SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of

ECONOMICS, MANAGEMENT, AND STATISTICS

Ph.D. program: **Economics and Statistics** Cycle: **XXXV**
Curriculum: **Statistics**

HOLISTIC APPROACH TO OPERATIONAL RISK: ISSUES, SOLUTIONS, AND DECISION MAKING

Surname: **PIACENZA**
Name: **FABIO**
Registration number: **854276**

Supervisor: Prof. **FRANCESCA GRESELIN**
Co-Supervisor: Prof. **RIČARDAS ZITIKIS**

Academic Year: 2022-2023

Abstract

Operational risk (OpRisk) emerges as a pivotal non-financial concern with far-reaching implications for financial institutions. Departing from conventional regulatory tasks encompassing data collection, capital requirement calculations, and report generation for managerial decisions, OpRisk functions are now actively pursuing proactive strategies to forestall or alleviate risk impacts. For instance, artificial intelligence techniques, increasingly integral for managerial insights, are now employed to extract additional information from data. This study advances the application of text analysis techniques, a foundational element of Natural Language Processing, to OpRisk event descriptions. The present work introduces a structured workflow for the application of text analysis techniques to the OpRisk event descriptions to identify managerial clusters (more granular than regulatory categories) representing the root causes of the underlying OpRisks.

However, these potent approaches exhibit limitations in influencing the impact of future loss events. In response, this research delves into the augmentation of traditional data sources, exploring alternative channels to identify potential events in their nascent stages and proactively manage their impact. An innovative facet involves the analysis of relevant tweets from X (formerly Twitter) for continuous scanning of the changing risk environment, aiming to detect early warnings about new types of potentially risky events. We demonstrate the seamless integration of these diverse methodologies into a comprehensive approach to OpRisk management, fostering a more holistic, forward-looking, and adaptive risk mitigation strategy.

The thesis is organized as follows. Chapter 1 introduces the most discussed concepts, starting from the operational risk, and continuing with the main integrated statistical methodologies, *i.e.*, text analysis, word embedding, uniform manifold approximation and projections (UMAP) for dimensionality reduction, and latent Dirichlet allocation (LDA) for topic modelling. Chapter 2 defines a first general workflow for OpRisk event descriptions analysis, identifying the funda-

mental steps (*i.e.*, text cleaning, vectorization, semantic adjustment, dimensionality reduction, and clustering) and applying it to a limited and quality assured data set, for which it was feasible to verify the accuracy through an intense involvement of OpRisk analysts. Chapter 3 generalizes and extends this workflow in several directions. In particular, the effort required from the OpRisk analysts is strongly reduced by the application of more efficient methodologies. Among them, we illustrate the benefits of employing powerful data representation based on dimension reduction, and of performing clustering fully based on topic modelling techniques. In addition, the analysis is applied to a much more challenging OpRisk data set, because it is much larger (the number of descriptions is around 100 times larger, while the number of terms is 5 times larger than the previous data set) and the descriptions are multi-language (*i.e.*, not all written in English). Moreover, the analysis is extended to social media data, to be forward-looking and provide OpRisk early warnings. Chapter 4 concludes the manuscript, summarizing the main contributions and sketching future research developments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the main concepts discussed in the thesis, starting from the operational risk, and continuing with the main integrated statistical methodologies, *i.e.*, text analysis, word embedding, Uniform Manifold Approximation and Projections (UMAP) for dimensionality reduction, and Latent Dirichlet Allocation (LDA) for topic modelling.

1.1 Operational risk

Operational risk (OpRisk) is a crucial aspect of risk management that encompasses the potential for losses resulting from inadequate or failed internal processes, systems, people, or external events. Understanding OpRisk is vital for organizations across various sectors to enhance resilience and ensure sustainable operations (Basel Committee on Banking Supervision, 2010).

OpRisk stands as a critical component within the broader landscape of risk management, encompassing a spectrum of potential threats that can impact an organization's ability to achieve its objectives. Unlike market and credit risks, which are often associated with financial instruments, OpRisk extends beyond financial considerations to encompass a diverse range of factors that can disrupt normal business operations (Lam, 2003).

1.1.1 Definition of OpRisk

OpRisk is commonly defined as the risk of loss resulting from inadequate or failed internal processes, systems, people, or external events. These risks are inherent in the day-to-day operations of an organization and can manifest in various forms, including human error, technology failures, fraud, legal and compliance issues, and external events such as natural disasters or geopolitical events (Power, 2005).

1.1.2 Main drivers of OpRisk

Understanding the drivers that contribute to OpRisk is crucial for organizations to proactively manage and mitigate potential threats (Carreño, 2013). The main drivers of OpRisk can be categorized as follows:

- People: human factors play a significant role in OpRisk. This includes errors or omissions by employees, inadequate training, and workforce-related issues. Understanding the human element is essential for developing effective risk mitigation strategies.

- Processes: inadequate or failed internal processes can lead to operational failures. This includes deficiencies in workflow, control systems, and operational procedures. Organizations need to continuously assess and enhance their processes to minimize the risk of disruptions.

- Systems: technological advancements have introduced new opportunities and challenges. OpRisk can arise from system failures, cyber threats, and technological deficiencies. The increasing reliance on complex systems necessitates robust risk management practices in the realm of technology.

- External Events: OpRisk is also influenced by external events that are beyond an organization's control. Natural disasters, political instability, and economic downturns are examples of external factors that can disrupt operations. Building resilience to such events is a key aspect of OpRisk management.

1.1.3 OpRisk management importance in operational resilience

Effectively managing OpRisk is crucial for maintaining the stability and sustainability of organizations. Operational disruptions can result in financial losses, reputational damage, and legal consequences. A robust OpRisk management framework enables organizations to identify, assess, and mitigate potential risks, thereby enhancing overall resilience (Fraser and Simkins, 2002).

According to European Banking Authority (2022), OpRisk has become increasingly relevant in the past years. With the pandemic, digitalization and the use of ICT by banks and their customers further accelerated and became indispensable. The digital transformation continued unabatedly, even after many containment measures related to the pandemic were relaxed.

The reliance of banks on digital and remote solutions to perform their daily operations, deliver their services to customers, and conduct business has resulted in augmented exposure and vulnerability to increasingly sophisticated cyberattacks and frauds. The scope and relevance of OpRisk further broadened along with technological advances, and underlines the importance of ensuring operational resilience. Moreover, banks are facing increased operational challenges since geopolitical tensions are playing an increasing role in the technological and digital space, with impacts felt across geographies. The Russian war of aggression against Ukraine has led to further heightened cyber risks, including threats to information security and business continuity, while sanctions implemented at an EU and global level in response may give rise to further legal risks.

Based on their answers to the Risk Assessment Questionnaire (RAQ, European Banking Authority, 2022), banks and analysts agree that cyber risk and data security are by far the most prominent OpRisk drivers. Conduct and legal risks are the second most important driver of OpRisk in both banks' and analysts' views, while risk of fraud continues to increase in banks' perceptions.

1.1.4 Regulatory context of OpRisk in the financial sector

In the early days of banking regulation, the focus was primarily on capital adequacy and credit risk, whereas OpRisk was often overlooked. The Basel Committee on Banking Supervision introduced the first international capital standards for banks, known as Basel I (Basel Committee on Banking Supervision, 1988). However, it did not explicitly address OpRisk. Banks were required to hold capital based initially on credit risk and then market risk (Basel Committee on Banking Supervision, 1996), neglecting the broader spectrum of risks, including operational ones. As financial markets evolved and technological advancements increased the complexity of banking operations, incidents of operational failures gained prominence. High-profile events, such as the collapse of Barings Bank in 1995 due to unauthorized trading, highlighted the need to explicitly address OpRisk (Bodur, 2012).

Basel II marked a significant step forward by acknowledging the importance of OpRisk (Basel Committee on Banking Supervision, 2004). It introduced the Advanced Measurement Approach (AMA) as one of the methods for calculating regulatory capital for OpRisk. Banks were required to develop their models for measuring and managing OpRisk, subject to supervisory approval. International financial institutions typically calculate capital requirements for OpRisk via the advanced measurement approach (AMA). Basel II was declined into the European Union jurisdiction through the Capital Requirement Regulations (CRR, European Parliament and Council of the European Union, 2013). The AMA is based on statistical models that are internally defined by institutions and comply with qualitative and quantitative requirements. While the qualitative requirements refer to managerial aspects (*e.g.*, set up an independent OpRisk management function in the financial institution, submit regular reporting to the top management on OpRisk exposures and loss experience), the quantitative ones focus on modeling aspects. In particular, the regulations define which data sources must be used to measure OpRisk:

- internal loss data (*i.e.*, loss data of OpRisk events occurred to the financial institution);
- external loss data (*i.e.*, loss data of relevant OpRisk events occurred to other financial institutions, collected from mass media or specific consortiums, *e.g.*, ORX);
- scenario analysis (*i.e.*, loss data of fictitious OpRisk events that could affect the financial institution with high impact and low probability); and
- business environmental and internal control factors (usually, implemented as key risk indicators, *i.e.*, quantitative measures that are monthly or quarterly observed to monitor the evolution of the exposure to OpRisk, *e.g.*, the time of availability for an IT system, measured in percentage for each month).

Another significant requirement specifies that the OpRisk capital requirement has to be calculated at the 99.9% confidence level, with a holding period of one year. This means that the financial institution may experience an annual loss higher than the capital requirement once every 1000 years, on average. The most-adopted implementation of AMA models is the loss distribution approach (Frachot *et al.* 2001, 2007), where the objective is to estimate the probability distribution of the annual loss amount for OpRisk.

Other possible methods to calculate the OpRisk capital requirement are not based on statistical models but are defined as simple deterministic functions of the "relevant indicator" (RI), which is calculated as the algebraic sum of profit and loss account items of the financial institution (*i.e.*, net interest income, income from shares and other variable/fixed-yield securities, net commissions/fees income, net profit or net loss on financial operations, and other operating income). These methods are:

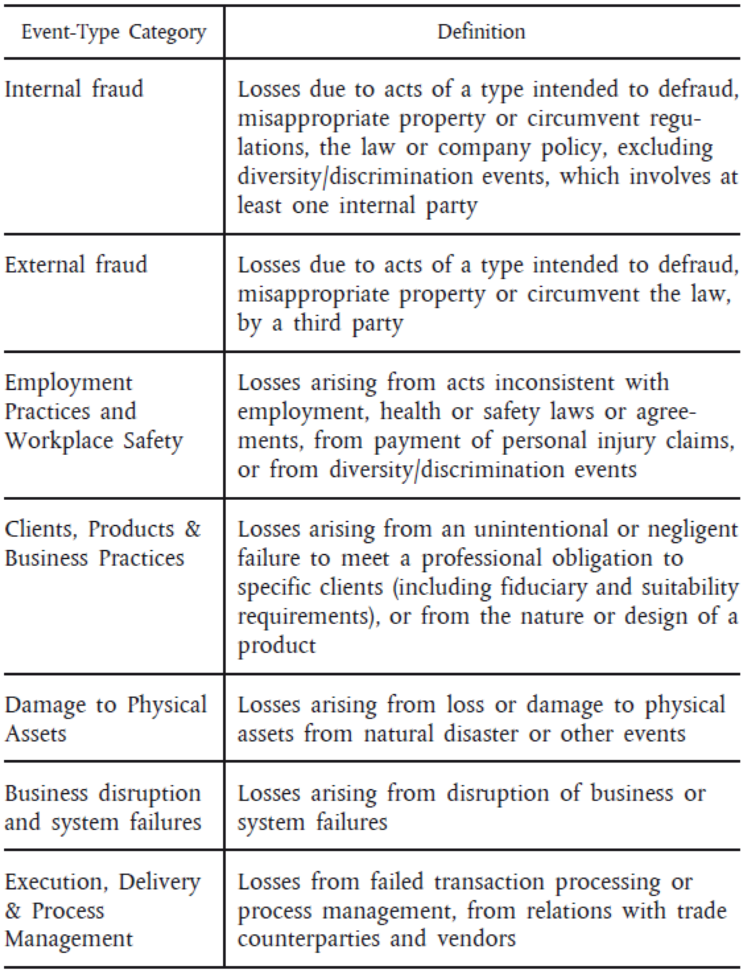
- the Basic Indicator Approach (BIA), where the capital requirement is equal to the 15% of the last 3-year average relevant indicator.

- the Standardized Approach (TSA), which is calculated as BIA, but applying 12-15-18% coefficients to the relevant indicator segments related to different business lines (*i.e.*, 12% for retail banking, retail brokerage, and asset management; 15% for commercial banking, and agency services; 18% for corporate finance, trading and sales, and payment and settlement) based on the riskiness of each activity supposed by regulators.

