_1): For all the k clusters obtained at the earlier step, ECD is applied to get level 1 clusters.
- **Level 2 clusters** (L_2): ECD is again applied with refined parameters to L_1 clusters to get further granularity (The parameters are refined using foundational algorithmic principles i.e., by reducing the cluster size threshold and increasing the similarity threshold). This meticulous process ensures that the clusters are more granular and specific, providing a deeper insight into the data.

Compared to HAC, the main advantages of community detection are its speed and ability to combine all the similar data points at once. On the other hand, HAC only combines two data points at a time, making it slower and prone to errors. Also, very similar points in HAC can result in totally different clusters depending on the initial configuration. This is not the case with ECD because many similar points are combined here in a single iteration rather than just 2 points. For instance, assume three similar data points: A, B, and C. The similarity of (A, B) is slightly greater than that of (A, C) and (B, C). In HAC, there can be a case where, after combining A and B, the similarity of (A, B) with C is insignificant. This is not the case with ECD, where all (A, B, C) will be combined simultaneously. Because of all these issues, ECD is much more suitable for obtaining coherent clusters.

3.4 Community detection algorithm Enhancements

The Enhanced Community Detection algorithm mentioned earlier clusters the data points based on cosine similarity between the points. In particular, it first identifies the community for the element, which includes all elements similar to that element. In many cases, communities can overlap, leading some elements to belong to multiple communities. However, we require each element to belong to a single community for the correct theme extraction. We must assign each element to only one community to form non-overlapping communities. Therefore, given an empty set of non-overlapping communities, we add an element to a new community only if the element is not a part of any existing community.

We observed a code flaw in sorting elements based on indices before forming non-overlapping communities, resulting in the loss of the position of the hub element (the one similar to all others). We identified and rectified it, increasing the cluster coherency by 20%. Further, we merged non-overlapping communities based on similarity, creating crisply defined clusters.

3.5 System Architecture

We now provide the end-to-end architecture for GENWISE. The block diagram is shown in Figure 1.

1. **GenAI Issue/Action Capture**: This stage is critical for the initial processing of input text. The generative AI model scans the provided text, identifying and extracting key information segments. Our custom prompt and RAG are enabled at this stage to capture the information precisely. As seen from the first block in Figure 1, these key segments are usually issues or actions relevant to the text's subject matter. For each extracted segment, we prompt the model to list a *succinct description* and a *concise label*.

2. **Vector Conversion**: Post extraction, each segment is converted into a vector form using its succinct description. For this conversion, we use state-of-the-art models as per the Massive Text Embedding Benchmark (MTEB)[5].

3. **Multi-Level Clustering**: In this phase, the vectorized data undergoes hierarchical clustering. This method groups the vectors into clusters

[5]The Massive Text Embedding Benchmark (MTEB) is an extensive benchmark developed to assess the performance of text embedding models on many tasks and datasets.https://huggingface.co/spaces/mteb/leaderboard

Table 3: Overall Quantitative comparison of BERTopic and GENWISE-predicted labels

Data Source	documents	Label Similarity		Increase
		BERTopic	GENWISE	
CFPB	2000	65%	69%	6.2%
Bills	1000	59%	62%	5.1%

based on their similarities. The multi-level aspect of this clustering allows the system to organize the data at various levels of granularity, facilitating a more nuanced understanding and categorization of the themes within the text.

4. **Representative Label Extraction**: The final step of the pipeline is the extraction of representative labels for each cluster. This process involves identifying the data point that accurately encapsulates each cluster's core theme or idea. We found that the hub element serves as a good representative for the cluster as it acts as the central element for forming a community. We choose the concise label of the hub element to label the cluster. This process is repeated for each level of the constructed hierarchy.

4 Experiments

This section provides three findings:-

- Dataset - A brief introduction to the datasets used for experimentation.
- Label comparison – We compared BERTopic and GENWISE-predicted labels quantitatively and qualitatively.
- Quantitative evaluation of Enhanced Community Detection algorithm and time complexity of related algorithms
- Industry data-based study on Fixed Term Effort (FTE) reduction with GENWISE

4.1 Dataset

For experiments, we considered two datasets from different domains.

CFPB[6] is a consumer complaints database by the Consumer Financial Protection Bureau (CFPB), which contains two kinds of labels for a complaint – Issue and Sub-issue (the issue & sub-issue mentioned by the consumer in the complaint). We have taken a sample of 2000 complaints from CFPB for our experiment.

