Aalto University
MS-E2177 - Seminar on Case Studies in Operations Research

Semantic risk clustering

Final Report

Santeri Paljakka (Project Manager)

Essi Nikula

Henrik Purokoski

Jaakko Paavilainen

Antti Kärkkäinen

May 31, 2026

Contents

1 Introduction

Inclus is a growth company specialized in participatory, visual, and interactive cloud software designed for collaborative and complex risk management. Inclus helps its clients to understand their risk landscape by mapping different risk scenarios and providing related software tools for risk analysis and management. One of the used tool in the process is risk registers. Risk registers are databases that maintain a comprehensive and current list of possible risk scenarios the customer has identified. The risk registers contain data entries, such as risk labels and textual descriptions of the risks.

In qualitative risk management, risks are primarily documented in written form. The unstructured textual data creates challenges for data analysis with traditional measures. Additionally, contribution by multiple stakeholders easily leads to several inconsistencies in the data, such as redundancy, and varying formats. When conducted manually, perceiving the whole risk landscape and understanding closely connected risks becomes infeasible as the risk register can have thousands of entries. Furthermore, traditional lexical clustering or keyword matching often fails to capture the informal information and linguistic themes within textual data (Kuhn et al., 2007). Grouping similar risks, identifying duplicates and creating meaningful visualizations would have real value for the usability of risk register data.

The recent development of Large Language Models (LLM) and Natural Language Processing (NLP) has enabled new possibilities for processing textual input and qualitative data. Methods such as Information Retrieval (IR), transformer-based technologies and, most recently, LLM embeddings can be used to process textual input to vectors, enabling comparison with different vector similarity measures. Clustering based on the semantic similarities of textual documents has been proposed as a part of topic modeling pipelines, for example by Grootendorst (2022) and Mersha et al. (2024). However, these advanced NLP methods are not yet studied in detail using actual risk data and risk registers.

As presented in a recent thesis work by Westergård (2025), Inclus has been utilizing these NLP techniques with risk register data by developing LLM capabilities, such as AI agent using Retrieval-Augmented Generation (RAG). Additionally, an initial trial of semantic risk clustering pipeline has proven potential in finding structure in qualitative risk data. The initial trial shows that semantic clustering offers a way to tackle the current issues with large and unstructured risk registers while providing new kinds of insights about the risk landscape. For the development of this new tool, comprehensive background study and rigorous testing is required to design a tool that employs best practices in the field

1.1 Objectives

The objective for this project is designing a semantic risk clustering pipeline for risk descriptions in risk registers, while comparing the performance of different technologies and design choices. At the end, we present results, limitations and capabilities, and recommendations for building an usable and reliable semantic risk clustering tool.

The process is composed of two main components: Building an extensive literature review to find industry best practices, and a practical testing using provided risk data set, selected embedding models and clustering algorithms. Our study focuses on the comparison of the technologies and identification of optimal design choices rather than software development or coding.

2 Literature Review

The literature review gives an overview of the semantic clustering pipeline and all the related methods based on references to the current literature. Its findings are used to scope down the testing phase of this project.

Subsection 2.1 provides background on risk data, followed by an overview on semantic clustering in Subsection 2.2 and preprocessing practices in Subsection 2.3. Subsections 2.4, 2.5, 2.6 and 2.7 then identify the most used methods and best practices related to embeddings, similarity, dimensionality reduction and clustering algorithms.

2.1 Risk Registers

Successfully managing risk is a critical part of ensuring any project, from building software to constructing a bridge, is completed successfully. According to Aven (2012), there is no one simple definition of a risk and the concept has evolved to encompass potential events, their consequences, and the associated uncertainties. To systematically track and mitigate these uncertainties, organizations rely on risk registers, which are foundational databases for logging qualitative risk data, textual descriptions, and mitigation strategies (Leva et al., 2017). An example of public risk register is the MIT AI Risk Repository that lists and categorizes risks related to the development of AI (Slattery et al., 2026). While risk management is a well-studied academic field, there is a lack of case studies where actual risk registers are utilized for advanced, modern NLP analysis, such as RAG or semantic clustering.

The broader application of AI, NLP, and LLMs to unstructured risk text has seen rapid development. Dolphin et al. (2026) managed to successfully utilize LLMs and semantic embeddings to extract structured risk factors from 10-K corporate financial filings and map them to predefined taxonomies. Beyond finance, transformer models such as BERT and GPT-4 are used also in the construction industry to automate risk classification from news and contract data, and generative AI has been applied to identify phase-specific project risks directly from complex technical documentation (Erfani and Khanjar, 2025) (Zou et al., 2017). While these studies clearly demon-

strate the powerful capabilities of LLMs in processing unstructured textual risk data, the specific application of semantic clustering to organize and analyze the raw risk descriptions remains an unexplored use case.

2.2 Semantic clustering

Clustering is a fundamental unsupervised learning method used to divide unlabeled data into clusters of shared characteristics. The aim is to achieve high similarity within clusters and dissimilarity between different clusters. In a linguistic context, Kuhn et al. (2007) introduce semantic clustering to capture the themes and skopos underlying a text. By treating texts as collections of interrelated concepts rather than mere word occurrences, this approach resolves issues of synonymy and polysemy. In the context of risk management, this helps classify information based on original human intent rather than simple keyword matching.

As Petukhova et al. (2025) emphasizes, clustering is an iterative information retrieval process requiring continuous refinement to align data representations with the analyst's intent. Since the algorithmic suitability depends heavily on the dataset structure and the intended application Zhang et al. (2020), preprocessing and model parameter tuning are critical to the meaningfulness of the output (Mersha et al., 2024). This necessitates multiple experimental cycles to achieve desired characteristics.

2.2.1 Modular pipeline

Modern semantic clustering has evolved into sophisticated Neural Topic Models (NTMs), most notably the BERTopic framework. As presented by Grootendorst (2022), this approach utilizes a modular pipeline that transforms documents into high-dimensional embeddings – a process that Yang and Kim (2025) demonstrate is most effective when using Large Language Model (LLM) embeddings. To manage this high-dimensional space, the pipeline typically integrates algorithms such as UMAP for dimensionality reduction and HDBSCAN for density-based clustering.

The sequential modular process is illustrated in Figure 1. Capturing the conceptual depth effectively requires integrating multiple, complementary perspectives of text semantics. As Zhang et al. (2020) argues, while traditional topic models are adept at identifying "global" word patterns across a dataset, modern word embeddings are superior at capturing "local" context-aware relationships. By utilizing transformer-based architectures and LLM embeddings, frameworks such as the BERTopic, are able to generate multidimensional vectors that bridge these two perspectives. As shown by Mersha et al. (2024) and Canli (2025), this integration is able to capture of deep, context-dependent meanings, leading to more coherent topic segmentation and providing a solid foundation for professional risk analysis.

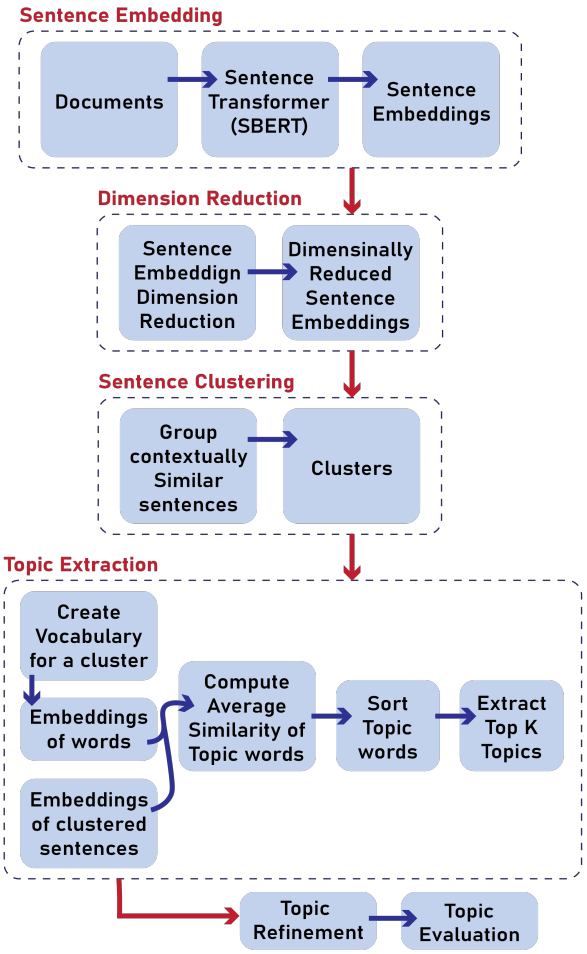

Figure 1: The semantic clustering pipeline based on literature. Adapted from Mersha et al. (2024).

2.3 Data preprocessing

Data preprocessing converts raw text into a formats suitable for computational analysis. While traditional bag-of-words (BoW) models treat words as discrete entities disregard semantic context, modern frameworks like BERTopic leverage Transformer-based Large Language Models (LLMs) to preserve deep linguistic relationships (Grootendorst, 2022).

The shift toward contextual embeddings has moved the analytical focus from linguistic cleaning toward semantic consistency. Because these models can extract meaning directly from raw text, there is a growing claim that heavy manual cleaning may no longer be strictly necessary. Indeed, Grootendorst (2022) demonstrates that BERTopic remains competitive with minimal preprocessing. However, Zhang et al. (2024) argue that structured cleaning is still necessary in noisy contexts, suggesting that benefits vary depending on the model. To address this trade-off, three distinct preprocessing levels can be evaluated as summarized in Table 1.