Considering that BIA and TSA tend to be overly conservative in terms of calculated OpRisk capital requirement, and considering their lack of risk sensitivity (not incorporating any risk driver), the international financial institutions and the significant domestic ones were highly incentivized to adopt the AMA approach. The need to implement AMA models required the financial institutions to employ resources with strong quantitative skills.

Another quantitative requirement asks institutions to be able to map their historical internal loss data into the business lines used for TSA, and into the event types reported in Table 1.1, and to regularly provide these data to competent authorities through the Common Reporting (CoRep) templates.

Table 1.1: Loss event types as described in the CRR.



Event-Type Category	Definition
Internal fraud	Losses due to acts of a type intended to defraud, misappropriate property or circumvent regulations, the law or company policy, excluding diversity/discrimination events, which involves at least one internal party
External fraud	Losses due to acts of a type intended to defraud, misappropriate property or circumvent the law, by a third party
Employment Practices and Workplace Safety	Losses arising from acts inconsistent with employment, health or safety laws or agreements, from payment of personal injury claims, or from diversity/discrimination events
Clients, Products & Business Practices	Losses arising from an unintentional or negligent failure to meet a professional obligation to specific clients (including fiduciary and suitability requirements), or from the nature or design of a product
Damage to Physical Assets	Losses arising from loss or damage to physical assets from natural disaster or other events
Business disruption and system failures	Losses arising from disruption of business or system failures
Execution, Delivery & Process Management	Losses from failed transaction processing or process management, from relations with trade counterparties and vendors

The 2008 financial crisis exposed weaknesses in risk management practices. In response, Basel III was introduced, with a renewed emphasis on strengthening the regulatory framework (Basel Committee on Banking Supervision, 2010). While Basel III primarily focused on addressing issues related to credit and market risk, it continued to recognize the significance of OpRisk. Perceiving the need for a more standardized approach to OpRisk, the Basel Committee, between 2011 and 2016, started reviewing the simpler approaches (Basel Committee on Banking Supervision, 2014), defining the Standardized Measurement Approach (SMA, Basel Committee on Banking Supervision, 2016), and then delivering the new Standardized Approach in the Basel III Final Reform (Basel Committee on Banking Supervision, 2017) also known as Basel IV.

During the same period, the Basel Committee realized that the inherent complexity of the AMA and the lack of comparability, arising from a wide range of internal modelling practices, have exacerbated the variability of capital requirement calculation, thus eroding the confidence in these measures. The Committee has therefore determined the withdrawal of internal modelling approaches and other current methods (AMA, TSA, and BIA) for OpRisk regulatory capital from the Basel Framework, substituting them with the new Standardized Approach. The latter is based on two components:

- Business Indicator Component (BIC), which is based on the Business Indicator (BI). The BI is similar to the Relevant Indicator (used for TSA and BIA), but more conservative since, *e.g.*, instead of net commissions/fees income (*i.e.*, the difference between commissions/fees income and commissions/fees expense), BI considers the maximum between commissions/fees income and commissions/fees expense; and, instead of the other operating income, BI considers the maximum between other operating income and other operating expense. The BIC is obtained from BI applying the coefficients reported in Table 1.2.

  Table 1.2: BI ranges and marginal coefficients.

Bucket	BI range (in €bn)	BI marginal coefficients
1	$\mathrm{BI} \leq 1$	12%
2	$1 < \mathrm{BI} \leq 30$	15%
3	$\mathrm{BI} > 30$	18%

  For example, given a BI = €35bn, then

  $$\mathrm{BIC} = (1 \times 12\%) + (30-1) \times 15\% + (35-30) \times 18\% = €5.37\mathrm{bn}.$$

- Loss Component (LC), which is given by the average annual OpRisk loss, based on the last 10 years' data, multiplied by 15.

The LC is used to calculate the Internal Loss Multiplier (ILM) as follows:

$$\mathrm{ILM} = \log\left(\exp(1) - 1 + \left(\frac{\mathrm{LC}}{\mathrm{BIC}}\right)^{0.8}\right).$$

The ILM is used to smooth the impact of OpRisk losses since the OpRisk capital requirement

(ORC), under the new Standardized Approach, is given by the product between BIC and ILM:

$$\text{ORC} = \text{BIC} \times \text{ILM}.$$

Initially, the new Standardized Approach had to enter into force from January 2022, but for several reasons (*e.g.*, the Covid-19 pandemic crisis, and the need to rule out the Reform in local jurisdictions) this starting date was postponed to January 2025. The Basel III Final Reform left some flexibility for the application of the new Standardized Approach in local jurisdictions. In particular, there is the national discretion to set the ILM equal to 1, and then the ORC equal to BIC. However, the OpRisk loss data, composing the ILM, have still to be collected and reported. According to the final text of the new CRR (European Parliament and Council of the European Union, 2023), to be officially released in the first semester of 2024, this option will be applied in the EU jurisdiction.

In the meantime, the current framework (based on AMA, TSA, and BIA methods) is still adopted and, for this reason, the European Commission decided to supplement their CRR with Regulatory Technical Standards (RTS) of the specification of the assessment methodology under which competent authorities permit institutions to use AMA for OpRisk (European Parliament and Council of the European Union, 2018). This regulation required that financial institutions using AMA adapted their internal modelling approach to show that they are enough robust and conservative. For example, in the case of the Monte Carlo method used to approximate the annual OpRisk loss distributions, it is required to measure the magnitude of the related sample error, where a possible methodology has been proposed by Greselin *et al.* (2019).

1.1.5 OpRisk managerial models

In the last year, to comply with European Parliament and Council of the European Union (2018) (in EU jurisdiction), the quantitative resources, employed in the OpRisk departments of financial institutions, completed their main efforts to improve AMA approaches. Therefore, considering that the new Standardized Approach approach does not require statistical modelling skills, quantitative resources moved part of their focus on the development of statistical models to be used for managerial (*i.e.*, meaning not strictly regulatory) purposes. Carrivick and Westphal (2019) state that the application of advanced analytics, including machine learning and artificial intelligence, will be a core part of any future strategy for the management of operational and non-financial risk. Among the case studies, they cite text mining for data augmentation, where the firm can use Natural Language Processing (NLP) for the tagging of losses to infer root causes from already existing free text descriptions.

Leo *et al.* (2019) report several machine learning applications in the OpRisk management context mainly focused, aside from cyber security cases, on problems related to fraud and suspicious transactions detection:

- Khrestina *et al.* (2017) propose a prototype for the generation of a report that allows for the detection of suspicious transactions. The prototype uses a logistical regression algorithm. They have also included a survey of six software solutions that are currently implemented at various banks for the automation of suspicious transaction detection and monitoring processes, but it is unclear whether these products apply machine learning techniques.

- In money laundering, criminals route money through various transactions, layering them with legitimate transactions to conceal the true source of the funds. The funds typically originate from criminal or illegal activities and can be further used in other illegal activities including the financing of terrorism. There has been extensive research on detecting financial crimes using traditional statistical methods, and more recently, using machine learning techniques. Clustering algorithms identify customers with similar behavioral patterns and can help to find groups of people working together to commit money laundering (Sudjianto *et al.*, 2010).

- A major challenge for banks, given the large volume of transactions per day and the non-uniform nature of many, is to be able to sort through all the transactions and identify those that are suspicious. Financial institutions utilize anti-money laundering systems to filter and classify transactions based on degrees of suspiciousness. Structured processes and intelligent systems are required to enable the detection of these money laundering transactions (Kannan and Somasundaram, 2017).

- Credit card fraud is significantly increasing annually, costing consumers and the industry billions of dollars. To manage the increasing fraud risk and minimize losses, banks have fraud detection systems in place. The systems are oriented towards increasing the detection rate while minimizing the false positive rate. Models are estimated based on samples of fraudulent and legitimate transactions in supervised detection methods, while in unsupervised detection methods, outliers or unusual transactions are identified as potential cases of fraud. Some reported challenges in credit card fraud detection are the non-availability of real data sets, unbalanced data sets, the size of the data sets, and the dynamic behavior of fraudsters. Bayesian algorithms, K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Bagging ensemble classifiers have been varyingly used in fraud detection systems. A comparative evaluation showed that the Bagging ensemble classifier based on decision tree algorithms works well, as it is independent of attribute values, and is also able to handle class imbalance (Zareapoor and Shamsolmoali, 2015).

- False alarms, namely transactions labeled as fraudulent being instead legitimate, are significant, causing concerns for customers and delaying the detection of actual fraudulent transactions. Large Canadian banks rely heavily on NN scores, ranging from 1 to 999, with 1 being the lowest chance of a fraudulent transaction, determined by neural network algorithms. Reportedly, 20% of transactions with a NN score greater than or equal to 990 are fraudulent, causing fraud analysts to inefficiently spend time investigating legitimate transactions. A meta-classifier (a multiple algorithm learning technique) applied to a post-neural network was shown to provide quantifiable savings improvements with a larger percentage of fraudulent transactions being caught (Pun and Lawryshyn, 2012).

- There are a few papers on fraud risk detection in credit cards and online banking. They concern credit card fraud detection in domains not specifically related to bank risk management or the banking industry. One would note that the algorithms they refer to were SVM, KNN, Naïve Bayes Classifier, and Bagging ensemble classifier based on a decision tree (Dal Pozzolo and Bontempi, 2015; Vaidya and Mohod, 2014).

- Sharma and Choudhury (2021) used unsupervised learning approaches, such as self-organizing maps (SOM), to detect fraudulent acts in areas like credit cards, money laundering, and financial statements.

More in general, a systematic review of the role of data analytics within OpRisk management, referred to financial services and energy sectors, can be found in Cornwell *et al.* (2023).

1.2 Text analysis

The explosive growth of digital content in recent years has led to an unprecedented volume of unstructured data. Text analysis, also known as text mining or Natural Language Processing (NLP), has emerged as a powerful set of techniques to transform unstructured textual data into structured and actionable information. As we navigate through the digital age, the ability to analyze and derive insights from textual data becomes increasingly vital for businesses, researchers, and policymakers.

Text analysis is instrumental in extracting valuable information from various sources, including social media, news articles, academic papers, and more. Businesses, researchers, and policymakers can leverage text analysis to gain a competitive edge, make informed decisions, and monitor trends. The significance of text analysis lies in its ability to convert large volumes of unstructured textual data into structured information, providing a basis for data-driven decision-making.

At its core, text analysis involves the use of computational techniques to process, analyze, and interpret textual data. This includes tasks such as text preprocessing, sentiment analysis, named entity recognition, and topic modeling. By breaking down textual content into meaningful components, text analysis enables the extraction of patterns, trends, and insights that may not be apparent through manual examination.

1.2.1 Text cleaning

Text cleaning is a crucial step in the text analysis pipeline. It involves removing noise and irrelevant information from the text, such as HTML tags, special characters, digits, and punctuation. Text cleaning ensures that the subsequent analysis is based on meaningful content.

Stop-words are common words that are often removed during text analysis because they do not carry significant meaning. Examples include “the”, “and”, “is”, etc. Removing stop-words helps focus the analysis on more meaningful terms. Several natural language processing libraries provide predefined stop-word lists for multiple languages.

N-grams are contiguous sequences of n words from a given sample of text or speech. They are used to capture the local structure and context of language. For example, bigrams (2-grams) represent pairs of consecutive words, and trigrams (3-grams) represent triplets of consecutive words.

1.2.2 Methodologies in text analysis

Text analysis encompasses a variety of methodologies, ranging from traditional statistical approaches to modern machine learning techniques. The choice of methodology depends on the specific goals of the analysis and the nature of the textual data.

Statistical methods, such as frequency analysis, are fundamental in text analysis. These techniques involve counting the occurrences of words or phrases to identify patterns and trends. While simple, statistical approaches can provide valuable insights, especially when dealing with large data sets (Aggarwal, 2012).

Machine learning plays a pivotal role in text analysis, offering advanced methods for tasks like sentiment analysis, classification, and clustering. Supervised learning models, such as support vector machines and neural networks, can be trained on labeled data sets, while unsupervised learning models, like k-means clustering, can discover patterns without predefined categories (Jurafsky and Martin, 2019).

The bag-of-words model is a simple yet powerful representation of text. It represents a document as an unordered set of words, ignoring grammar and word order but keeping track of word frequency. Each word becomes a feature, and the document is represented as a vector of word frequencies, leading to the representation of a document-by-term matrix. This model forms the foundation for many text analysis tasks, including document classification and clustering.

TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It takes into account both the frequency of the word in the document (Term Frequency) and the rarity of the word in the corpus (Inverse Document Frequency). The resulting TF-IDF score helps identify words that are significant to a specific document but not common across the entire corpus. The TF-IDF score for a term t in a document d is calculated as follows:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

where $\text{TF}(t,d)$ is the Term Frequency of term t in document d, and $\text{IDF}(t)$ is the Inverse Document Frequency of term t across the entire corpus.

The idea behind IDF is to assign higher weights to terms that are less common in the entire corpus, indicating their potential significance. The IDF of a term t is calculated using the formula:

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right)$$

where N is the total number of documents in the corpus, and $\text{DF}(t)$ is the document frequency of term t, representing the number of documents in the corpus that contain term t.

The use of the logarithm in the IDF formula helps mitigate the impact of extremely common terms by downscaling their IDF values. This ensures that terms with lower document frequency receive higher IDF scores, indicating their uniqueness and potential importance.

Cosine similarity is a metric used to measure the similarity between two vectors. In the context of text analysis, such vectors often are defined as the TF or the TF-IDF representations of documents. Cosine similarity calculates the cosine of the angle between the vectors, providing a measure that ranges from 0 (completely dissimilar) to 1 (identical). This metric is commonly used in document similarity analysis, clustering, and information retrieval. The cosine similarity between two vectors A and B is calculated as follows:

$$\text{Cosine Similarity}(A,B) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

where $A \cdot B$ is the dot product of vectors A and B, and $||A||$ and $||B||$ are the Euclidean norms of vectors A and B, respectively.

1.2.3 Applications of text analysis

Text analysis finds applications across diverse domains, revolutionizing the way organizations and researchers interact with textual data.

In the business realm, text analysis aids in market research, customer feedback analysis, and competitive intelligence. By analyzing customer reviews, social media mentions, and news articles, businesses can make data-driven decisions to improve products, services, and overall customer satisfaction (Feldman and Sanger, 2006).

In healthcare, text analysis facilitates the extraction of valuable information from medical records, research papers, and clinical notes. This can improve patient outcomes, enable early disease detection, and support medical research by identifying patterns in large volumes of biomedical literature (Friedman *et al.*, 1998).

Social media platforms generate vast amounts of textual data, providing insights into public opinion, trends, and sentiment. Text analysis on social media content helps marketers, policymakers, and researchers to understand the public sentiment, track emerging issues, and engage with their audience effectively (Gupta and Gupta, 2018).

1.3 Word embedding

The representation of words in a way that captures their semantic meaning and relationships is a fundamental challenge in NLP and machine learning. Traditional approaches often relied on sparse and high-dimensional representations, such as a document-by-term matrix, where each word is represented by a vector with a size equal to the vocabulary. While simple, these representations cannot capture semantic similarities and relationships between words.

Word embeddings, on the other hand, provide a dense and continuous representation of words in a lower-dimensional space, where the geometric distances between vectors reflect semantic relationships. This paradigm shift has significantly improved the performance of NLP models, enabling them to better understand context, relationships, and nuances in language.

The motivation behind word embeddings lies in the limitations of traditional representations and the desire to capture the meaning of words in a more nuanced manner. Sparse representations, such as bag-of-words, treat each word as an isolated entity without considering its context or semantic connections to other words. In contrast, word embeddings aim to embed words in a continuous vector space, preserving semantic relationships and contextual information.

Word embeddings have proven to be highly effective in various NLP tasks, including machine translation, sentiment analysis, and named entity recognition. By representing words in a continuous space, embeddings enable models to generalize better and capture subtle nuances in language, leading to improved performance on a wide range of tasks.

1.3.1 Word embedding models

Several word embedding models have been developed, each with its own approach and strengths. In this section, we will explore some of the prominent models, namely `Word2Vec`, `GloVe`, `BERT`, and `fastText`.

Word2Vec

`Word2Vec`, developed by Mikolov *et al.* (2013), is a widely used word embedding model that aims to learn continuous vector representations for words by predicting the context in which words occur. It introduces two architectures, Continuous Bag of Words (CBOW) and Skip-gram, both utilizing shallow neural networks to learn word embeddings:

- The CBOW model predicts the target word given its context. It takes the context words as input and predicts the target word in the center. The training objective is to maximize the likelihood of predicting the target word.

- The Skip-gram model, in contrast, predicts the context words (surrounding words) given the target word. It takes a single word as input and predicts the context words within a certain window. Like CBOW, the training objective is to maximize the likelihood of predicting the context words.

`Word2Vec` has demonstrated excellent performance on various NLP tasks and is known for its efficiency in learning high-quality word embeddings from large corpora.

GloVe (Global Vectors for word representation)

`GloVe`, introduced by Pennington *et al.* (2014), is a word embedding model that focuses on capturing global relationships between words by considering the entire corpus during training. It constructs a global word-word co-occurrence matrix and then factorizes it to obtain word embeddings.

The training objective of `GloVe` is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. `GloVe` embeddings are known for their ability to capture semantic relationships and exhibit excellent performance on word analogy tasks.

BERT (Bidirectional Encoder Representations from Transformers)

`BERT`, introduced by Devlin *et al.* (2019), represents a breakthrough in word embeddings by leveraging bidirectional context. Unlike traditional models that consider only left or right context, `BERT` utilizes a transformer architecture to capture contextual information from both directions.

`BERT` is pre-trained on large amounts of text data in an unsupervised manner and has achieved state-of-the-art results on various NLP benchmarks. It has become a cornerstone for many downstream NLP tasks, serving as a feature extractor or fine-tuning base. Training `BERT` requires a significant computational effort. For this reason, an efficient strategy can be to start from pre-trained `BERT`, and then fine-tune it using a subject-specific data set.

FastText

`FastText`, proposed by Bojanowski *et al.* (2017), extends traditional word embeddings by considering subword information. Instead of representing words as a whole, `fastText` breaks them down into smaller subword units called *n*-grams. This allows `fastText` to capture morphological information and handle out-of-vocabulary words.

`FastText` has proven effective for languages with rich morphology and performs well on tasks such as text classification and language modeling.

1.3.2 Applications of word embeddings

Word embeddings have found widespread applications across various NLP tasks, revolutionizing the way machines understand and process human language. Some notable applications include:

- Machine Translation: word embeddings improve the quality of machine translation by capturing semantic relationships between words in different languages.
- Sentiment Analysis: models trained on word embeddings exhibit better sentiment analysis performance by understanding the contextual meaning of words in a sentence.
- Named Entity Recognition (NER): word embeddings assist in recognizing named entities by providing richer semantic information about words in context.

- Text Summarization: embeddings help models understand the importance of words in a document, facilitating more accurate text summarization.

The ability of word embeddings to capture contextual and semantic information makes them indispensable for a wide range of NLP applications.

1.3.3 Challenges and future directions

Despite their success, word embeddings are not without challenges. One significant challenge is the handling of polysemy, *i.e.*, words with multiple meanings. Additionally, embeddings may capture biases present in training data, leading to biased representations.

Future research directions in word embeddings include addressing these biases, improving the interpretability of embeddings, and exploring methods to handle rare or unseen words more effectively. Additionally, research efforts continue to push the boundaries of pre-training models like `BERT` and optimize their efficiency for various downstream tasks.

1.4 Uniform Manifold Approximation and Projection (UMAP)

In the era of big data, the need to understand and interpret high-dimensional data sets is pervasive across various fields, including machine learning, biology, and data visualization. Dimensionality reduction techniques play a crucial role in simplifying complex data while preserving its inherent structure. Uniform Manifold Approximation and Projection (UMAP), introduced by McInnes *et al.* (2018), is a relatively recent addition to the suite of dimensionality reduction methods that has gained attention for its ability to address some limitations of traditional techniques.

The motivation behind dimensionality reduction techniques stems from the curse of dimensionality, a phenomenon where high-dimensional data become sparse, and distances between points become less meaningful. Traditional methods like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) have been widely used, but each has its limitations. PCA is linear and may not capture non-linear structures, while t-SNE can struggle with preserving global structure and is computationally expensive.

UMAP is motivated by the desire to overcome these limitations. It seeks to provide a flexible and efficient approach to dimensionality reduction that can handle both local and global structure in the data.

1.4.1 Mathematical foundations of UMAP

The mathematical foundations of UMAP delve into the complexity of constructing a low-dimensional representation that faithfully captures both local and global structures in high-dimensional data. At its core, UMAP combines topological and metric considerations to create a mapping that preserves pairwise similarities between data points.

Let X represent the high-dimensional data, and Y the corresponding low-dimensional representation. UMAP aims to learn a mapping $f : X \to Y$ such that the pairwise similarities between data points are maintained. The optimization problem involves minimizing a cost function that quantifies the discrepancy between the fuzzy topological structure in the high-dimensional space and the low-dimensional space.

Topological Considerations

UMAP introduces a fuzzy topological structure to capture relationships between data points. It defines a set of neighborhoods for each data point in both the high-dimensional and low-dimensional spaces. The notion of neighborhood is essential for preserving the local structure of the data.

In the high-dimensional space, the fuzzy set F_i^X represents the neighborhood of data point x_i. Similarly, in the low-dimensional space, the fuzzy set F_i^Y corresponds to the neighborhood of its counterpart y_i. The goal is to ensure that the relationships between neighborhoods in both spaces are maintained.

Metric Considerations

UMAP leverages metric considerations to align the pairwise similarities between data points in the high-dimensional and low-dimensional spaces. The focus is on optimizing the low-dimensional representation to minimize the mismatch between the fuzzy topological structures.

The cross-entropy between the fuzzy sets F_i^X and F_i^Y quantifies this mismatch. The cross-entropy is defined as:

$$C_i = -\sum_j p_{i,j}^X \log(q_{i,j}^Y)$$

where $p_{i,j}^X$ and $q_{i,j}^Y$ are probabilities associated with the pairwise similarities in the high-dimensional and low-dimensional spaces, respectively. The probabilities are computed based on the distances between data points in their respective spaces.

Optimization Procedure

The optimization procedure involves adjusting the low-dimensional representation Y iteratively to minimize the cross-entropy. This is typically achieved using stochastic gradient descent methods, where the gradients are computed with respect to the embedding coordinates y_i.

The embedding coordinates are updated based on the negative gradient of the cross-entropy:

$$y_{i,new} = y_{i,old} - \eta \frac{\partial C_i}{\partial y_i}$$

Here, η represents the learning rate, controlling the step size in the optimization process.

Global Structure Preservation

UMAP's unique contribution lies in its ability to balance local and global structure preservation. While many dimensionality reduction techniques focus solely on local relationships, UMAP's cost function incorporates global considerations. This ensures that the resulting low-dimensional representation captures not only the fine-grained details but also the broader patterns and structures in the data.

Embedding Stability

UMAP introduces the concept of "embedding stability" to address the sensitivity of the algorithm to hyperparameters and initialization. Embedding stability assesses the consistency of the embedding across multiple runs with slightly perturbed inputs. A stable embedding is more likely to be a reliable representation of the underlying data structure (McInnes *et al.*, 2018).

1.4.2 Comparison with Other Dimensionality Reduction Techniques

To appreciate the strengths and weaknesses of UMAP, it is useful to compare it with other widely used dimensionality reduction techniques, such as PCA and t-SNE:

- Linear vs. Non-linear: PCA is a linear dimensionality reduction technique that identifies the axes along which the data varies the most. While PCA is efficient and well-suited for linear structures, it may not capture non-linear relationships in the data. UMAP, in contrast, is designed to handle non-linear structures, making it a more flexible choice for complex data sets.

- Local vs. Global Structure: t-SNE is known for its ability to capture local structures in the data. However, it tends to struggle with preserving global structures, and its computational cost can be prohibitive for large data sets. UMAP addresses these issues by combining global and local considerations, providing a more balanced approach to dimensionality reduction.

In addition to its algorithmic advantages, UMAP is often praised for its computational efficiency. The scalability of UMAP makes it applicable to large data sets, which can be challenging for t-SNE. This practical consideration contributes to the popularity of UMAP in real-world scenarios where processing large amounts of data is common.

1.4.3 Limitations of UMAP and best practices

While UMAP offers numerous advantages for diverse applications, it is essential to consider certain factors and follow best practices:

- Data preprocessing: appropriate data preprocessing is crucial for the success of UMAP. Standardizing or normalizing input features before applying UMAP ensures that variables with different scales do not disproportionately influence the results. Additionally, addressing missing data and handling outliers contributes to the robustness of the algorithm (McInnes *et al.*, 2018).