Bills is a generic dataset summarising the bills discussed in the U.S. Congress Bills (Adler and Wilkerson, 2018). This dataset has 21 high-level and 114 low-level human annotated labels. A sample of 1000 summaries has been considered for our experiment.

4.2 Label comparison

The ground truth (provided in the annotated datasets) and predicted labels (from LDA, BERTopic) are compared qualitatively and quantitatively with the labels generated by GENWISE. The results are mentioned in Table 3 and Table 4. In contrast to current clustering algorithms that produce a single label after processing, GENWISE automatically generates a hierarchy of labels, providing a more nuanced understanding of the data. Furthermore, GENWISE generates more informative labels than those produced by either BERTopic, LDA or the annotated labels. We evaluated it quantitatively by comparing the semantic similarity of the annotated label and the complaint. This process involved a manual comparison of a subset of the labels (presented in Table 4). Bge-large-en-v1.5[7] has been used to compute the embedding of labels and complaints, and dot product has been used to compute the similarity. For both datasets, on average, the semantic similarity of GENWISE-predicted labels is 5% higher than that of BERTopic labels (mentioned in Table 3). For the examples provided in Table 4, Labels derived from GENWISE predictions have a higher semantic similarity than BERTopic.

4.3 Quantitative comparison

We compared the performance of old and Enhanced Community Detection algorithms using commonly used metrics, such as topic coherence and topic diversity. Both measures are based on the hub element (label) of community detection, as it represents that cluster. They are calculated for the last level of the hierarchy. For a cluster, topic coherence is implemented as normalized pointwise mutual information for n points closest to the hub element (n is taken as 3 for our study) (Bouma, 2009). Its value ranges from (-1, 1), where higher values show a more significant intracluster correlation. Topic

[6] https://www.consumerfinance.gov/data-research/consumer-complaints/

[7] https://huggingface.co/BAAI/bge-large-en-v1.5

Table 4: Qualitative & Quantitative comparison of BERTopic and GENWISE-predicted labels

			BERTopic		GENWISE	
Data Source	**Complaint/Bill Summary**	**Issue -> Sub-issue**	**Label**	**Similarity**	**Label**	**Similarity**
CFPB	(ID 7284263) I submitted a letter to the Credit Bureaus to correct these erroneous accounts. I think you have not validated these accounts in accordance with Sections 609, and I will pursue legal action against them.	Problem with a credit reporting company's investigation into an existing problem -> Their investigation did not fix an error on your report	34_include_pursue_suspicious_prior	56%	Unauthorized credit report ->Request for removal of erroneous items under Fair Credit Reporting Act	77%
CFPB	(ID 7317133) I filed a dispute for incorrect information on my credit report.I received an email from the credit bureau stating that they are assuming the disputes are coming from a 3rd paty.They in fact did not come from a 3rd party.I even called them to verify it was me and they still refused to process my disputes.	Problem with a credit reporting company's investigation into an existing problem -> Their investigation did not fix an error on your report	32_dispute_verify_incorrect_information	69%	Unauthorized credit report -> Dispute Not Processed by Credit Bureau	82%
Bills	(112-S-3595) Amends the Internal Revenue Code to exempt from passive loss rules any activity of a taxpayer carried on by a high technology research small business pass-thru entity. Defines "high technology research small business pass-thru entity"	Domestic Commerce -> Small Businesses	1_tax_credit_revenue_internal	59%	Exempting High Technology Research Small Businesses -> Exemption from Passive Loss Rules for High Technology Research Small Business	87%
Bills	(110-HR-614) Amends titles XI and XIX (Medicaid) of the Social Security Act (SSA) to remove the cap on Medicaid payments for Puerto Rico, the Virgin Islands, Guam, the Northern Mariana Islands, and American Samoa.	Public Lands ->Dependencies & Territories	2_health_medicare_service_care	52%	Removing Cap on Medicaid Payments ->Amendment to Social Security Act	73%

Table 5: Cohesion for Old & Enhanced community detection

Similarity threshold	**Cohesion (Old)**	**Cohesion (Enhanced)**	**Increase**
60%	58%	76%	30%
65%	63%	74%	16%
70%	68%	76%	11%

diversity measures the intercluster correlation and is calculated by computing the pairwise similarity between the most representative members of every cluster. A larger diversity score indicates clusters that are distinct with the least overlap.