Table 1: Preprocessing Strategies

Approach	Mechanism	Suitability in project	Primary source
Linguistic cleaning	**Subtractive:** Removing noise, stop-words, lemmatization	**Baseline:** Standardizes grammar and handles domain-specific noise and inflections	Yang and Kim (2025)
Semantic canonicalization	**Generative:** LLM-rewriting into standardized registers before embedding	**Solution:** Normalizes human stylistic variance while preserving nuance	Zhang and Soh (2024)
Feature amplification	**Structural:** Restructuring text into Subject-Predicate-Object	**Enhancement:** Captures expert intent rather than statistical patterns	Viswanathan et al. (2023)

2.3.1 Structural design

While LLM-driven normalization can reduce redundancy, it must be applied with caution to avoid losing semantic depth. Petukhova et al. (2025) notes that advanced techniques, such as text summarization, do not always improve clustering performance and may remove subtle linguistic cues. To mitigate this, preprocessing can be treated as an iterative process that extends into the clustering phase to filter vocabularies that lack semantic meaning (Mersha et al., 2024).

In addition to content-based cleaning, structural design of the input is also relevant for semi-structured data, such as risk registers. Liao et al. (2023) argue that using document templates to maintain relationships between anchor fields (e.g., a risk title) and their descriptions is more robust than simple sequential strings. This is because the order and structure of fields determine the orientation of the embedding in the vector space, necessitating a strategic trade-off between computational cost and the preservation of thematic details.

2.4 LLM Embeddings

Text embeddings are numerical representations of natural language, created by mapping text into a continuous, high-dimensional vector space. In this resulting space, semantic similarity between texts corresponds to their spatial proximity.

2.4.1 Encoder vs. decoder architectures

The foundation for semantic embeddings was laid by models using shallow neural networks such as Word2Vec (Mikolov et al., 2013). The transition from these word-level

models to transformer-based architectures solved limitations of context-free embeddings. These encoder architectures such as BERT (Devlin et al., 2019) and Sentence-BERT (Reimers and Gurevych, 2019) utilize bidirectional attention to simultaneously integrate context from the entire input sequence across discrete sub-word tokens. This parallel processing, combined with relatively low parameter counts and token pooling, allows for the computationally efficient generation of embeddings at scale.

Recently, decoder-only LLMs have been shown capable of beating encoder-based models in generating embeddings (Muennighoff, 2022; Wang et al., 2024c). Unlike encoders, decoder architectures utilize causal attention where tokens only see preceding context. To represent the entire sequence, these models extract the hidden state of an end-of-sequence (EOS) token that aggregates the preceding context into the final embedding vector. The advantage of LLM-based embedders lies in the detailed context learned during large-scale training, which can capture complex nuances. However, their significantly higher parameter counts make decoder models more computationally expensive than encoder-based alternatives (Wang et al., 2024b). The technical distinctions between these approaches are summarized in Table 2.

Table 2: Comparison of Embedding Architectures

Architecture	**Attention**	**Extraction method**	**Primary advantage**
Encoder (SBERT)	Bidirectional	Mean/CLS Pooling	High computational efficiency
Decoder (LLM)	Causal	EOS Token Hidden State	Superior semantic nuance

State-of-the-art models primarily rely on contrastive learning, where models are trained to bring semantically related text pairs together in the embedding space while pushing unrelated pairs apart (Wang et al., 2024a). During training, the model is fed pairs of text consisting of query and passage. The training objective is to maximize their similarity and minimize similarity of query against multiple negative samples.

2.4.2 Instruction tuning and MTEB evaluation

A single embedding space is rarely optimal across all downstream tasks. To address this, Su et al. (2023) introduce instruction tuning which allows single embedding models to adapt their representations based on natural language task description, such as instructing the model to represent a text for retrieval or clustering. This eliminates the need for task-specific fine-tuning or maintaining separate models.

The current primary benchmark, the Massive Text Embedding Benchmark (MTEB), introduced by Muennighoff et al. (2023), provides a systematic framework for evaluating and comparing embedding models across diverse NLP tasks and domains. For risk analysis, MTEB's clustering benchmark is the most relevant metric. Current top performing families in this category, such as F2LLM (Zhang et al., 2025) and KaLM

(Hu et al., 2025), demonstrate the strong performance lead of LLM-based embedders over traditional encoder models.

2.5 Vector databases and similarity

Traditional databases are often inadequate for high-dimensional vectorized data, such as text embeddings. To combat this issue, modern artificial intelligence systems usually use specialized vector databases (VDB) (Ma et al., 2025) that provide efficient storage and search techniques. The main advantage of VDBs is their ability to efficiently find vectors similar to the given query vector, also known as similarity search. The two main groups of methods for similarity search are the nearest-neighbor search (NNS) and approximate nearest-neighbor search (ANNS). While NNS finds the exact nearest neighbor, ANNS allows for some error in the result, balancing search accuracy and computational load (Ma et al., 2025).

2.5.1 Similarity metrics

Efficient similarity search is essential for many downstream AI tasks ranging from semantic clustering —which relies on precise distance measurements (Murtagh and Contreras, 2012)— to retrieval-augmented generation (RAG), introduced in Lewis et al. (2020). RAGs enhance generative AI models by using documents in an external VDB. They usually use similarity search to find the documents that are best suited for the task given.

To performe similarity search, various metrics can be used. Some of the most common similarity metrics belong to the Minkowski distance family, which is defined by

$$dist(d_a, d_b) = \left(\sum_{k=1}^{n} |d_{a,k} - d_{b,k}|^p \right)^{\frac{1}{p}}$$

, where d_a and d_b are the two vectors compared. The two most common special cases of this are Manhattan distance ($p = 1$), and Euclidian distance ($p = 2$) (Murtagh and Contreras, 2012).

A commonly used metric in semantic clustering is the cosine similarity

$$sim(d_a, d_b) = \frac{d_a \cdot d_b}{||d_a|| \cdot ||d_b||}$$

, where $d_a \cdot d_b$ is the inner product and $||d_a|| \cdot ||d_b||$ is the product of their lengths. Unlike distance-based metrics, cosine similarity measures the angle between the vectors. This minimizes the effect of document length, which is useful in clustering applications, where the semantic meaning is of interest (Mehta et al., 2020). This is also applicable in the risk context, since risk descriptions may vary in length.

In addition to these commonly used methods, some studies introduce specialized alternatives to better capture semantic nuances. For instance, Sidorov et al. (2014)

introduce a soft cosine similarity metric that takes into account word similarity rather than assuming independence, such as traditional cosine. Zhelezniak et al. (2019) suggest that using non-parametric rank correlation coefficients can increase performance in cases where cosine similarity is unfit.

2.6 Dimensionality reduction

Despite efficient similarity search methods offered by vector databases, clustering embeddings with very high dimensions can be computationally inefficient. Additionally, this kind of data is difficult to interpret and visualize. Thus, the embeddings are usually dimensionally reduced before clustering (Aggarwal et al., 2001; Van der Maaten and Hinton, 2008). To achieve this, multiple different methods can be used.

The most commonly used method for dimensional reduction is Principal Component Analysis (PCA), which uses eigenvectors and eigenvalues to transform a dataset into a new coordinate system. This process identifies the directions of maximum variance, enabling the reduction of noise and redundancy by prioritizing principal components that capture the most significant structural information (Jolliffe, 2025).

In addition to PCA, some other techniques can be used. Van der Maaten and Hinton (2008) introduces t-distributed Stochastic Neighbor Embedding (t-SNE), a method that uses a probabilistic approach to map high-dimensional data into a two– or three–dimensional space by preserving local structures. McInnes et al. (2018) present Uniform Manifold Approximation (UMAP), which leverages Riemannian geometry to preserve both local and global data structures at a much higher computational speed. This provides a more scalable alternative to t-SNE that maintains the "big picture" of high-dimensional embeddings. These methods are particularly useful for visualizing and interpreting the data and clusters.

2.7 Clustering algorithms

K-means is a partitioning algorithm widely used as a baseline in clustering applications due to its efficiency. Presented by McQueen (1967), it assigns each data point to one of the k clusters by minimizing the within-cluster sum of squares. Each cluster is represented by the mean of its points. It is well suited for large problems but suffers from poor selection of k.

Different hierarchical clustering methods are popular largely due to the interpretability of the results. Agglomerative hierarchical clustering starts with each datapoint as their own cluster and merges them together one by one with specified rules. Ward Jr (1963) proposed a method, later known as Ward linkage, to merge the two clusters with minimum sum of squared errors, a similar approach that is used in k-means.

Density based clustering methods are introduced later with the benefit of no assumptions on the shape of the underlying data. Campello et al. (2015) introduce HDBSCAN, which extends the traditional DBSCAN by converting it into a hierarchical approach.

Furthermore, a variety of other clustering methods have been proposed to cluster textual documents. A few examples are spectral clustering (Ng et al., 2001), fuzzy c-means (Bezdek, 2013), deep autoencoders (Berahmand et al., 2022) and ensemble clustering (Strehl and Ghosh, 2002).

2.7.1 Evaluation and validation

Evaluation and validation of the final clustering remains a fundamental challenges in unsupervised machine learning. Usually, several metrics are used in the evaluation process. Two common internal validation metrics include the silhouette score and Davies-Bouldin Index. The silhouette score measures how similar a data point is to its own cluster compared to other clusters. A higher score indicating more distinct clusters. Davies-Bouldin Index calculates the average similarity between each cluster and its most similar neighboring cluster. Ranging from 0 to 1, lower average indicates a lower overlap and thus better, separate clustering.

The internal validation metrics rely, however, only on the feature space and the chosen distance measure, and will often fail to capture the semantic meanings in the data. To tackle this, in topic modeling, various metrics have been proposed to evaluate the context and coherence of different text documents. These include topic coherence, topic diversity, NPMI and Umass.

Given the inherent difficulty of quantitative validation, qualitative assessment by human experts remains as a common practice in academic research (Eklund and Forsman, 2022). More recently, large language models have also been shown to have potential to add scalability while keeping the standard very close to human assessment (Miller and Alexander, 2025).