- Parameter sensitivity: UMAP comes with several hyperparameters, such as the number of neighbors, minimum distance, and metric choices. Sensitivity to parameter settings is common, and users should experiment with different configurations to find the most suitable values for the specific data set and objectives. Visualizing the UMAP representations with varying parameters can offer valuable insights into the stability and robustness of the results.

- Interpretability: while UMAP excels in capturing complex relationships, interpreting the exact meaning of dimensions in low-dimensional space can be challenging. Users should approach UMAP as a tool for visualization and feature extraction rather than as a black-box model. It is crucial to complement UMAP results with domain knowledge and an understanding of the specific context in which it is applied.

1.4.4 Applications of UMAP

The versatility of UMAP has led to its adoption across various domains, providing valuable insights and solutions to a wide array of data analysis challenges. In this section, we explore some notable applications of UMAP and its impact on different fields.

Bioinformatics and single-cell RNA sequencing

UMAP has found significant applications in the field of bioinformatics, particularly in the analysis of single-cell RNA sequencing (scRNA-seq) data. The complexity of scRNA-seq data sets, with numerous genes and cells, makes them ideal candidates for dimensionality reduction techniques. UMAP's ability to capture both local and global structures allows researchers to visualize and interpret the intricate transcriptional landscapes of individual cells (Becht *et al.*, 2019). Its effectiveness in identifying cell types, states, and transitions within heterogeneous cell populations has contributed to advancing our understanding of cellular diversity and dynamics.

Machine learning and feature extraction

In the context of machine learning, UMAP can serve as a powerful tool for feature extraction. High-dimensional data sets often pose challenges in terms of redundant or irrelevant features. UMAP can be employed to distill the essential information, resulting in a low-dimensional representation that enhances model performance. The application of UMAP as a preprocessing step improves model generalization, particularly in scenarios with high-dimensional data prone to overfitting. Supervised UMAP, where the algorithm is trained on labeled data, further enhances its utility for classification tasks (McInnes *et al.*, 2018).

Clustering and anomaly detection

The low-dimensional representations generated by UMAP are well-suited for clustering and anomaly detection tasks. UMAP can reveal the underlying structure in the data, making it easier to identify distinct groups or outliers. Clustering algorithms applied to UMAP-transformed data can unveil natural partitions, while anomaly detection models can benefit from the clear separation of normal and anomalous instances. This application extends UMAP's utility beyond visualization to tasks that require the identification of patterns and anomalies in the data (Ramos *et al.*, 2020).

Neural network embeddings

In the field of deep learning, UMAP has found applications as an embedding layer within neural network architectures. The low-dimensional embeddings generated by UMAP can serve as informative inputs to subsequent layers in a neural network. This approach leverages UMAP's ability to capture meaningful representations and has been shown to enhance the performance of neural networks, particularly in tasks with complex, non-linear relationships. UMAP embeddings can contribute to improved training efficiency and model interpretability in deep learning applications (McInnes *et al.*, 2018).

1.5 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA), introduced by Blei *et al.* (2003), is a widely used generative probabilistic model for topic modeling, a crucial task in natural language processing and information retrieval. LDA has emerged as a prominent technique for unsupervised topic modeling. This powerful probabilistic model provides a framework for discovering hidden thematic structures within a collection of documents. LDA has found applications in several fields such as information retrieval, social media analysis, and content recommendation.

The motivation behind LDA arises from the challenge of making sense of vast amounts of unstructured text data. As the volume of textual information on the internet continues to grow exponentially, the ability to automatically identify and categorize topics within documents becomes crucial. LDA offers a principled and scalable approach to discover the latent thematic structures that govern the generation of documents.

LDA is one of the main methodologies for topic modeling. Topics, in the context of LDA, represent latent thematic patterns that are assumed to generate the observed documents. Each document is viewed as a mixture of topics, and each topic is characterized by a distribution over words. The intuition is that documents exhibit multiple topics, and each topic contributes to the generation of words in a document with a certain probability.

1.5.1 Generative process of LDA

The generative process of LDA provides a conceptual framework for understanding how documents are probabilistically generated. It involves a series of steps that mimic the way an author might compose a document. The generative process assumes that each document in the corpus is created through the following steps:

1. For each document: Choose a distribution over topics.
2. For each word in the document:
    (a) Choose a topic from the distribution over topics.
    (b) Choose a word from the topic's distribution over words.

These assumptions capture the idea that documents are mixtures of topics, and each topic is a distribution over words. The generative process reflects the inherent variability in document composition and the diversity of topics covered in a corpus.

The generative process can be mathematically formalized using probabilistic graphical models. Let's denote the observed variables as W (words), and the hidden or latent variables as Z (topics), θ (document-topic distribution), and β (topic-word distribution). The joint distribution of the observed and latent variables can be expressed as:

$$P(W,Z,\theta,\beta|\alpha,\eta) = \prod_{d=1}^{D} \left(P(\theta_d|\alpha) \prod_{n=1}^{N_d} P(w_{d,n}|\beta_{z_{d,n}},\eta) \right).$$

Here, D is the number of documents, N_d is the number of words in document d, α and η are hyperparameters, and $\beta_{z_{d,n}}$ represents the distribution over words for topic $z_{d,n}$.

LDA uses the Markov chain Monte Carlo (MCMC) to decode the generative process. Specifically, given a set of documents and a previously defined number of topics K, MCMC estimates the distributions corresponding to each topic as well as the mixture probabilities for any document on topics. In Figure 1.1, the dark node W represents words, which is the only observed variable in the model. The light node Z represents the topic, which hides inside the document.

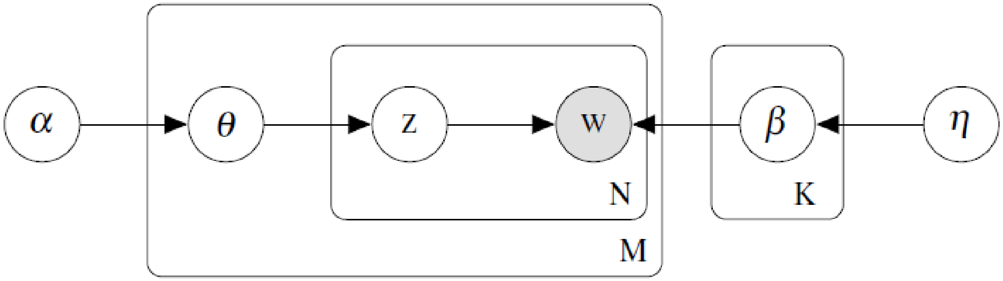


Figure 1.1: Graphical representation of LDA.

The topic distribution under each document is a Multinomial distribution $Mult(\theta)$ with its Dirichlet distribution conjugate prior $Dir(\alpha)$. The word distribution under each topic is a Multinomial distribution $Mult(\beta)$ with its conjugate prior $Dir(\eta)$. To generate the n^{th} word in the certain document, first, we select a topic z from document-topic distribution $Mult(\theta)$, then we select a word under this topic $w|z$ from topic-word distribution $Mult(\beta)$. This is the generative process:

1. Draw $\theta_m \sim Dir(\alpha)$
2. For each topic $k \in \{1,\ldots,K\}$
    - Draw $\beta_k \sim Dir(\eta)$
3. For each word w_n in document $m, n \in \{1,\ldots,N\}$
    - Draw topic $z_n \sim Mult(\theta_m)$
    - Draw word $w_n|z_n \sim Mult(\beta_k)$

An interesting problem concerns the definition of the correct number of topics. Several strategies can be used to solve that problem. They can be based on assessing goodness-of-fit through already noted measures, such as perplexity. Otherwise, the assessment is done through a subjective evaluation of the researchers by visualizing the plotted clustering results or checking the highest probability words of the topics.

1.5.2 Challenges and considerations

While LDA has proven to be a powerful tool for topic modeling, it is not without challenges and needs some further considerations:

- Model complexity: the assumption of a fixed number of topics in LDA can be limiting in practice. Real-world corpora may exhibit dynamic and evolving themes that are not well-captured by a static topic model. Extensions such as Dynamic Topic Models (DTM) have been proposed to address this limitation (Blei and Lafferty, 2007).

- Interpretability: although LDA provides a principled way to discover topics, interpreting these topics can be subjective. Assigning human-interpretable labels to topics is a challenging task, and the quality of topic interpretation may vary based on the data set and the number of topics (Chang *et al.*, 2009).

- Computational efficiency: inference in LDA can be computationally demanding, especially for large data sets. Variational Inference and Gibbs Sampling, while effective, may require significant computational resources. Approximate methods and parallelization techniques are often employed to enhance efficiency (Newman *et al.*, 2010).

1.5.3 Applications of LDA

LDA has found applications across a spectrum of domains, showcasing its versatility in uncovering latent thematic structures. There are notable applications of LDA in different fields:

- Text classification and document retrieval: LDA's ability to capture the underlying topics in a document makes it valuable for tasks such as text classification and document retrieval. By representing documents as distributions over topics, LDA embeddings can be used to measure document similarity and enhance the performance of information retrieval systems (Blei and Lafferty, 2009).

- Social media analysis: the abundance of user-generated content on social media platforms presents a rich source of data for analysis. LDA has been employed to extract topics from social media posts, enabling insights into trending discussions, sentiment analysis, and identification of influential themes (Hong and Davison, 2010).

- Content recommendation: in content recommendation systems, understanding the latent topics within user preferences is crucial. LDA has been utilized to model user preferences based on their interactions with content. By identifying topics associated with users, personalized recommendations can be generated (Wang *et al.*, 2011).

- Biomedical informatics: in the biomedical domain, LDA has been applied to analyze large collections of scientific literature and identify key topics within research articles. This facilitates literature review, trend analysis, and knowledge discovery in fields such as genomics and clinical research (Cohen *et al.*, 2004).

- Market research and customer feedback: LDA finds applications in market research by analyzing customer feedback, reviews, and survey responses. By extracting topics from textual data, businesses can gain insights into customer preferences, identify areas for improvement, and tailor products or services accordingly (Griffiths and Steyvers, 2004).

Chapter 2

A text analysis of Operational Risk loss descriptions

Based on:

Di Vincenzo D., Greselin F., Piacenza F., & Zitikis R. (2023).
A text analysis of operational risk loss descriptions.
Journal of Operational Risk, 18(3), 63–90.
https://doi.org/10.21314/JOP.2023.003

Di Vincenzo D., Greselin F., Piacenza F., & Zitikis R. (2022).
A text analysis of operational risk loss descriptions.
Tenth International Hybrid Conference on MATHEMATICAL AND STATISTICAL METHODS FOR ACTUARIAL SCIENCES AND FINANCE - MAF2022 - Book of Abstracts, 80–80.
https://drive.google.com/file/d/1ZHWO4CnXp1U4Mw6u5RlRGzaXCIzIC6Ze/view

2.1 Introduction

The operational risk (or OpRisk) is defined as the risk of loss resulting from inadequate or failed internal processes, people and systems, or from external events, and also includes the legal risk (European Parliament and Council of the European Union, 2013). International financial institutions typically manage this risk inside specific OpRisk management functions, which perform the activities prescribed by the regulations, such as:

- Data collection (*e.g.*, recording loss data, performing scenario analyses and tracking risk indicators)
- Capital requirement calculations using Advanced Measurement Approach (AMA) internal models
- Reporting of loss data

To perform the above-mentioned activities, financial institutions have to define and implement databases to collect and store the necessary information. In the case of loss events due to OpRisk, at least the following attributes are collected:

- Loss amounts
- Dates (occurrence, discovery and accounting)
- Affected organizational units
- Basel loss event types (Internal Fraud; External Fraud; Employment Practices and Workplace Safety; Clients, Products & Business Practices; Damage to Physical Assets; Business Disruption and System Failures; Execution, Delivery & Process Management)
- Event descriptions

The OpRisk databases contain the above-mentioned structured data, which are used for regulatory activities. However, during the last years, the OpRisk functions have been increasingly required to move beyond their regulatory tasks, providing a more effective contribution in order to pro-actively manage the risk, and prevent or mitigate its impact. This development gives new importance to OpRisk databases, and in particular to OpRisk event descriptions, which are usually defined as free text fields. The possibility to make all the information in these databases, including event descriptions, more fully available to the OpRisk analysts represents a valuable opportunity to improve the knowledge about loss events and to design the most adequate mitigation strategies.