On the sample of the CFPB dataset mentioned above, we calculated cluster cohesion and diversion across a range of similarity thresholds for both old and Enhanced Community Detection algorithms. On average, cohesion increased by 20% for all such experiments. Topic diversity was similar across both old and Enhanced Community Detection algorithms.

Table 5 shows the coherence across different similarity thresholds.

End-to-end time comparison: We also noted the time taken to perform different components of the end-to-end pipeline in Table 6. Overall, BERTopic took the least time to run the complete pipeline. Since GENWISE uses a large language model to generate the descriptions from the document, it takes much more time than BERTopic end-to-end. Another thing to note is the time taken in an online setting. GENWISE takes the same time as BERTopic in an online setting, as the first two pipeline stages for both BERTopic and GENWISE are pre-computed. However, each text snippet has multiple levels of granularity due to the themes provided by GENWISE at different granularity levels, which provides more information on the documents than BERTopic. Compared to TopicGPT, another LLM-based pipeline involving prompting, GENWISE is 50% faster as TopicGPT uses multiple prompts to run the complete pipeline.

Table 6: Time taken (in minutes) to generate themes on CFPB end-to-end using BERTopic, TopicGPT and GENWISE.

Time Taken (in min)	**BERTopic**	**TopicGPT**	**GENWISE**
Input Processing	0	0	5
Embedding + Clustering	0.7	0	2
Topic Generation	0.1	5	0.1
Topic Assignment	0.1	10.1	0.1
Total	0.9	15.1	7.2

4.4 Industry data-based study

Lastly, we report the experimental investigation on industry data, which is only a small part of complete unstructured data. The task involves analysing and extracting frequently appearing themes among customer text complaints received through the customer support helpline or email. The customer complaint dataset comprises a diverse set of complaints. The dataset comprises

Table 7: Data distribution

Time Period	Records	Customers
Aug-2022	132K	110K

Table 8: Results on industry based dataset

Records	Existing process	GENWISE
Total records (Aug 2022)	132K	132K
Complaints from high-risk category (Based on Complaint categorization)	12K	12K
Records for manual review	942	50
Actionable complaints (Opportunities)[9]	8	8
Issue hit rate[10]	<1%	16%

approximately 190 attributes[8] for each complaint, which describes a customer's spending history and other customer-specific information. One of these 190 fields is a complaint field in textual format. This complaint field describes major issues faced by the customer. This dataset is crucial for financial control and ensuring compliance with regulations. It helps promptly address high-risk complaints and issues to prevent potential legal or financial risks. In such cases, taking necessary actions as quickly as possible is essential. Table 7 provides details on the data distribution for the dataset.

In our research, we investigated the effectiveness of using GENWISE in reducing the amount of Fixed Term Effort (FTE) required in the existing complaint categorization process. FTE refers to the predetermined manual work or resources assigned to a specific task for a set duration. The existing process involves Customer Care Professionals (CCPs) manually filtering and reviewing complaints, which can be time-consuming and inefficient. GENWISE is a tool designed to automate this process and provide direct guidance to CCPs in identifying critical complaints, thus reducing the need for manual efforts.

As we discovered, the implementation of GENWISE led to a significant decrease of 95% in the FTE required for the existing process. Of 942 complaints in the industry-based dataset, 95% (892 complaints) were categorized as low-risk, enabling CCPs to concentrate only on the 50 high-risk-themed complaints that required manual review for actionable items. For a clear presentation of the outcomes obtained from the industry-based dataset, please refer to Table 8.

Additionally, the implementation of GENWISE significantly increased the issue hit rate. Previously, the rate was less than 1% , with only eight instances identified out of 942 complaints. However, with GENWISE, the rate dramatically rose to 16%, with 8 cases identified from a smaller sample size of 50 complaints.

5 Observations and Lessons Learnt

This paper outlines our method for extracting themes from unorganized and unlabeled textual data using specific knowledge in the field. Our approach is particularly effective in quickly identifying main themes in extensive data collections. Additionally, by using Enhanced Community Detection, we attained more connected and refined outcomes. The flexibility of this inclusive process enables smooth application on various datasets, with minimal adjustments and parameter tuning needed.