2.7.2 Current state-of-the-art

With the recent success of transformer based embedding models, the evaluation of semantic clustering has shifted towards evaluating the full pipeline instead of the different components separately. Widely considered as the foundation of recent state-of-the-art pipelines in topic modeling is BERTopic presented by Grootendorst (2022) which considers topic modeling as a clustering task, consisting of transformer embeddings, which are fed to UMAP, and then clustered using HDBSCAN.

HDBSCAN is frequently favored due to few robust theoretical properties. Unlike centroid based methods, the number of clusters do not have to be specified and it is capable of identifying clusters with varying shapes and sizes. HDBSCAN also allows data points to be outliers and does not force them into clusters when they are not aligned with the dense regions. HDBSCAN is supported in textual data clustering context for example by Saha (2023), Wang and Magrabi (2025), Mersha et al. (2024) and Ningrum et al. (2026).

While HDBSCAN seems to be the more widely used clustering algorithm, there are arguments also for agglomerative hierarchical clustering. Ufeli et al. (2025) argue

that noise discarding models, like HDBSCAN, will lose some of the multi-level hierarchical structure. Janssens et al. (2025) point out that HDBSCAN is prone to over-segmentation. Finally, agglomerative hierarchical clustering is often cited to create the most human interpretable results (Janssens et al., 2025), (Märzinger et al., 2021), (Feng et al., 2026).

3 Data and methods

The components of semantic clustering pipeline are tested in practice. The findings from the literature are used to scope the tested methods and to design the validation and evaluation of the results. The goal of the empirical testing is to provide practical insights for the risk clustering tool development.

This section starts by introducing the used risk data sets in Subsection 3.1. Subsection 3.2 describes the used data formatting as a preprocessing step before risk descriptions are embedded. The selected methods from the literature review are listed in Subsection 3.3. Finally, Subsection 3.4 discusses the process of finding the best methods for a risk clustering tool by different evaluation metrics and grid search.

3.1 Data

The performance of the selected methods is compared on three different data sets. Table 3 lists the used risk registers. Two risk registers are mock data provided by Inclus for internal testing. The third one is the MIT AI Risk Repository that contains over 1500 risks imposed by the development of AI (Slattery et al., 2026). The MIT AI Risk Repository is filtered to only entries that contain a full description of the risk. These data sets provide varying formats and labeling which is important for the verification of the clustering results.

Table 3: Used data sets

Data set name	Description	Number of risks
inclus mock data	Mock data containing diverse labeling	58
ERM risks	The outdated operational risks of Inclus	43
MIT AI risks	Register of risks imposed by AI maintained by MIT	1529

The risk registers are in a tabular format where each row contains data for one risk. The columns of the table, vary but all the used risk registers contain fields risk title and description. Additionally, fields such as category, responsible unit, related standards or strategic goals are listed in the mock data set. As the risk registers contain different fields and a varying number of entries, a comprehensive and diverse testing of the developed pipeline is enabled.

3.2 Data formatting

The qualitative risk data stored in risk registers has two main challenges for a semantic analysis pipeline. Firstly, the data contains multiple meaningful fields, including the risk name, description and other labels or keywords such as location or category data. Instead of embedding only the risk description, it would add a lot to the meaning if these different risk characteristics could be included in a meaningful way. The second challenge is that the risk register can be maintained by multiple stakeholders leading to incoherent formatting. It is initially unclear how much this stylistic variance has an effect on the clustering result. The idea of the preprocessing is to filter out the inconsistencies in the data, revealing only the expert intent behind the risk entries in the risk register. The clustering results are compared between preprocessed risk entries and the original data points in the testing phase.

This project uses a LLM-based preprocessing step where the rows from the risk register are normalized. A LLM can handle varying formats and columns labels, avoiding a need to define custom preprocessing pipelines for each separate input risk register. For each row in the risk register, the LLM is prompted to format the descriptions into clear and natural subject, predicate, object format. Additional fields in the register are combined with description by generating declarative sentences from the labels and keywords. For example, if the risk has a label "ISO14001 - Environmental and Management System" on a column "Relates to ISO standard" in the original risk register, a sentence "This risk relates to ISO standard ISO14001 - Environmental and Management System" could be appended to the risk description. The hyphothesis is that the embedding model handles this free text data better than structured formats such as csv or json. This is due to reduced structural noise from the structural tokens and the fact that the models are mainly trained on free text data such as web articles.

3.3 Method selection

Figure 2 demonstrates the process of semantic clustering. The pipeline consists of the preprocessing of risk descriptions or risk register entries, embedding the preprocessed text samples to multidimensional vectors, dimensionality reduction and finally the clustering algorithm. The different methods for each process step are compared on the sample data sets.

Figure 2: Semantic clustering pipeline.

The methods for all the semantic clustering pipeline steps are selected based on

the best practices from the literature. Table 4 presents the selected methods for each step. The results with preprocessed risk descriptions is compared to clusters without preprocessing. The same applies also for dimensionality reduction after the embedding step where no intermediate processing is compares to PCA and UMAP dimensionality reduction methods. For distance metrics and clustering algorithm, the selected methods are Euclidian distance and cosine similarity with agglomerative hierarchical clustering and HDBSCAN clustering algorithm. Some of the methods also have hyperparameters that require tuning for meaningful comparison. The semantic clustering pipeline also allows for using different distance metrics for dimensionality reduction and clustering.

Table 4: Methods selected for pipeline evaluation.

Process Step	**Tested Method**	**Hyperparameters**
Preprocessing	Homogenizing the input data	Format
	No preprocessing	-
Embedding models	F2LLM-v2	-
	Text-embedding-3-large	# of dimensions
	Harrier-oss-v1	Instruction prefix
Dimensionality reduction	PCA	# of components
	UMAP	# of components, # of neighbours, min distance
	No dimensionality reduction	-
Distance metrics	Euclidian distance	-
	Cosine similarity	-
Clustering algorithms	Agglomerative hierarchical	# of clusters, linkage
	HDBSCAN	Min cluster size, min samples, epsilon

For the embedding models, three potential candidates were chosen for further evaluation based on their performance on the MTEB benchmark: F2LLM-v2, Harrier-oss-v1, and text-embedding-3-large. F2LLM-v2, developed by Codefuse-AI, is a family of open-source models with an emphasis on multilingual capabilities. Its 14B-parameter variant is currently the top performer in MTEB's clustering benchmark (Muennighoff et al., 2023). Microsoft's Harrier-oss-v1 is a open-weight suite, whose 27B-parameter version currently ranks second only to three F2LLM-v2 variants on the same benchmark. Notably, Harrier requires a short instruction prompt to describe the specific embedding task being performed. Text-embedding-3-large serves as a proprietary baseline, representing the best-performing OpenAI model on the clustering bench-

mark. Since its benchmark performance is not quite on the level of the two other models, it will be interesting to see if the leaderboard placements translate to differences in semantic risk clustering task.

3.4 Evaluation methods and metrics

Two kinds of metrics are used for clustering result evaluation. Quantitative metrics measure how well the data is partitioned and how separated and compact different clusters are. However, the quantitative metrics depend on the selected distance metrics and the feature space. Furthermore, the pure numerical scores do not tell anything about the meaning of the formed clusters. Qualitative metrics use human-in-the-loop or LLMs to validate the formed clusters, and how do they make sense contextually. This is more laborous than pure numerical scores but is important for the validation of the results. Two dimensional visualizations of the clusters can be used in the validation process. Another option is to use metrics for topic coherence inside the clusters, such as NPMI and Umass. These metrics try to evaluate how similar the used wordings inside the clusters actually are.

The developed pipeline uses silhuette score and Davis-Boulding index as numerical scores to evaluate the different clusterings produced by the pipeline. As the number of different pipeline options becomes large, not all results can be checked manually. The quantitative measures can give indicative results and trends of the differences between clustering methods. After gathering a large number of numerical scores, a few representative samples are selected for more thorough qualitative analysis. Insights from the clusterings and their differences are gathered from visualizations in two dimensions.

To compare different pipeline option listed in Table 4, a comprehensive grid search is used. Here, clustering pipeline is run for all possible combinations of different methods, their hyperparameters and data sets. The formed clusters are then evaluated using numerical scores. The attained data points and trends can then be visualized and analyzed manually.

4 Results

This section presents the empirical findings from the testing phase of the semantic clustering pipeline. The results are evaluated using both statistical metrics as well as visual analysis to identify the most effective configuration for the risk clustering tool.

Subsection 4.1 focuses on the quantitative performance of the tested methods. The pipeline configurations are compared across preprocessing practices, embedding models, dimensionality reduction techniques, and clustering algorithms. To broaden the analysis, Subsection 4.2 explores the interrelationships between the pipeline parametrizations. Finally, Subsection 4.3 builds upon the top-performing configurations to provide a qualitative assessment of the generated risk clusters and their validity.

4.1 Quantitative performance

This subsection assesses the quantitative performance of the semantic clustering pipeline by systematically isolating and testing individual component configurations. The pipeline performance is evaluated using statistical metrics such as the Silhouette score and the Davies-Bouldin index, to measure cluster quality, separation, and density. In addition, embedding execution speeds and total pipeline runtime are monitored by computational benchmarks to measure algorithmic efficiency. To ensure robustness and scalability, these evaluations are performed across three different risk registers: Inclus Mock Data, ERM risks, and MIT AI risks datasets.

4.1.1 Preprocessing

A quantitative comparison was conducted between clustering pipelines utilizing preprocessed data and those using raw data. Figure 3 presents the resulting silhouette scores and embedding times for each dataset via boxplots.