The present work is among the first ones that have addressed the application of text analysis techniques to the OpRisk event descriptions. Text analysis, together with speech recognition and

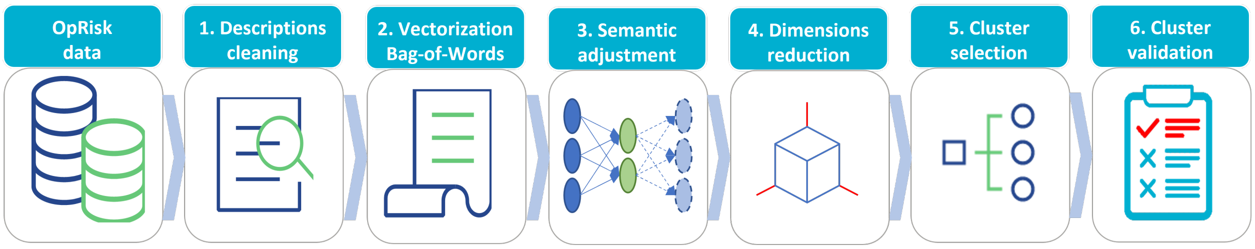


Figure 2.1: Workflow for OpRisk descriptions analysis.

automatic translation, is one of the main tasks of Natural Language Processing (NLP), which is a branch of Artificial Intelligence (AI). In particular, to the best of our knowledge, for the first time in literature, the present work defines a general structured workflow that can be applied to the OpRisk descriptions to analyze them for several purposes. This overarching framework complements the one already applied to quantitative data.

The proposed workflow includes the following steps (as represented in Figure 2.1):

1. Description cleaning (*e.g.*, splitting of different languages, removing stop-words, reducing words to their lemmas).

2. Text vectorization (building a document-by-term matrix, where each element is properly weighted).

3. Semantic adjustment (enriching the document-by-term matrix, considering the semantic similarity among words).

4. Dimensionality reduction (building a 2D representation of the data, where each point is an event description, and similar ones are represented as clusters of points).

5. Cluster selection (tagging of points within each cluster by OpRisk analysts).

6. Cluster validation (application of clustering and topic modelling techniques to validate and support the clustering performed by the analysts).

The remainder of the chapter is structured as follows. Section 2.2 gives a literature review of text analysis applied to OpRisk. Section 2.3 describes in detail the steps of the proposed workflow. Section 2.4 reports an application of the proposed workflow to descriptions of the Common

Reporting (CoRep) OpRisk data set of the UniCredit banking group (all data elaborations and analyses in this section are performed using software R (R Core Team, 2023)). Finally, Section 2.5 summarizes the main achievements and results of this work, discussing also possible extensions in several directions.

2.2 Literature review

For the frameworks and methods of analysis applied to quantitative OpRisk data and the related challenges, we refer the reader to, for example, the work of Soprano *et al.* (2010), Cope *et al.* (2009), Lambrigger *et al.* (2007), Shevchenko and Wüthrich (2006), Danesi *et al.* (2016), and Bazzarello *et al.* (2006).

Turning to qualitative data, the existing literature proposes only a few solutions for the analysis of textual data related to OpRisk loss event descriptions.

Pakhchanyan *et al.* (2022) apply machine learning techniques to OpRisk descriptions in order to automatically classify events into Basel event types. Note that, while they adopt supervised methods to classify OpRisk events into pre-defined regulatory categories, they do not propose solutions to identify new managerial (more granular) clusters that can be used to understand the root causes of the underlying risks. The classification of OpRisk events is also discussed by Zhou *et al.* (2021), who propose semi-supervised methods in order to include unlabeled data in the training stage.

Wang *et al.* (2018, 2022) investigate the main OpRisk factors by applying the Latent Dirichlet Allocation (LDA), but without reporting many details on the applied descriptions cleaning and text vectorization.

Data Study Group team (2019) provide a preliminary proof-of-concept for the potential usefulness of statistical and NLP approaches in OpRisk modelling, applying LDA and long short-term memory neural networks (LSTM).

Carrivick and Westphal (2019) suggest that text analysis methodologies can be useful to gain deeper insights into the OpRisk data, although without proposing detailed approaches.

A recent literature review on the application of text analysis in the financial sector (Bach *et al.*, 2019) reveals that the main research focus is on stocks price prediction, financial fraud detection and market forecast. In the literature, there are several proposals to manage fraud risk (which is a part of OpRisk) by making use of text analysis. For example, Holton (2009) proposes a methodology to detect financial frauds by identifying and classifying emails with disgruntled communications.

2.3 Workflow for OpRisk descriptions analysis

2.3.1 Descriptions cleaning

Descriptions are prepared for analysis using some cleaning procedures. The set of all descriptions (or documents) to be analyzed is called "corpus". Procedures to clean texts include the following ones:

- Data anonymization: applying routines to retrieve and delete (or substitute with conventional tags) any personal information and dates from texts, for compliance with GDPR (European Parliament and Council of the European Union, 2016) and for analytical purposes (Francopoulo and Schaub, 2020).
- Splitting of different languages: applying routines to recognize and separate parts of text written in different languages (Jauhiainen *et al.*, 2019).
- Ignoring cases, which can be done by case-folding each letter into lowercase.
- Removing punctuations and digits.
- Removing frequent words that do not contain much information (also called stop-words), such as articles, pronouns, conjunctions, and words like "of", "about", "that", etc. Lists of stop-words are readily available for the main languages. In particular, for the application described in Section 2.4, the stop-word list, related to the English language, has been derived from meta::cpan (2021).
- Using regular expressions to detect special characters (*e.g.*, "ù", "ä", "|", etc.) and remove them.

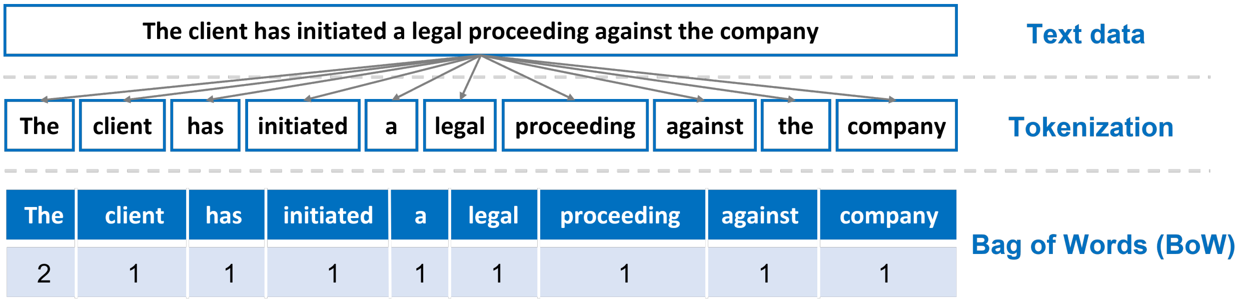
- Reducing words to their lemmas (*e.g.*, “pay” from “paying”, “client” from “clients”), substituting each word with the corresponding canonical form. The lemmatization reduces the number of distinct words in a text corpus and increases the frequency of occurrence for some of them.

2.3.2 Text vectorization with the Bag-of-Words approach

In the Bag-of-Words (BoW) approach (Harris, 1954), the data set is transformed into a matrix, where:

- row i represents the i-th document, d_i;
- column j represents the j-th term, w_j, of the transformed data set; and
- in cell (i, j) of the document-by-term matrix, we store the Term Frequency (TF), $\mathrm{TF}(d_i, w_j)$, of the term w_j in the document d_i (*i.e.*, the number of times w_j appears in d_i) (Luhn, 1957).

An example of BoW representation is given in Figure 2.2.



The	client	has	initiated	a	legal	proceeding	against	company
2	1	1	1	1	1	1	1	1

Figure 2.2: Example of Bag-of-Words representation (without stop-words removal).

Another common approach to text vectorization in text analysis is known as “Term Frequency – Inverse Document Frequency” (TF-IDF) method. The Inverse Document Frequency is a scoring of how rare a word is across documents (Spärck Jones, 1972):

$$\mathrm{IDF}(w_j) = \log \frac{n}{n_j},$$

where n is the number of documents in the corpus, and n_j is the number of documents in which word w_j appears. In the TF-IDF method (Bollacker *et al.*, 1998), the value of word w_j in document

d_i is given by

$$\text{TF-IDF}(d_i, w_j) = \text{TF}(d_i, w_j) \times \text{IDF}(w_j),$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, and m is the dictionary size.

The similarity between documents can be calculated using the "cosine similarity" (Singhal, 2001). Considering the documents d_s and d_t ($s, t \in \{1, \ldots, n\}$), represented by

$$\boldsymbol{x}_s = (\text{TF-IDF}(w_1, d_s), \ldots, \text{TF-IDF}(w_m, d_s)),$$

$$\boldsymbol{x}_t = (\text{TF-IDF}(w_1, d_t), \ldots, \text{TF-IDF}(w_m, d_t)),$$

their cosine similarity is given by the cosine of the angle between the two vectors representing the two descriptions:

$$\text{CS}(d_s, d_t) = \frac{\boldsymbol{x}_s \cdot \boldsymbol{x}_t}{\|\boldsymbol{x}_s\| \|\boldsymbol{x}_t\|} \in [0, 1].$$

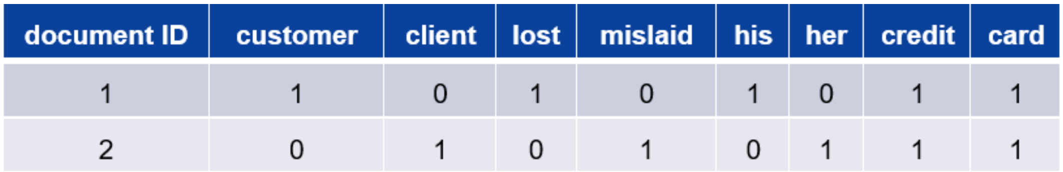
2.3.3 Semantic adjustment

The TF and TF-IDF approaches alone are not able to capture semantic information, such as the semantic similarity between synonyms. In fact, even if two documents are almost identical in terms of meaning, the similarity between them on the basis of TF or TF-IDF could be low due to scarce word matching. In the following example, we compare two descriptions:

1. "The customer lost his credit card" and
2. "The client mislaid her credit card"

The TF matrix is reported in Table 2.1 (stop-word "the" has been removed).

Table 2.1: TF document matrix.



document ID	customer	client	lost	mislaid	his	her	credit	card
1	1	0	1	0	1	0	1	1
2	0	1	0	1	0	1	1	1

Cosine similarity between the descriptions can be calculated as

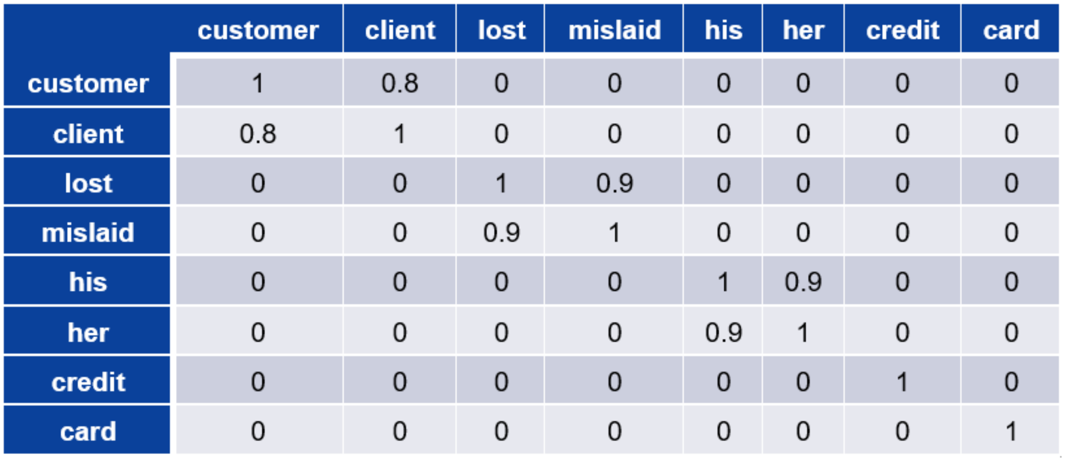
$$\mathrm{CS}(d_1,d_2) = \frac{\boldsymbol{x}_1 \cdot \boldsymbol{x}_2}{\|\boldsymbol{x}_1\|\|\boldsymbol{x}_2\|} = \frac{(1,0,1,0,1,0,1,1)\cdot(0,1,0,1,0,1,1,1)}{\sqrt{5}\sqrt{5}} = \frac{2}{5} = 0.4.$$

Even though the documents are almost identical, CS is low due to the poor word overlap. The columns “customer”, “lost” and “his” should be correlated respectively with the value of columns “client”, “mislaid” and “her”, since they represent the same concepts. To consider semantic similarity, an adjustment can be applied to the document-by-term matrix using word embedding techniques, such as `word2vec` (Mikolov *et al.*, 2013).