1. **Hierarchical clustering representation suits broad user community**: We started with regular clustering algorithms, which gave us a single label. Moreover, these labels usually fall in the medium range of granularity, i.e., they convey the subject of the cluster but not exactly what the cluster is about. Through ongoing engagement with our stakeholders, we discerned that they require labels at multiple levels - a broader label and then a label that tells precisely about the cluster. Depending upon the use case and the team utilizing the clustering output, multiple granularity levels would be required in the clustering. Thus, we designed a hierarchical clustering pipeline with three levels of clustering. The granularity of the clustering increased with each level. This helped us create a product aligned with the business requirements.

2. **Streamlined computation for real-time analysis**: Initially, we tried to run the entire approach in real-time, which was slow and sub-optimal. Maximizing computational tasks through batch processes is crucial to optimize the overall pipeline's latency, minimizing the load during inference. This objective was accom-

[8]Due to privacy reasons, this dataset cannot be released.

[9]Throughout this paper, 'actionable complaints' (opportunities) refers to high-risk complaints that warrant quick action.

[10]The term "issue hit rate" refers to the percentage derived from the ratio of actionable complaints to the total number of complaints manually reviewed.

plished by conducting various steps, such as pre-processing, RAG, LLM issue detection, embedding generation, and clustering labelling as offline batch processes. During inference, the focus was narrowed to efficient clustering analysis through Enhanced Community Detection, ensuring streamlined pipeline performance.

3. **Importance of appropriately labelling a cluster**: Following identifying a cluster, it becomes imperative to aptly label it, allowing users to grasp its essence succinctly. As highlighted in the prompt engineering section, we engaged various stakeholders to achieve this, tapping into their domain expertise. This collaboration proved invaluable in prompt engineering and the RAG-mentioned steps above. These methods underwent meticulous refinement, yielding verbose and succinct labels tailored to the specific requirements articulated in the prompt. Notably, we emphasized phrases near the cluster's centroid to discern the cluster's optimal semantic essence as mentioned in the Topic Generation step.

4. **Evaluation of the pipeline**: The critical challenge in our process stems from the subjective and domain-specific nature of generating content using Large Language Models (LLMs). To overcome this challenge, we carefully examined specific data segments, sometimes using a list of keywords related to particular customer issues. We used the ground truths obtained from these segments as benchmarks to assess the performance of our pipeline. By comparing the ground truths with the results generated by our pipeline, we created a confusion matrix for analysis. Additionally, incorporating some random data mixed with data containing a known set of themes allowed us to discover valuable insights in specific areas where improvements to the model were needed. This experiment was carried out with different random mixes and known theme distributions to check if the model could identify the themes independently.

6 Conclusion

Motivated by an industrial setting - going beyond accuracy and looking for trust and interactivity - we built and presented an end-to-end system, GENWISE, to derive themes from the text. Our system makes descriptive analytics and reporting much easier and more natural for users. We made it trustworthy through features such as a clustering hierarchy and appropriate labels. During this journey, we encountered several gaps in academic solutions for the clustering. We presented lessons learnt while overcoming these challenges and supporting demands from business stakeholders.

References

E Scott Adler and John Wilkerson. 2018. Congressional bills project: 1995-2018.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Konstantinos G Derpanis. 2005. Mean shift clustering. *Lecture Notes*, 32:1–4.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Cédric Févotte and Jérôme Idier. 2011. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9):2421–2456.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Alexander Hoyle, Rupak Sarkar, Pranav Goel, and Philip Resnik. 2023. Natural language decompositions of implicit content enable better text representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13188–13214, Singapore. Association for Computational Linguistics.

Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. 2014. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE.

Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676, Singapore. Association for Computational Linguistics.

Sengjie Liu and Christopher G Healey. 2023. Abstractive summarization of large document collections using gpt. *arXiv preprint arXiv:2310.05690*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Christopher E Moody. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. 2023. Topicgpt: A prompt-based topic modeling framework. *arXiv preprint arXiv:2311.01449*.

Arik Reuter, Anton Thielmann, Christoph Weisser, Sebastian Fischer, and Benjamin Säfken. 2024. Gptopic: Dynamic and interactive topic representations. *arXiv preprint arXiv:2403.03628*.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2023. Large language models enable few-shot clustering. *arXiv preprint arXiv:2307.00524*.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, et al. 2024. Tnt-llm: Text mining at scale with large language models. *arXiv preprint arXiv:2403.12173*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023b. Goal-driven explainable clustering via language descriptions. *arXiv preprint arXiv:2305.13749*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. Clusterllm: Large language models as a guide for text clustering. *arXiv preprint arXiv:2305.14871*.