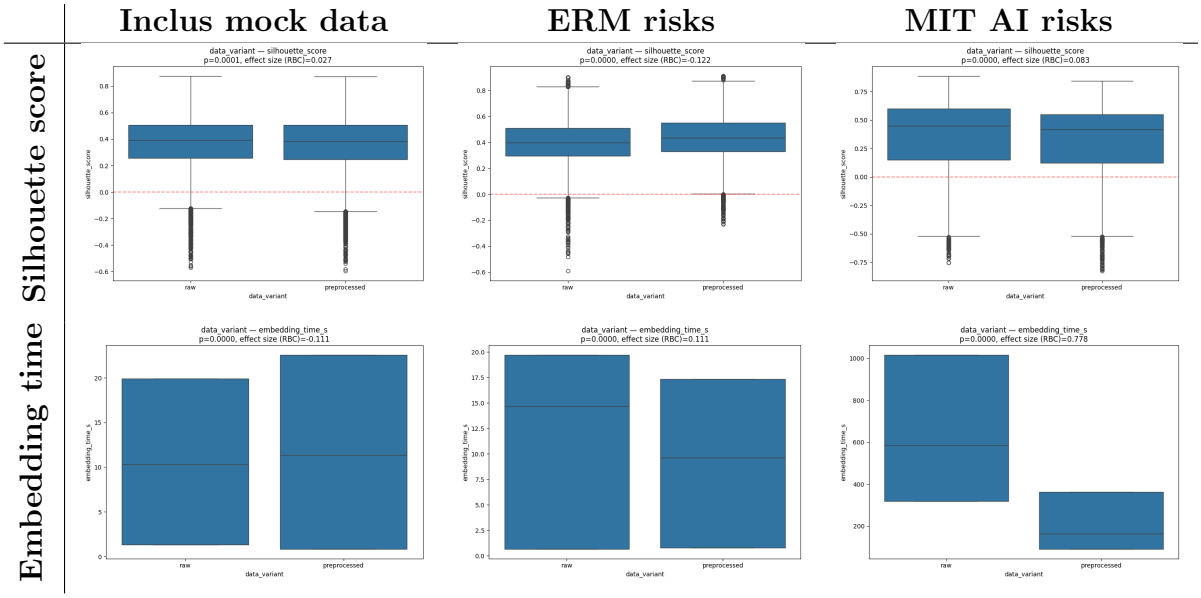

Figure 3: Comparative analysis of preprocessing methods across different datasets. The top row tracks the Silhouette score (y-axis) across two data variants on the x-axis from left to right: (1) raw, and (2) preprocessed. The bottom row displays the corresponding embedding time in seconds using the same x-axis layout.

As shown in Figure 3, the silhouette scores remained consistent across both data variants. However, embedding times exhibited higher variance; notably, the raw data required significantly longer embedding times for the larger MIT AI risks dataset. Preprocessed data, by contrast, stabilized and minimized execution overhead as dataset size increased. Overall, the inclusion of a preprocessing step did not yield a significant improvement in quantitative clustering performance.

4.1.2 Embedding models

The choice of embedding model was evaluated based on clustering quality and computational overhead. Figure 4 illustrates the silhouette scores and embedding times for three models across the target datasets.

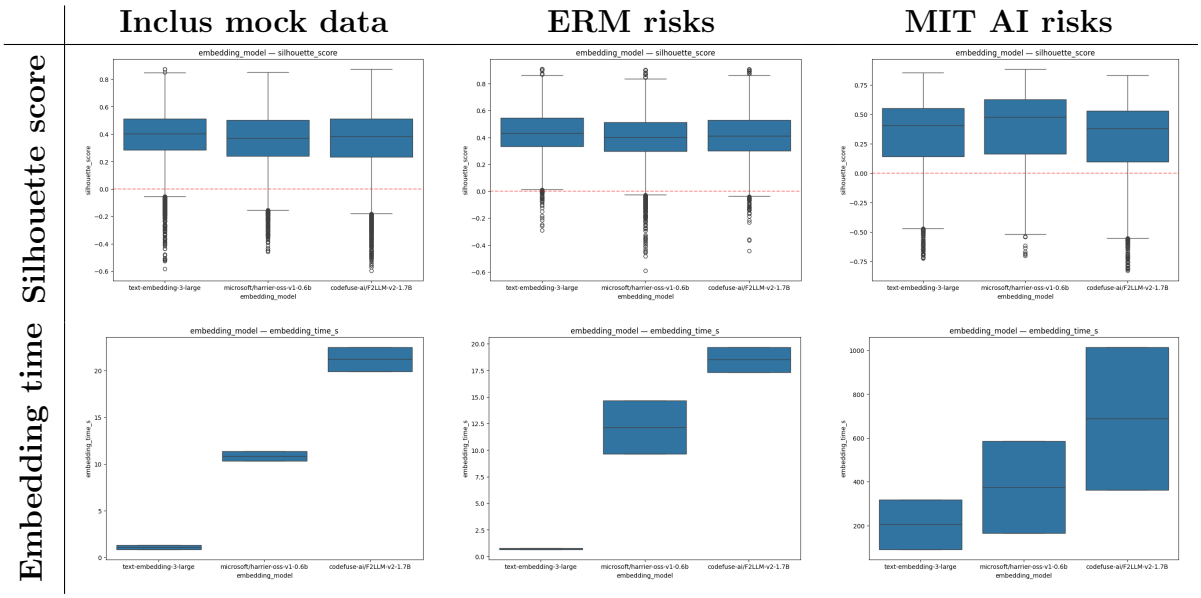

Figure 4: Comparative analysis of embedding models across different datasets. The boxplots isolate the performance of three alternative text embedding models (x-axis), mapped from left to right: (1) text-embedding-3-large, (2) microsoft/harrier-oss-v1-0.6b, and (3) codefuse-ai/F2LLM-v2-1.7B. Performance is evaluated against the Silhouette score (top row) and embedding execution time in seconds (bottom row).

As seen in Figure 4, the silhouette scores remain relatively consistent across all tested models, with medians generally falling between 0.3 and 0.5 for all datasets. This indicates that the choice of model has a marginal impact on the underlying cluster density and separation in these specific contexts.

However, the computational efficiency varies drastically. The bottom row of Figure 4 shows that the API-based `text-embedding-3-large` model is significantly faster than the local alternatives. This disparity is most evident in the larger MIT AI risks dataset, where the OpenAI model completes the task in roughly 200 seconds, whereas the local LLM-based models require between 600 and 1000 seconds. Consequently, while clustering quality is comparable, the API-based approach offers a substantial advantage in processing speed.

4.1.3 Dimensionality reduction

The impact of dimensionality reduction on clustering performance was evaluated by comparing raw embedding space (none) against PCA and UMAP. Figure 5 presents the silhouette scores, Davies-Bouldin indices, and total pipeline times for each configuration.

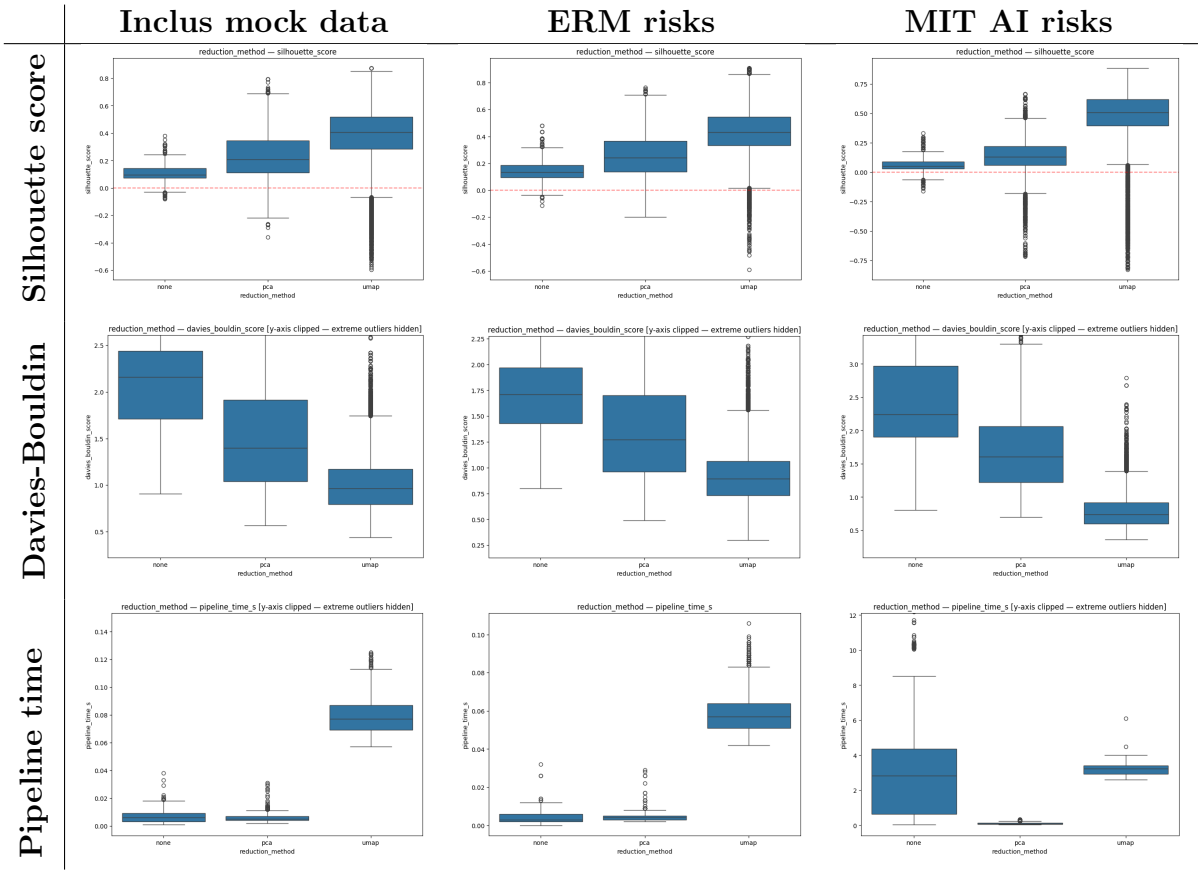

Figure 5: Comparative analysis of dimensionality reduction methods across different datasets. The configurations are compared on the x-axis from left to right: (1) no reduction, (2) PCA, and (3) UMAP.