The `Word2vec` is built on a neural network-based algorithm to represent words in a vector space, so that different words that share a common concept are “close” as measured by cosine similarity. The cosine similarity between words therefore represents a measure of semantic similarity between them.

For example, assume that the word-similarity matrix, shown in Table 2.2, is obtained by applying `word2vec`.

Table 2.2: Word similarity matrix.

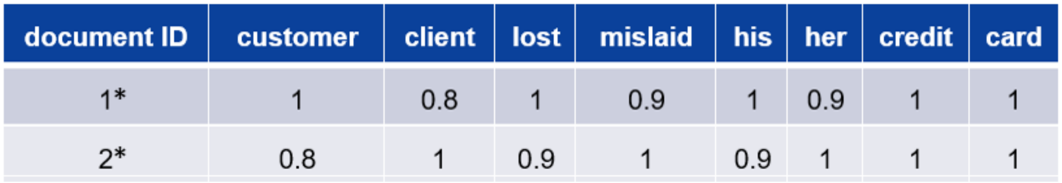


	customer	client	lost	mislaid	his	her	credit	card
customer	1	0.8	0	0	0	0	0	0
client	0.8	1	0	0	0	0	0	0
lost	0	0	1	0.9	0	0	0	0
mislaid	0	0	0.9	1	0	0	0	0
his	0	0	0	0	1	0.9	0	0
her	0	0	0	0	0.9	1	0	0
credit	0	0	0	0	0	0	1	0
card	0	0	0	0	0	0	0	1

The word similarity matrix allows each “zero” element of the document-by-term matrix to be updated with the value of the most similar word included in the same row of the matrix, scaled by the respective word similarity score (Shanavas *et al.*, 2021).

The resulting semantic-aware document-by-term matrix is reported in Table 2.3.

Table 2.3: Semantic-aware document-by-term matrix.



document ID	customer	client	lost	mislaid	his	her	credit	card
1*	1	0.8	1	0.9	1	0.9	1	1
2*	0.8	1	0.9	1	0.9	1	1	1

The cosine similarity between the two documents can now be recalculated on the basis of the semantic-aware document-by-term matrix:

$$\mathrm{CS}(d_{1^*}, d_{2^*}) = \frac{(1, 0.8, 1, 0.9, 1, 0.9, 1, 1) \cdot (0.8, 1, 0.9, 1, 0.9, 1, 1, 1)}{\sqrt{7.26}\sqrt{7.26}} = 0.992.$$

The similarity score between the two documents, considering the semantic adjustment, increases from 0.4 to around 0.99, reflecting the actual similarity between them.

2.3.4 Dimensionality reduction

After introducing a semantic measure of similarity to extract information from texts, the next step of the proposed workflow is to identify clusters of similar descriptions. A convenient approach is to make use of dimensionality reduction methods, used to map document vectors from the word space to a space whose reduced dimensionality is user-defined. The Latent Semantic Analysis (LSA) (Dumais *et al.*, 1988) is based on the Singular Value Decomposition (SVD) in which the document-by-term matrix A (see Figure 2.3) is reduced to a set of orthogonal factors from which the original matrix can be approximated. Multidimensional projection techniques such as Least Square Projection (LSP) (Paulovich *et al.*, 2008) can also be adopted to preserve neighborhood relations.

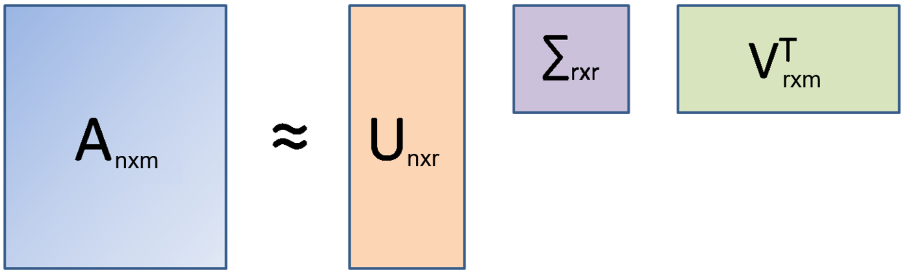


Figure 2.3: Singular value decomposition (SVD) representation.

Since the similarity between documents can still be measured in the reduced space represented by the matrix U (see Figure 2.3), text objects can be ranked by their similarity. For example, by identifying a point in the space (representing an event description), the text objects in its neighborhood can be identified.

2.3.5 Cluster selection

Dimensionality reduction is used to build a 2D representation of the data, where each point is an event description and similar ones are represented as a cluster of points (Eler *et al.*, 2018). Using the 2D representation, the analysts can explore a large volume of documents, identifying clusters of similar documents as groups of points close to each other, allowing them to understand their content and assign tags, as "credit card forgery", as reported in the example of Figure 2.4. For example, we suppose that most of the blue points can be tagged by the analysts as "credit card forgery".

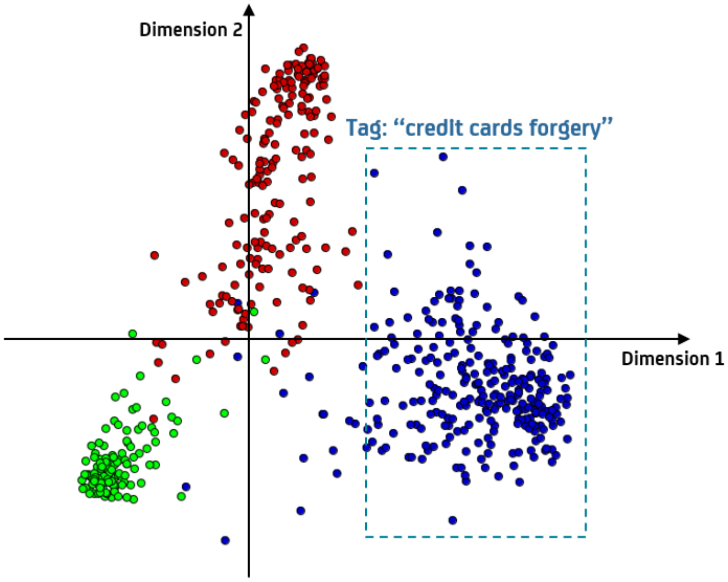


Figure 2.4: A graphical representation showing document similarities in the 2D space: each point represents a document, and each color represents a document cluster.

2.3.6 Cluster validation

Once the analysts have tagged the events belonging to the identified clusters, it is possible to apply statistical clustering and topic modelling techniques to validate their decisions. These techniques can be also applied to support analysts' activity when there is a huge amount of data to be tagged. Several methods can be adopted for this task. We have identified the following approaches as being the most used and popular ones:

- *k*-means clustering (Macqueen, 1967), where the data are partitioned into *k* groups, such that the sum of the squared Euclidean distances between the points and centers of the assigned clusters is minimized. The algorithm, starting from *k* initial centers, iterates until convergence by recalculating the centers of clusters and reassigning the points to the clusters on the basis of distances. Since the initial *k* centers are randomly selected, it is a good practice to rerun the algorithm with several initializations in order to select the best clustering among the results. *k*-means is implemented in the function `kmeans` of the R programming language. The quality of the obtained clustering can be assessed using the silhouette plots (Rousseeuw, 1987). For each data point, a silhouette value is calculated (and plotted), measuring how similar it is to its own cluster (cohesion) compared to the other clusters (separation). This value belongs to the range $[-1, +1]$, where a high value indicates that the point is well matched to its cluster and poorly matched to other clusters. If most points have high values, then the obtained clustering is appropriate. The average value, called the silhouette index, is usually adopted as a synthetic index of clustering quality.

- Spherical *k*-means clustering, which is based on cosine distance (*i.e.*, 1 minus the cosine similarity) instead of the Euclidean distance. Note that this method is equivalent to scaling data to unit length, and then using standard *k*-means. This method is suggested to mitigate the effect of different document lengths (Dhillon and Modha, 2001), and it is implemented in the R package `skmeans` (Hornik *et al.*, 2012).

- Clustering via Gaussian finite mixture models implemented in the R package `mclust` (Scrucca *et al.*, 2016).

- Trimmed *k*-means clustering implemented in the R package `tclust` (Fritz *et al.*, 2012). In particular, the trimmed *k*-means clustering is obtained by the function `tclust`, setting 1 as the eigenvalue restriction factor (*i.e.*, the ratio of the maximum eigenvalue to the minimum one).

- Mixtures of Unigrams described by Nigam *et al.* (2000) and implemented by the function `mou_EM` in the R package `DeepMOU` (D'Ippolito *et al.*, 2021). Parameter estimation is performed by means of the Expectation-Maximization (EM) algorithm.

- Deep Mixtures of Unigrams presented by Viroli and Anderlucci (2021) and implemented by the function `deep_mou_gibbs` in the R package `DeepMOU`. Parameter estimation is performed by means of Gibbs sampling.

- Dirichlet-Multinomial Mixtures model described by Anderlucci and Viroli (2020) and implemented by the function `dir_mult_GD` in the R package `DeepMOU`. Parameter estimation is performed by means of a Gradient Descend algorithm.

- Latent Dirichlet Allocation (LDA) is a generative statistical model that explains a set of observations through unobserved groups. It is an example of a topic model, where observations (*e.g.*, words) are collected into documents. It assumes that the words in a document are drawn from k topics, and that each topic is characterized by a probability distribution over the available words. Each document is supposed to contain a certain number of topics. The application of LDA in the context of text mining is described by Blei *et al.* (2003). LDA is implemented by the function `FitLdaModel` in the R package `textmineR` (Jones, 2021). Parameter estimation is performed by means of Gibbs sampling.

The consistency between the cluster selection, performed as described in Section 2.3.5, and the results of the aforementioned approaches can be assessed through the accuracy measure, which is calculated as follows for a classification method:

$$\text{Accuracy} = \frac{\text{Number of correctly classified data}}{\text{Total number of data}}.$$

Several other measures can be used to assess the performances of classification methods (*e.g.*, precision, recall, the F_1 score), but accuracy appears to be the most intuitive and sufficiently general to be applied to the aforementioned approaches.

2.4 Application to OpRisk data

The objective of this application is to analyze the descriptions of the CoRep OpRisk data set from the UniCredit banking group using all the approaches described in the previous sections. CoRep refers to Common Reporting, which is the set of all data that the financial institutions have to periodically report to their Supervisory Authorities (*e.g.*, European Central Bank). Among the CoRep reports, there is the C17.02 template, which reports information (including the description) on OpRisk events leading to gross loss amounts higher than or equal to € 100,000 (gross means without considering any recovery). The analyzed data set is composed of the OpRisk data which are relevant for the C17.02 template, which was introduced in 2018. Each record of this data set represents an OpRisk event, and the main fields report the following data:

- Event ID: the ID of the OpRisk event.
- Date of Accounting: the first accounting date of the event.
- Event Type: the Basel Event Type level 1 classification of the event.
- Gross Loss: the total gross loss amount of the event (*i.e.*, the sum of economic impacts related to the event: losses, provisions, and releases of provisions) in euros.
- Description: a text field reporting an English anonymized description, having a maximum of 250 characters.

This application concerns the part of CoRep OpRisk data set related to the OpRisk events having event type “Clients, Products & Business Practices” and first accounting date between 2018 and 2021. This selection leads to a data set composed of 644 events with relevant descriptions.

The analysis is performed using the R packages `quanteda` (Benoit *et al.*, 2018), `word2vec` (Wijffels, 2021), and the ones mentioned in Section 2.3.6.

First of all, the descriptions are cleaned as described in Section 2.3.1. There is no need for language splitting and data anonymization, since such descriptions are all entered in English language and without any personal information. The stop-word list, already specified in Section 2.3, has been obtained through the R package `stopword` (Benoit *et al.*, 2021).

Since the analyzed descriptions are short texts (having max 250 characters), we apply the TF weighting schema without any IDF scaling, as motivated by the finding of Anderlucci *et al.* (2019) in their application to similarly structured data. We point out that the main purpose of the IDF scaling is to reduce the weight of terms that are used in many documents under the hypothesis that if a word is used in many descriptions, then it is not informative, and then it is not useful to discriminate different clusters of data. However, most of the non-informative terms have been already excluded by removing the stop-words from the text corpus. Therefore, by applying IDF scaling to short texts, we would risk reducing the weights of some informative words that characterize the clusters. This aspect is verified in the next steps of the analysis.