The analysis in Figure 5 reveals that dimensionality reduction significantly enhances cluster quality across all datasets. UMAP consistently produces the highest silhouette scores and the lowest Davies-Bouldin indices, indicating superior cluster separation and density compared to PCA or using no reduction. While PCA offers a marginal improvement over the raw data, it does not reach the performance levels of UMAP.

Regarding computational efficiency, Figure 5 shows that PCA is the fastest method, often outperforming the "none" baseline by reducing the data complexity before clustering. UMAP introduces a visible increase in pipeline time, though it remains within a reasonable range for these datasets. In conclusion, UMAP is the preferred dimensionality reduction technique due to its substantial positive impact on quantitative metrics, despite the slight increase in processing time.

4.1.4 Clustering algorithm

The final stage of the pipeline evaluation compared different distance metrics and clustering algorithms. Figure 6 briefly illustrates the impact of distance metrics on silhouette scores.

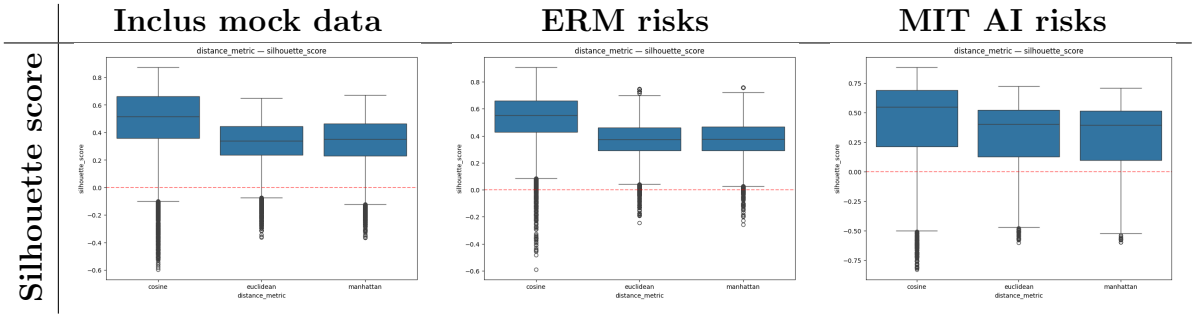

Figure 6: Comparative analysis of distance metrics across different datasets. The boxplots evaluate the performance of alternative mathematical vector space formulas mapped on the x-axis from left to right: (1) Cosine, (2) Euclidean, and (3) Manhattan, measured against the resulting Silhouette score on the y-axis.

As shown in Figure 6, the cosine distance metric consistently provides the highest median silhouette scores across all three datasets, outperforming both Euclidean and Manhattan distances.

The primary analysis focused on the performance of clustering algorithms, comparing Agglomerative clustering against HDBSCAN. Figure 7 details the quantitative metrics for these models.

According to Figure 7, HDBSCAN demonstrates a clear advantage in clustering quality. It achieves significantly higher median silhouette scores and lower Davies-Bouldin indices than Agglomerative clustering across all datasets. In the MIT AI risks dataset, HDBSCAN reaches silhouette scores above 0.5, whereas Agglomerative clustering remains closer to 0.25.

Computational efficiency, represented by the pipeline time in Figure 7, shows that both algorithms perform similarly on smaller datasets (Mock and ERM). However, Agglomerative clustering shows slightly higher variance and total time on the larger MIT AI risks dataset. Given the superior density and separation metrics, HDBSCAN is identified as the optimal clustering algorithm for this pipeline.

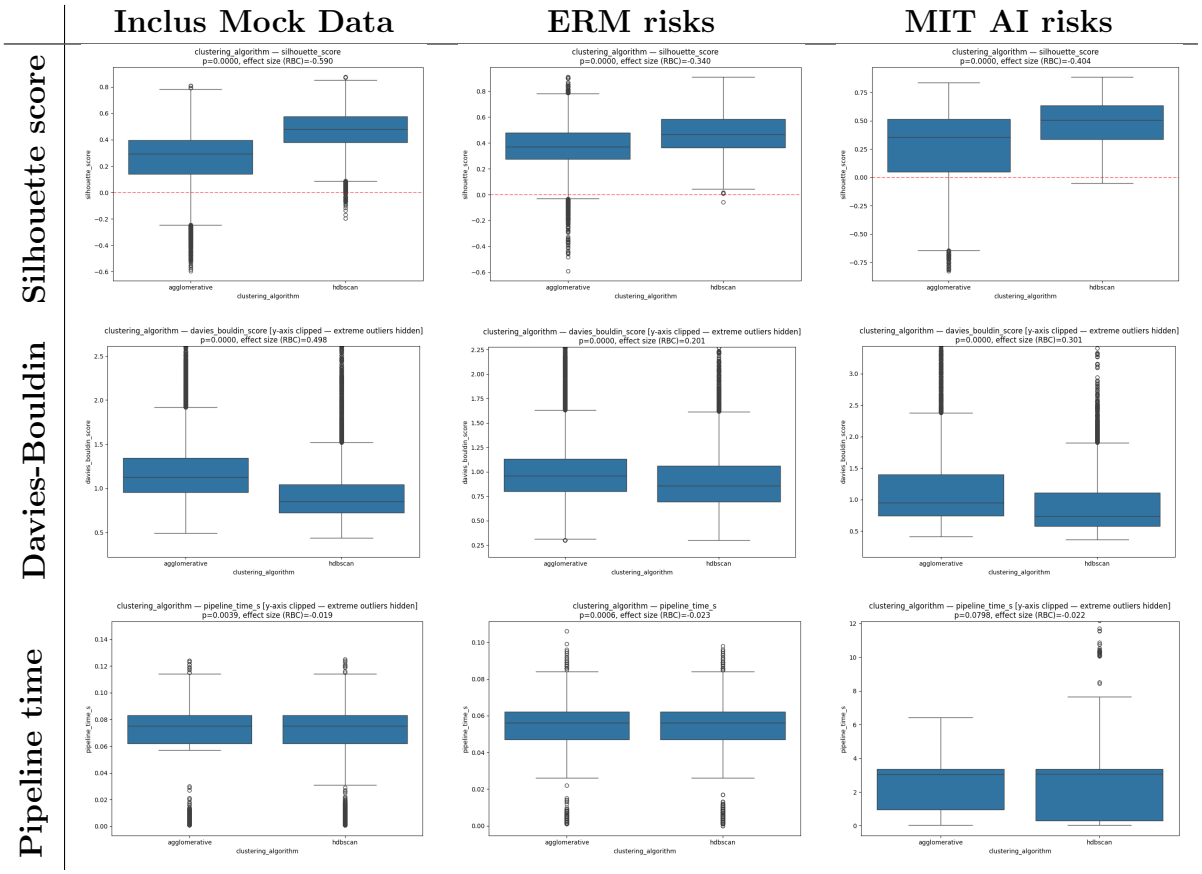

Figure 7: Comparative analysis of clustering models across different datasets. Performance is evaluated across three vertical rows: Silhouette score, Davies-Bouldin index, and entire pipeline computational runtime in seconds. The individual algorithmic configurations are compared on the x-axis from left to right: (1) Agglomerative Hierarchical Clustering and (2) HDBSCAN.

4.2 Interrelationship between pipeline parameterizations

In addition to straightforward comparison between methods, a short pairwise analysis was conducted to detect possible interrelationships between the different parts of the pipeline. It is beneficial to understand whether certain methods reinforce one another or are incompatible with each other. For instance, Euclidean and Manhattan distance measures can suffer from data sparsity and other inherent properties of high-dimensional spaces, suggesting they could perform better when used with dimensionality reduction techniques than without them.

From the heatmap shown in Figure 8, a drop on performance of the pipeline using Euclidean or Manhattan distance measures is visible on no dimensionality reduction compared to PCA and UMAP. However, the drop is visible also for cosine distance indicating that it is overall worse to not use any dimensionality reduction method. In general, the combined effects seem to be consistent with the performance of individual components and do not indicate any specifically better or worse parameter

combinations.

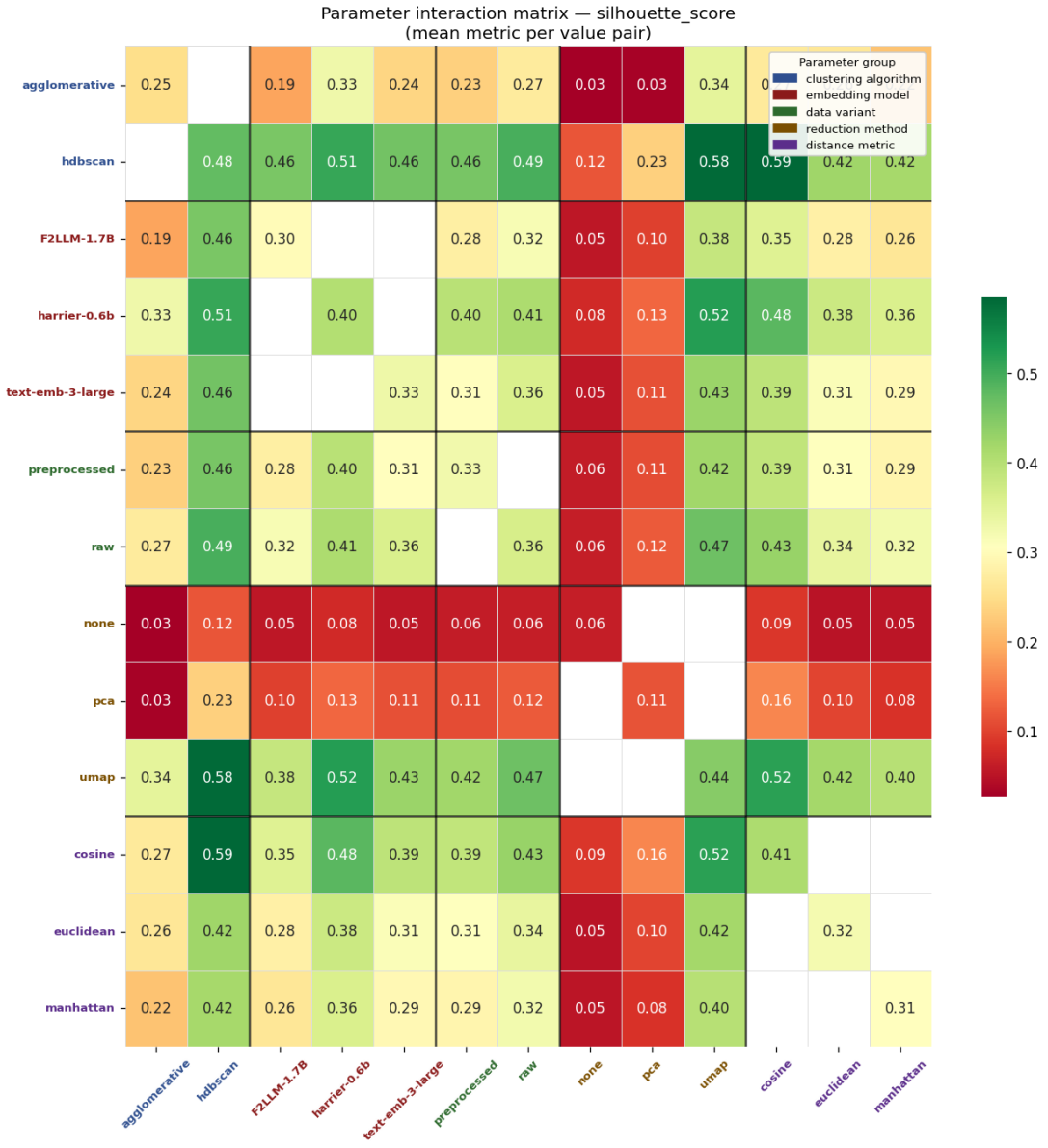

Figure 8: Parameter interaction matrix on the MIT AI risk dataset. The heatmap plots different combinations of pipeline components against each other to track their pairwise combined effect on the silhouette score. Both axes display an identical set of modular pipeline options, which are plotted chronologically from top to bottom on the vertical axis and from left to right on the horizontal axis, across five tested categories: (1) Agglomerative and HDBSCAN clustering models; (2) F2LLM-1.7B, harrier-0.6b, and text-emb-3-large embedding models; (3) raw and preprocessed data variants; (4) None, PCA, and UMAP dimensionality reductions; and (5) Cosine, Euclidean, and Manhattan distance metrics.

4.3 Qualitative assessment

The top performing pipeline configurations based on quantitative metrics were chosen for a qualitative assessment. This included a two-dimensional visualization of the final clustering, a keyword extraction using Term Frequency - Inverse Document Frequency (TF-IDF) method and a human assessment of the cluster cohesion and keyword suitability. A visualization of the MIT AI risk dataset is shown in figure 9. This gives a rough intuition whether the clustering is at all successful or not. TF-IDF is a Bag-of-Words (BoW) method which obtains labelings for each of the clusters. For each cluster, TF-IDF identifies words that appear frequently within that cluster but rarely across the other clusters. These words act as keywords for the specific cluster and should identify it uniquely. Table 5 shows few examples of two separate clusters. These are of course only a representative example and the human assessment on the cluster cohesion and comparing the keywords to the clusters were more thorough than these few risks, however, the full dataset is not listed here.

Table 5: Cluster keywords and example representative risks for selected clusters of the MIT AI risk dataset.

Cluster Details	**Core TF-IDF Keywords**	**Risk Description**
Cluster 0 Size: $N = 114$	attacks, vulnerabilities, attack, adversarial, prompt, inference	"Programmers are accustomed to using code generation tools such as Github Copilot for program development, which may bury vulnerabilities in the program."
		"The software development toolchain of LLMs is complex and could bring threats to the developed LLM."
Cluster 1 Size: $N = 82$	sensitive, leaking, inferring, correctly, compromise, personal	"Privacy Leakage means the generated content includes sensitive personal information."
		"The model is trained with personal data in the corpus and unintentionally exposing them during the conversation."

The qualitative assessment yielded two primary conclusions. First, the different Bag-of-Word based methods can be useful, however, as they rely on statistical keyword matching, they suffer from all the unpleasant properties of heterogeneous text data that text embeddings are solving with better success. Thus, TF-IDF can be a computationally cheap solution which is better than going through hundreds of risks by hand, but which will likely be outperformed by a LLM based labeling. Second, one has to be careful with HDBSCAN noise and how it affects the quantitative metrics. On one hand, as visually demonstrated in Figure 9, genuine outliers are isolated successfully. These can be the most important and interesting data points of the risk register. On the other hand, as the quantitative metrics prefer more compact clusters, data might be left out as noise too easily. The clusters and surrounding noise elements of MIT dataset are shown in the two-dimensional visualization in Figure 9.

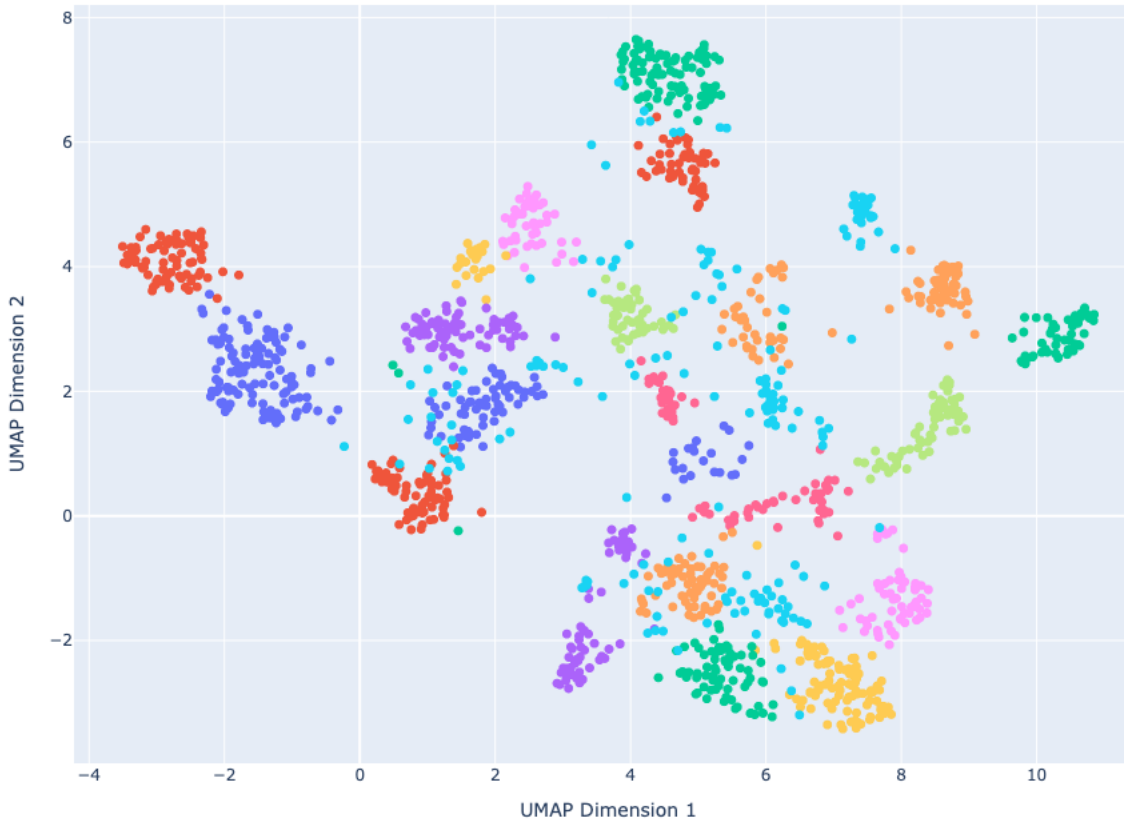

Figure 9: 2D visualization of the MIT dataset

5 Discussion

The results indicate that the developed risk clustering pipeline operates as intended. Yet, it is essential to discuss the empirical findings as well as evaluate the overall performance of the tool. By analyzing the strengths and operational constraints, this section also provides recommendations for further development.

Subsection 5.1 addresses the practical usability of the tool, focusing on operator accessibility, system adaptability, and the management of dynamic modifications. This discussion smoothly transitions into Subsection 5.2, which evaluates the inherent limitations and weaknesses of the current framework. Lastly, building on the previous sections' discussion, Subsection 5.3 presents potential improvements of the tool for further development.

5.1 Usability

Translating the experimental semantic clustering pipeline into a production tool involves some usability considerations.

Risk registers are not static databases, as new risks are continuously added. A production tool should efficiently handle individual data modifications. The current methodology processes datasets in bulk, which may raise concerns regarding the computational overhead of continuous updates. However, empirical testing indicates that this processing overhead is generally small for standard applications. During testing, executing the dimensionality reduction and clustering on a large dataset of approximately 1800 risks took only about 1.5 to 3 seconds. When a user adds a single new risk, the computationally heavy embedding process is isolated strictly to that new text. Recalculating the clusters using the combined vectors is very efficient. Therefore, the most robust and solution is to execute a full recalculation of the pipeline after every data modification.

However, if the tool is eventually scaled to accommodate extremely large databases, constant recalculation may begin to induce user interface lag. Therefore, implementing incremental clustering approaches may be needed. The technical viability of these updates depends on the pipeline's method choices. For dimensionality reduction, PCA allows immediate linear projection of new embeddings via an existing transformation matrix and UMAP supports nonlinear iterative procedure that maps new risk into the established space. For clustering, HDBSCAN supports approximate predictions, assigning new points to established clusters based on density without refitting the model. Conversely, agglomerative hierarchical clustering requires rebuilding the entire distance matrix, making it incompatible with direct incremental updates. As an alternative clustering method agnostic, an updated system could simply assign the new risk entry to the closest existing cluster. However, both proximity assignment and HDBSCAN approximation only map data to existing categories but cannot form entirely new clusters. Further, if cluster centroids or dimensionality reduction spaces are not recalculated after adding a risk they eventually become inaccurate. Thus, for

larger registers a hybrid approach may be beneficial: daily additions and removals are handled via rapid proximity assignment or approximation, and a recalculation is triggered once a defined volume threshold or time interval is reached.