We obtain a document-by-term matrix having 644 rows (*i.e.*, the number of descriptions) and 1037 columns (*i.e.*, the length of the dictionary consisting of all the unique words included in the cleaned descriptions).

We apply LSA to the document-by-term matrix to obtain a 2D representation, reported in Figure 2.5, where the axes $V1$ and $V2$ represent the first two LSA dimensions.

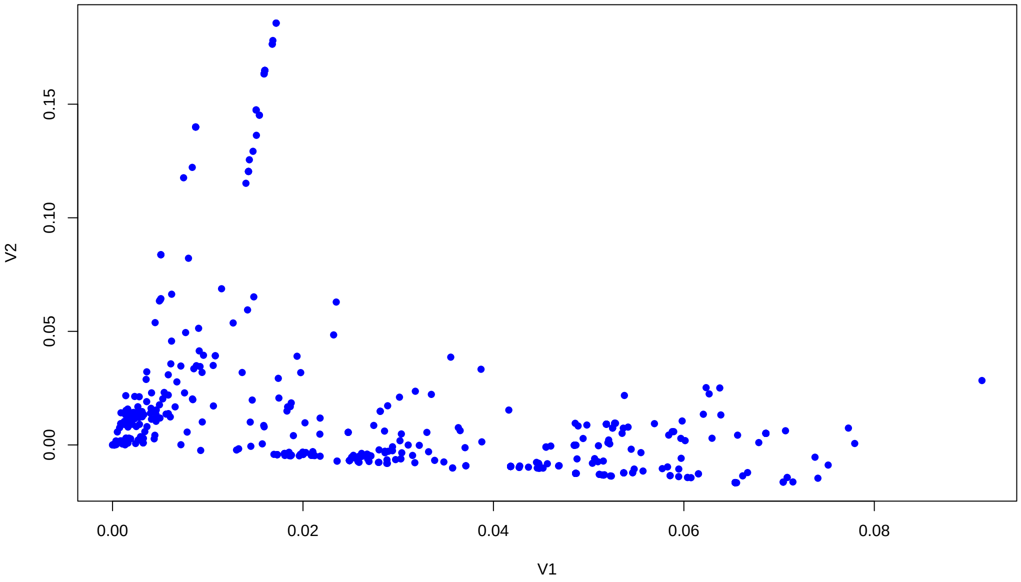


Figure 2.5: 2D representation of the document-by-term matrix

The next step is to generate the semantic-aware document-by-term matrix using the approach described in Section 2.3.3. We use a pre-trained word embedding obtained by the `word2vec` approach (available at *NLPL word embeddings repository* (2017), selecting ID=40, *i.e.*, ”English CoNLL17 corpus”). This allows us to obtain the word similarity matrix by calculating the cosine similarity between all the pairs of words included in the dictionary of the data set. The word similarity matrix is then used to adjust the document-by-term matrix, as described in Section 2.3.3. Similarly to Shanavas *et al.* (2021), we use a similarity matrix that only contains similarity values higher than 0.8 in order to keep the semantic adjustment free of noise (*i.e.*, medium-to-low similarity due more to randomness than similar meaning). Some rationales for the selection of threshold 0.8 are reported in the next steps of the analysis. After applying LSA, we obtain the 2D representation of the semantic-aware document-by-term matrix in Figure 2.6.

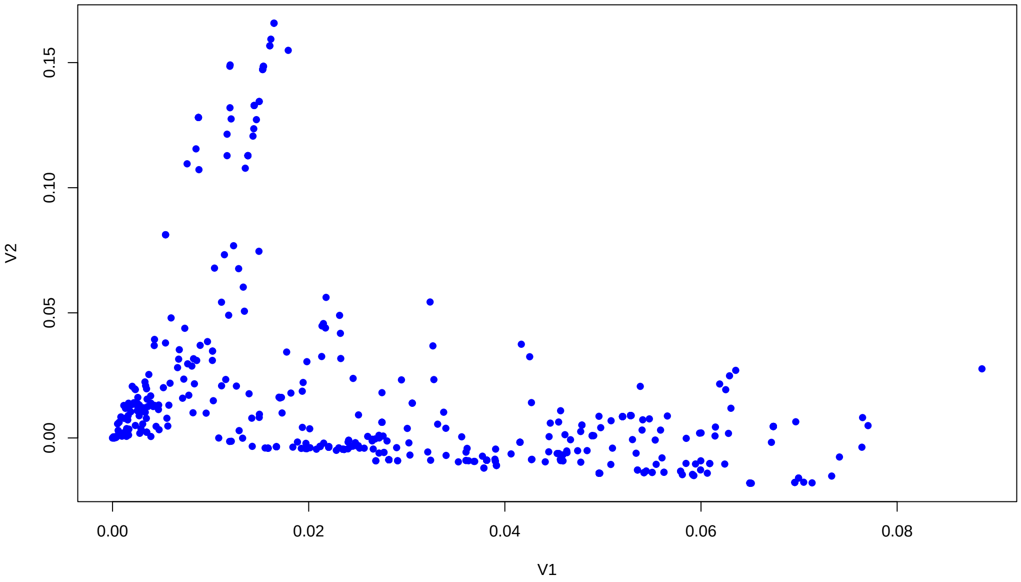


Figure 2.6: 2D representation of the semantic-aware document-by-term matrix.

This plot supports the activity of the analysts who, in our application, after examining which points lie closer to each other, decide to tag two clusters of events (clusters 1 and 2 in Figure 2.7) and to also designate a cluster of residual events (cluster 3 in Figure 2.7). There are two groups of semantically related words, identified by the analysts, that are contained in all descriptions within clusters 1 and 2, respectively. Based on these common-meaning words, it emerges that the two identified clusters (representing two different root causes for OpRisk) and the residual cluster can be described as follows:

1. Disputes related to irregularities in interest rate calculations (composed of 384 events)

2. Disputes related to mortgages in foreign currency (composed of 48 events)

3. Other events (composed of 212 events)

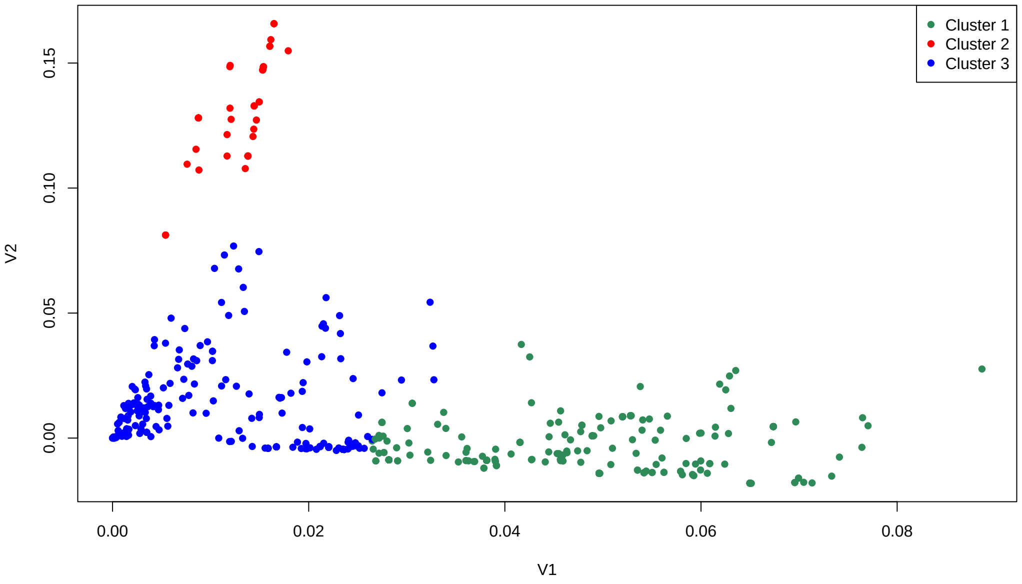


Figure 2.7: 2D representation of the semantic-aware document-by-term matrix with the identified clusters, with similarity threshold 0.8: cluster 1 represents "disputes related to irregularities in the interest rate calculations", cluster 2 identifies "disputes related to mortgages in foreign currency", and other events are in cluster 3.

As can be seen by comparing Figure 2.5 with Figures 2.6 and 2.7, the semantic-aware document-by-term matrix allows to include into the clusters also descriptions expressing similar concepts, even when they do not include the same significant words identifying the clusters.

To further motivate the exclusion of IDF scaling, Figure 2.8 graphs the 2D representation of the semantic-aware TF-IDF matrix (*i.e.*, the TF-IDF matrix including the semantic adjustment based on similarity threshold 0.8) with clusters previously identified by the analysts.

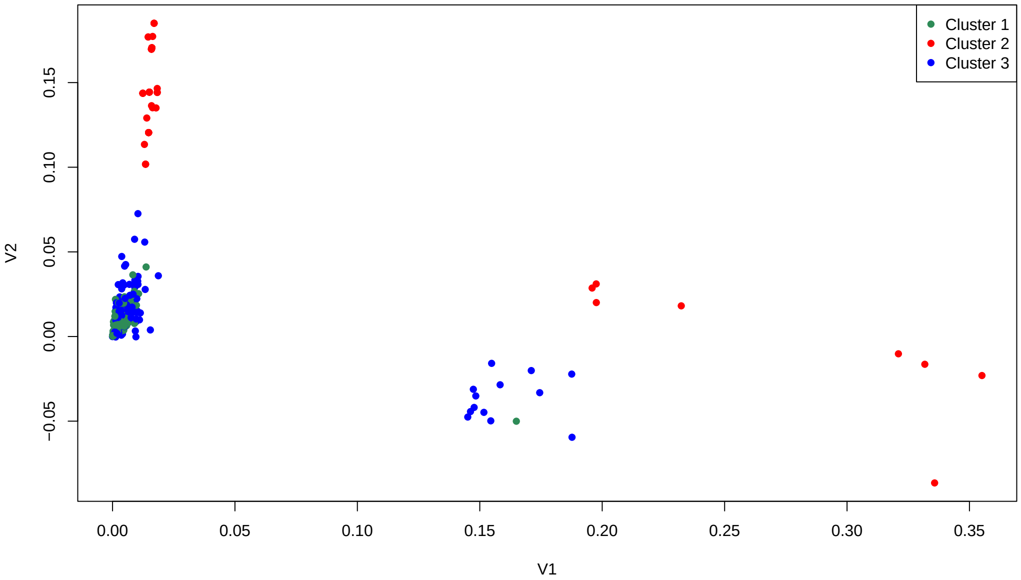


Figure 2.8: 2D representation of the semantic-aware TF-IDF matrix with the identified clusters: cluster 1 represents "disputes related to irregularities in the interest rate calculations", cluster 2 identifies "disputes related to mortgages in foreign currency", and other events are in cluster 3.

In Figure 2.8 there appears a significant overlap between clusters 1 and 3. In fact, since events related to cluster 1 are identified by a few words that are included in all its descriptions, the IDF scaling significantly reduces the weights of such terms, moving most of the related points very close to the chart origin (*i.e.*, very close to the point $(V1 = 0, V2 = 0)$ in the chart). This representation would make it very hard for the analysts to distinguish between cluster 1 (disputes related to irregularities in the interest rate calculations) and the residual cluster.

To motivate the selection of similarity threshold 0.8, we also report the charts obtained with similarity thresholds 0.7 (in Figure 2.9) and 0.9 (in Figure 2.10).

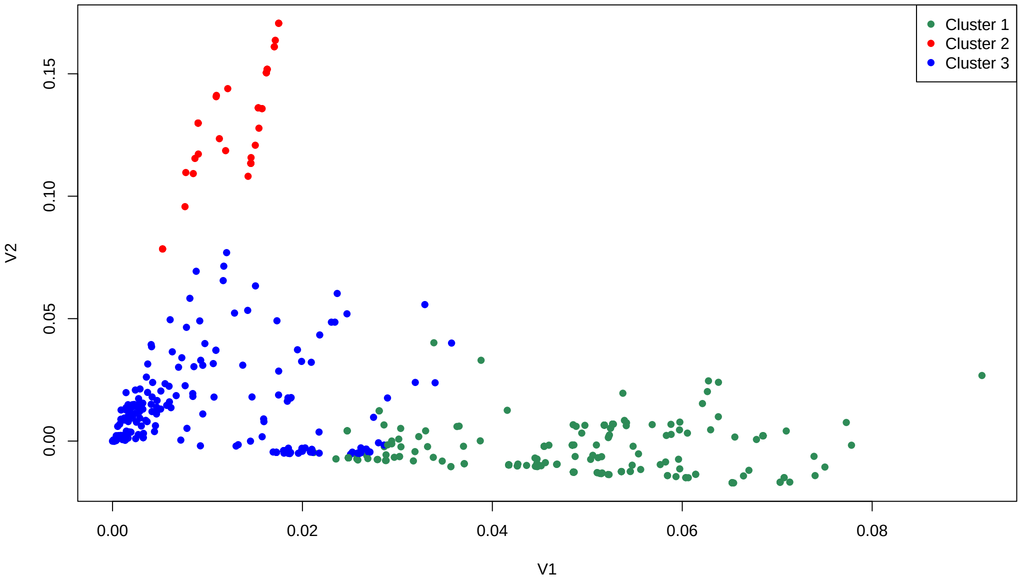


Figure 2.10: 2D representation of the semantic-aware document-by-term matrix with the identified clusters, considering similarity threshold 0.9.