Empirical results indicate that the best pipeline configuration depends on the input data. Hyperparameters must adapt to the specific dataset to yield good results. To ensure the tool remains usable without requiring constant intervention or extensive technical knowledge, the system could implement an automated quantitative validation framework. Whenever an entirely new dataset is imported the software automatically executes a small grid search to determine the best configuration for that specific data iteration.

Finally, to explore further computational efficiency improvements, we experimented with developing a universal dimensionality reduction model. The hypothesis was that aggregating all available corporate risk datasets to calculate a master PCA matrix would accelerate the pipeline by eliminating the need to fit new models for every individual risk register. Testing revealed that a global model produces higher variance and slightly lower average accuracy, as risk vocabularies are highly heterogeneous across different registers. Dataset-specific dimensionality reduction does not consume prohibitive computing resources and preserves the ability to utilize UMAP, which our results demonstrated to be the better technique.

5.2 Limitations

Although the semantic clustering pipeline works well, it is important to consider limitations related to the algorithm performance, data privacy, and architecture.

We first examine how outliers are handled. HDBSCAN filters out infrequent data points as noise, but in risk management, an outlier may represent a rare, significant event. The randomness of UMAP can cause some consistency issues. If random numbers are not specified precisely, running the same data multiple times can yield different visual maps and cluster boundaries.

One potential limitation is that there is no universal configuration for the pipeline. The optimal combination of embedding models, dimensionality reduction, and distance metrics varies depending on the dataset. Due to this variability, it is difficult to deploy the tool as-is without manual fine-tuning for each new risk register.

There is a trade-off between security and performance. If the top-performing embedding models rely on cloud-based APIs, this can cause issues with highly confidential risk data, since organizations may be restricted from using external clouds.

Finally, as already noted in the Subsection 5.1, the pipeline is limited by its inability to handle dynamic risk registers, since adding new risks to the database currently requires running the entire pipeline from the beginning.

5.3 Further development

Previous subsections have already pointed out several practical ways to improve the risk clustering tool. To transform this prototype into a fully realized risk management application, future development could focus on the following areas: allowing real-time data updates without heavy recalculations, investigating the pipeline performance with different languages, taking into account how different risks affect one another, and creating a user-friendly interface where analysts could easily review and correct the system's groupings.

Perhaps the most important improvement in the pipeline is the ability to make changes, such as adding new risks to the register, without having to run the entire pipeline from the beginning. As noted in Subsection 5.1, recalculating everything due to minor changes is slow and computationally expensive in large risk registers. Future versions should explore ways to instantly place new risks into existing clusters. By using mathematical shortcuts built into algorithms like HDBSCAN or using saved PCA matrices, the tool could determine new risks immediately. This would keep the system fast and responsive without disrupting the existing cluster structure.

Another important step is to expand the tool's language capabilities. Currently, the datasets used in this study have been in English. However, the data could also contain risks in Finnish or Swedish, or even a mixture of multiple languages. Therefore, it could be useful to assess, as part of the future development work, how well different embedding models handle multilingual data. The goal is to ensure that similar risks, regardless of the language, are mapped to the same geometric space.

One way to go beyond simple word similarities is to examine and understand how different risks interact with one another. For example, descriptions of an IT system failure and a financial loss may use entirely different vocabulary, yet they are directly linked in practice. By combining the semantic text embedding pipeline with a Knowledge Graph or a network map of risk dependencies, the tool could group risks based on their cause-and-effect relationships alongside simple word matching.

Finally, it would be useful to introduce a feature that would allow operators to easily fix the system's mistakes. Instead of displaying passive results, this interface would allow analysts to manually correct mistakes, such as dragging a misplaced risk into a better group or merging two categories together. The system could then save these manual corrections as training rules, allowing the underlying algorithm to adapt and align with human expertise over time. Additionally, an LLM model could be integrated to automatically generate a clear title and a short summary for each final cluster.

6 Conclusion

This project has designed and evaluated a semantic risk clustering pipeline for qualitative risk data. The risk descriptions listed in risk register are used to build meaningful visualizations of the risk landscape. To enable further development of the tool for Inclus and its customers, a set of different pipeline choices were compared. These pipeline configurations include preprocessing practices, embedding model, dimensionality reduction and clustering algorithm. The comparison of the options were conducted with a comprehensive literature review and manual testing with actual risk registers.

The evaluation and validation framework included both qualitative measures using metrics such as silhuette score and Davis-Boulding index and quantitative analysis with 2D-visualizations and keyword matching. The pipeline with UMAP dimensionality reduction and HDBSCAN clustering algorithm performed better than other options over all tested data sets consistently. As significant differences were not found with the embedding model choice, although the used model had an effect on the processing time. The advantage on a preprocessing and homogenizing of the input risk entries depends on the data set. If the risk description length vary significantly, preprocessing seems to help to form more concistent clusters.

On top of the used technologies, also other parameters affect the clustering result. Hyperparameters, such as the number of clusters and the number of dimensions used in the clustering, must be validated for each case and data set seprately. For this process, a grid search of all the options is found to be useful tool. However, the number of options has to be limited and carefully chosen for the final tool to be usable for the end customers. The testing framework and reports developed during the project will give Inclus the required tools to continue the testing work and to develop and integrate the functionalities into the existing Inclus toolkit.

References

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, pages 420–434. Springer, 2001.

Terje Aven. The risk concept—historical and recent development trends. *Reliability Engineering & System Safety*, 99:33–44, March 2012. ISSN 0951-8320. doi: 10.1016/j.ress.2011.11.006. URL https://www.sciencedirect.com/science/article/pii/S0951832011002584.

Kamal Berahmand, Fatemeh Daneshfar, Maryam Dorosti, Mohammad Javad Aghajani, et al. An improved deep text clustering via local manifold of an autoencoder embedding. *Research Square*, 2022.

James C Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer Science & Business Media, 2013.

Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–51, 2015.

Hikmet Canli. Evaluating the semantic clustering power of llms on erp support texts. In *Proceedings of the 9th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–7, 2025. doi: 10.1109/IDAP68205.2025.11222140.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Rian Dolphin, Joe Dursun, Jarrett Blankenship, Katie Adams, and Quinton Pike. Taxonomy-aligned risk extraction from 10-K filings with autonomous improvement using llms, January 2026. URL http://arxiv.org/abs/2601.15247. arXiv:2601.15247 [cs] version: 1.

Anton Eklund and Mona Forsman. Topic modeling by clustering language model embeddings: Human validation on an industry dataset. In *Proceedings of Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 635–643, 2022.

Abdolmajid Erfani and Hussein Khanjar. Large language models for construction risk classification: A comparative study. *Buildings*, 15(18), September 2025. ISSN 2075-5309. doi: 10.3390/buildings15183379. URL https://www.mdpi.com/2075-5309/15/18/3379.

Jadeyn Feng, Allison Lau, Melinda Hodkiewicz, Caitlin Woods, and Michael Stew-

art. Semantic and engineering-based embedding for classification list development. *Machine Learning and Knowledge Extraction*, 8(3):61, 2026.

Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure, March 2022. URL http://arxiv.org/abs/2203.05794. arXiv:2203.05794 [cs].

Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, et al. Kalm-embedding: Superior training data brings a stronger embedding model. *arXiv preprint arXiv:2501.01028*, 2025.

Wannes Janssens, Matthias Bogaert, and Dirk Van den Poel. A Comparative Analysis of Topic Reduction Techniques for BERTopic. *IEEE Access*, 13:204087–204103, 2025. doi: 10.1109/ACCESS.2025.3638956.

Ian Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, pages 1945–1948. Springer, 2025.

Adrian Kuhn, Stéphane Ducasse, and Tudor Gîrba. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230–243, 2007. ISSN 0950-5849. doi: https://doi.org/10.1016/j.infsof.2006.10.017. URL https://www.sciencedirect.com/science/article/pii/S0950584906001820. 12th Working Conference on Reverse Engineering.

M. C. Leva, N. Balfe, B. McAleer, and M. Rocke. Risk registers: Structuring data collection to develop risk intelligence. *Safety Science*, 100:143–156, December 2017. ISSN 0925-7535. doi: 10.1016/j.ssci.2017.05.009. URL https://www.sciencedirect.com/science/article/pii/S0925753517308627.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Haofu Liao, Aruni RoyChowdhury, Weijian Li, Ankan Bansal, Yuting Zhang, Zhuowen Tu, Ravi Kumar Satzoda, R. Manmatha, and Vijay Mahadevan. Doctr: Document transformer for structured information extraction in documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19527–19537, 2023. doi: 10.1109/ICCV51070.2023.01794.

Le Ma, Ran Zhang, Yikun Han, Shirui Yu, Zaitian Wang, Zhiyuan Ning, Jinghan Zhang, Ping Xu, Pengjiang Li, Wei Ju, Chong Chen, Dongjie Wang, Kunpeng Liu, Pengyang Wang, Pengfei Wang, Yanjie Fu, Chunjiang Liu, Yuanchun Zhou, and Chang-Tien Lu. A comprehensive survey on vector database: storage and retrieval technique, challenge, June 2025. URL http://arxiv.org/abs/2310.11703. arXiv:2310.11703 [cs].

Thomas Märzinger, Jan Kotík, and Christoph Pfeifer. Application of hierarchical agglomerative clustering (HAC) for systemic classification of pop-ip housing (PUH) environments. *Applied Sciences*, 11(23):11122, 2021.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Vivek Mehta, Seema Bawa, and Jasmeet Singh. Analytical review of clustering techniques and proximity measures. *Artificial Intelligence Review*, 53(8):5995–6023, December 2020. ISSN 0269-2821, 1573-7462. doi: 10.1007/s10462-020-09840-7. URL https://link.springer.com/10.1007/s10462-020-09840-7.

Melkamu Abay Mersha, Mesay Gemeda Yigezu, and Jugal Kalita. Semantic-driven topic modeling using transformer-based embeddings and clustering algorithms. *Procedia Computer Science*, 244:121–132, 2024. ISSN 18770509. doi: 10.1016/j.procs.2024.10.185. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050924029867.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, September 2013. URL http://arxiv.org/abs/1301.3781. arXiv:1301.3781 [cs].

Justin K. Miller and Tristram J. Alexander. Human-interpretable clustering of short text using large language models. *Royal Society Open Science*, 12(1): 241692, January 2025. ISSN 2054-5703. doi: 10.1098/rsos.241692. URL https://royalsocietypublishing.org/doi/10.1098/rsos.241692.

Niklas Muennighoff. Sgpt: GPT sentence embeddings for semantic search, 2022. URL https://arxiv.org/abs/2202.08904.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: An overview. *WIREs Data Mining and Knowledge Discovery*, 2(1):86–97, January 2012. ISSN 1942-4787, 1942-4795. doi: 10.1002/widm.53. URL https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.53.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

Ariska Fitriyana Ningrum, Dannu Purwanto, and Abdel Nasser Sharkawy. Evaluating clustering methods for semantic representation of disaster news using BERT embeddings and HDBSCAN. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 11(3):784–794, February 2026. ISSN 2527-4864, 2685-8223. doi: 10.33480/jitk.v11i3.7204. URL https://ejournal.nusamandiri.ac.id/index.php/jitk/article/view/7204.

Alina Petukhova, João P. Matos-Carvalho, and Nuno Fachada. Text clustering with large language model embeddings. *International Journal of Cognitive Computing in Engineering*, 6:100–108, December 2025. ISSN 26663074. doi: 10.1016/j.ijcce.2024.11.004. URL https://linkinghub.elsevier.com/retrieve/pii/S2666307424000482.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Rohan Saha. Influence of various text embeddings on clustering performance in NLP, May 2023. URL http://arxiv.org/abs/2305.03144. arXiv:2305.03144 [cs].

Grigori Sidorov, Alexander Gelbukh, Helena Gómez-Adorno, and David Pinto. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, 18(3):491–504, September 2014. ISSN 1405-5546. doi: 10.13053/CyS-18-3-2043. URL http://www.scielo.org.mx/scielo.php?script=sci_abstract&pid=S1405-55462014000300007&lng=es&nrm=iso&tlng=en.

Peter Slattery, Alexander K Saeri, Emily AC Grundy, Jess Graham, Michael Noetel, Risto Uuk, James Dao, Soroush Pour, Stephen Casper, and Neil Thompson. The AI risk repository: A meta-review, database, and taxonomy of risks from artificial intelligence. *Patterns*, 7(5):101517, 2026. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2026.101517. URL https://www.sciencedirect.com/science/article/pii/S2666389926000267.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3: 583–617, 2002.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.71. URL https://aclanthology.org/2023.findings-acl.71/.

Chukwutem Pinic Ufeli, Mian Usman Sattar, Raza Hasan, and Salman Mahmood.

Enhancing customer segmentation through factor analysis of mixed data (famd)-based approach using k-means and hierarchical clustering algorithms. *Information*, 16(6):441, 2025.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. Large language models enable few-shot clustering, 07 2023.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024a. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text Embeddings by Weakly-Supervised Contrastive Pre-training, February 2024b. URL http://arxiv.org/abs/2212.03533. arXiv:2212.03533 [cs].

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand, August 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.642. URL https://aclanthology.org/2024.acl-long.642/.

Ying Wang and Farah Magrabi. Assessing the transferability of BERT to patient safety: Classifying multiple types of incident reports. *BMJ Health & Care Informatics*, 32(1):e101146, August 2025. ISSN 2632-1009. doi: 10.1136/bmjhci-2024-101146. URL https://informatics.bmj.com/lookup/doi/10.1136/bmjhci-2024-101146.

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

Alexander Westergård. Improving risk registers with an AI assistant. Master's thesis, School of Science, Aalto University, 2025.

Chibok Yang and Yangsok Kim. Enhancing topic coherence and diversity in document embeddings using LLMs: A focus on bertopic. *Expert Systems with Applications*, 281:127517, 2025. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2025.127517. URL https://www.sciencedirect.com/science/article/pii/S095741742501139X.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An LLM-based framework for knowledge graph construction, 2024. URL https://arxiv.org/abs/2404.03868.

Peng Zhang, Suge Wang, Deyu Li, Xiaoli Li, and Zhikang Xu. Combine topic

modeling with semantic embedding: Embedding enhanced topic model. *IEEE Transactions on Knowledge and Data Engineering*, 32(12):2322–2335, 2020. doi: 10.1109/TKDE.2019.2922179.

Xingyu Zhang, Yanshan Wang, Yun Jiang, Charissa Pacella, and Wenbin Zhang. Integrating structured and unstructured data for predicting emergency severity: an association and predictive study using transformer-based natural language processing models. *BMC Medical Informatics and Decision Making*, 24, 12 2024. doi: 10.1186/s12911-024-02793-9.

Ziyin Zhang, Zihan Liao, Hang Yu, Peng Di, and Rui Wang. F2LLM technical report: Matching SOTA embedding performance with 6 million open-source data. *CoRR*, abs/2510.02294, 2025. doi: 10.48550/ARXIV.2510.02294. URL https://doi.org/10.48550/arXiv.2510.02294.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Hammerla. Correlation Coefficients and Semantic Textual Similarity. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 951–962, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1100. URL https://aclanthology.org/N19-1100/.

Yang Zou, Arto Kiviniemi, and Stephen W. Jones. Retrieving similar cases for construction project risk management using Natural Language Processing techniques. *Automation in Construction*, 80:66–76, August 2017. ISSN 0926-5805. doi: 10.1016/j.autcon.2017.04.003. URL https://www.sciencedirect.com/science/article/pii/S0926580517303175.

7 Self-assessment

This self-assessment reflects on the execution of the project by addressing four key questions: how closely did the implementation follow the initial plan, what were the most significant successes and shortcomings, and what insights can be offered in hindsight.

(1) How closely did the actual implementation of the project follow the initial project plan? Were there any major departures and, if so, what?

Overall, the project's implementation closely followed the key objectives and architectural milestones defined in the initial project plan. However, two deviations from the original plan emerged during the development phase, one structural and one related to the schedule, both of which nevertheless contributed to the project's progress.

The structural change relates to the analysis of the models. According to the original task list, the analysis of text embedding models and clustering algorithms would have been performed and the models tested separately before combining them. However, the entire pipeline was evaluated as a single entity from the outset. We realized early on that the performance of clustering depends heavily on the preceding dimensionality reduction and distance metrics, which is why separate testing would have been counterproductive. Furthermore, it made more sense to run the entire process chain as a single unit so that we could better understand how the choices made for individual components affected the final Silhouette scores and cluster arrangements.

The change in schedule is related to the preparation of this final report. Originally, we had planned to write this final report steadily and continuously throughout the whole course, leaving most of the writing work for the later stages of the project –with the exception of the literature review, which we intended to prepare right at the beginning. However, we began drafting the final report at a very early stage and found that writing it was actually beneficial, as it helped us justify practical design solutions theoretically. In addition, preparing the necessary presentations was easier because we already had so much usable content ready.

(2) In what regard was the project successful?

The project yielded positive results in terms of both technical implementation and project management efficiency. Most importantly, the end result met the client's expectations, and the client was satisfied with the risk clustering tool that was developed.

We were able to produce a functional risk clustering tool prototype and comprehensively address nearly all initial research questions. From a project management perspective, our strategy, in which development and writing tasks were handled in advance, provided a comfortable buffer that completely eliminated last-minute panic

at the end of the project. This was especially helpful when, in the end, we no longer had as much time for the project due to other responsibilities and tasks.

In addition, communication within the team and with the client went very smoothly, and we kept them up to date on what had been done and they commented if something needed to be changed. In fact, we managed to keep weekly meetings with the team going almost until the end of the project. We also met with the client quite a few times to keep the project requirements clear. In the end, none of the risks identified during the planning phase materialized into actual obstacles, which was largely thanks to good communication.

(3) In what regard was it less so?

Despite the project's overall success, certain areas proved challenging and left room for improvement.

The primary technical challenge was the unambiguous validation of clustering results and LLM outputs. Verifying unsupervised machine learning models and text labels is inherently subjective and lacks a single objective truth. Silhouette score might provide mathematical clarity regarding cluster density and separation, however it does not necessarily reflect business utility. Additionally, determining whether a cluster makes logical sense requires subjective qualitative evaluation.

Also, a minor note about the literature review: it may have included an excessive and unnecessary number of references. It was difficult to determine which pieces of information actually offered insight and which were irrelevant to the topic and out of scope.

In terms of procedures, the team experienced a significant slowdown in momentum right at the very end. Since the core engineering, data processing, and preliminary reporting were completed early, there was a certain amount of complacency in the final phase. The initial enthusiasm that drove the team forward during the first four months did not fully last until the final stretch of May. Maintaining that energy would have allowed us to further deepen our empirical data analysis and explore more experimental variations.

(4) What could have been done better, in hindsight?

In hindsight, the project ran smoothly without major systemic problems. The project team worked efficiently, although stricter internal deadlines for the final analytical conclusions could have prevented the momentum from slowing down in the final stages of the project. The client was very supportive and provided clear data and feedback, which prevented scope from expanding. To conclude, the biggest improvement would have been better energy in the final phase, which would have led to more in-depth analysis of the results and even some adjustments to the model mentioned in the "Discussion" Section 5.