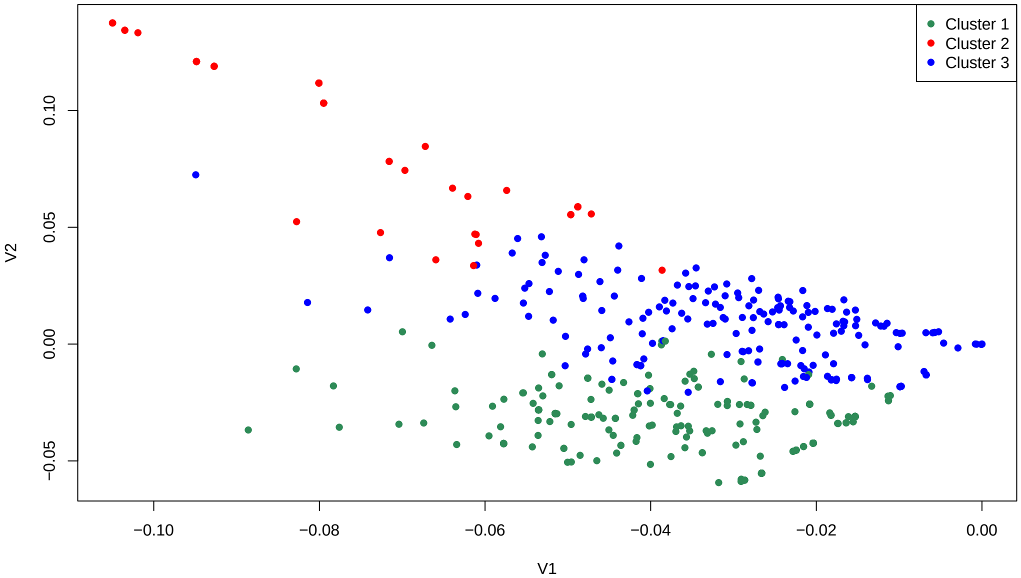


Figure 2.9: 2D representation of the semantic-aware document-by-term matrix with the identified clusters, considering similarity threshold 0.7.

We can see from Figures 2.9 and 2.10 that considering threshold 0.7 completely alters the initial configuration, whereas considering 0.9 leaves the configuration very similar to the non-semantic-aware one (Figure 2.5). Therefore, we can consider the similarity threshold 0.8 (or, at least, the values within a small neighborhood of 0.8) as the best trade-off between including too much noise (*i.e.*, threshold 0.7) and not including any appreciable semantic adjustment (*i.e.*, threshold 0.9).

Taking into account the knowledge of the analysts, who identified three clusters (*i.e.*, the two clusters based on common root cause events and the cluster of residual data), we run a k-means clustering with $k = 3$. Moreover, 1000 random starting points are used to avoid being sensitive to a specific starting point selection. Results are reported in Figure 2.11.

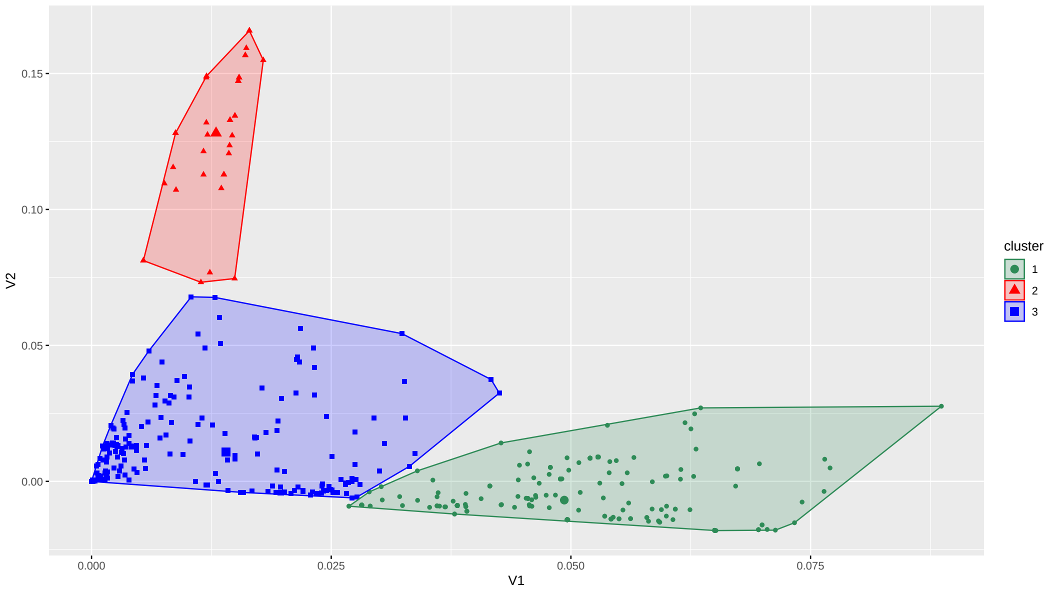


Figure 2.11: k-means clustering with $k = 3$ and 1000 starting points.

The good quality of the k-means clustering is confirmed by the silhouette plot, in which the average silhouette index is 0.71 (Figure 2.12).

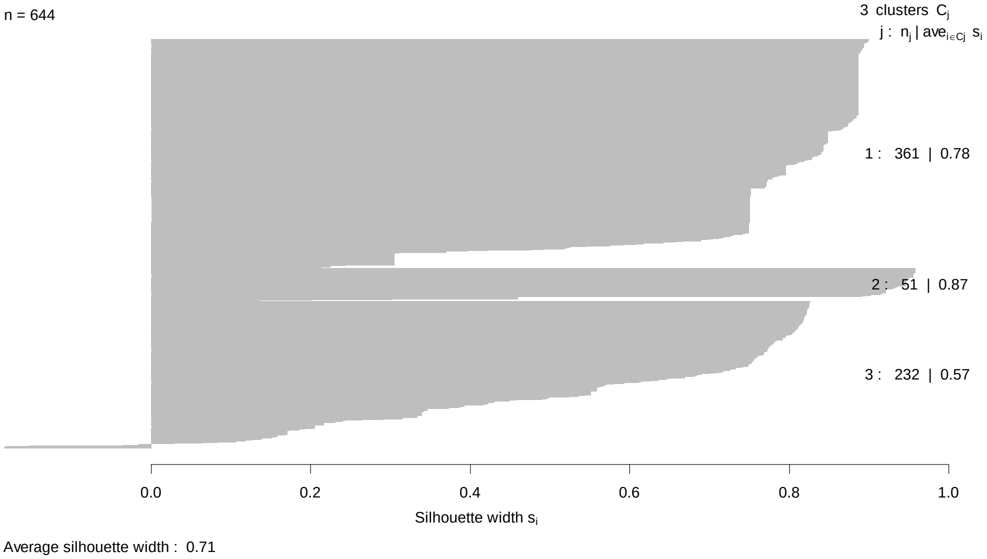


Figure 2.12: Silhouette plot of the *k*-means clustering.

The *k*-means clustering assigns 361 and 51 events to clusters 1 and 2, respectively. As there are 26 misclassified events out of 644 with respect to the analysts' selection, we obtain an accuracy of around 96% for the *k*-means clustering.

Other methods described in Section 2.3.6 are also applied to the data. The accuracies of all considered approaches are reported in Table 2.4, along with their ranking from most to least accurate.

Table 2.4: Accuracy indexes.

Rank	Method	Accuracy (%)
1	*k*-means	95.96
2	Gaussian finite mixture models	95.19
3	Trimmed *k*-means	95.03
4	Latent Dirichlet Allocation	85.40
5	Dirichlet-Multinomial Mixtures	76.05
6	Mixtures of Unigrams	70.92
7	Deep Mixtures of Unigrams	69.98
8	Spherical *k*-means	61.02

From the reported results, we observe that:

- The highest accuracy (*i.e.*, 96%) is shown by k-means clustering, applying 1000 different starting points, To additionally motivate the exclusion of IDF scaling, we also apply the k-means method to the semantic-aware TF-IDF matrix, recalculating the first two LSA dimensions. In this case, the accuracy drops to 68%. This significant decrease with respect to the results obtained without the IDF scaling is consistent with Figure 2.8 (showing substantial overlapping between clusters 1 and 3).

  To also provide further justification for the similarity threshold 0.8, we calculate all the accuracy values that we would obtain by applying k-means to the first two LSA dimensions recalculated on the semantic-aware document-by-term matrix based on similarity thresholds between 0.7 and 0.9 (with step 0.05).

  Table 2.5: Accuracy indexes for similarity thresholds.

Threshold	Accuracy (%)
0.70	89.29
0.75	95.81
0.80	95.96
0.85	90.99
0.90	90.68

  The results, reported in Table 2.5, confirm that the similarity threshold 0.8 is the best setting also in terms of accuracy.

- Spherical k-means (*i.e.*, the k-means based on the cosine distance) ranked last (*i.e.*, 61%). It is applied with 10,000 starting points, although this does not substantially improve accuracy. The poor performance of spherical k-means could be due to the similar lengths of analyzed descriptions. In this case, the normalization of the vectors representing the descriptions does not seem to be effective in discriminating the correct clusters. Intuitively, comparing Figures 2.11 and 2.13, it can be noted that the similarity among data is much more due to their Euclidean distance than to the angles between the vectors representing each pair of points.

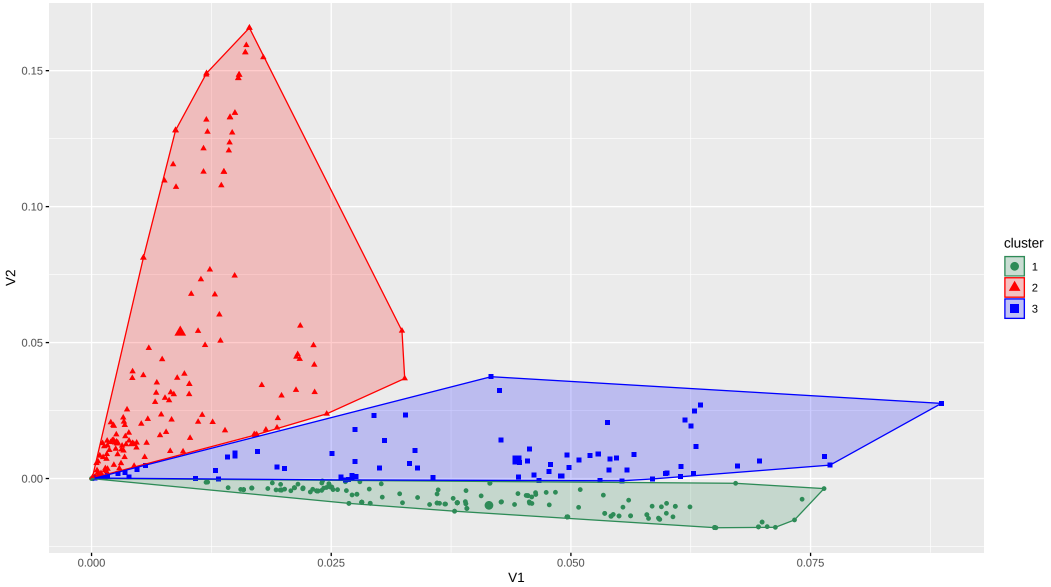


Figure 2.13: Spherical *k*-means clustering.

- Gaussian finite mixture models provide a slightly lower accuracy than *k*-means (*i.e.*, 95%). This level of accuracy has been achieved by considering a spherical family with variable volume (*i.e.*, each cluster can include a different number of observations) and equal shape (*i.e.*, each cluster has approximately the same variance so that the distribution is spherical). This setting leads to a configuration similar to the one obtained by the *k*-means clustering and, consequently, to a similar accuracy level. The obtained clustering is reported in Figure 2.14.

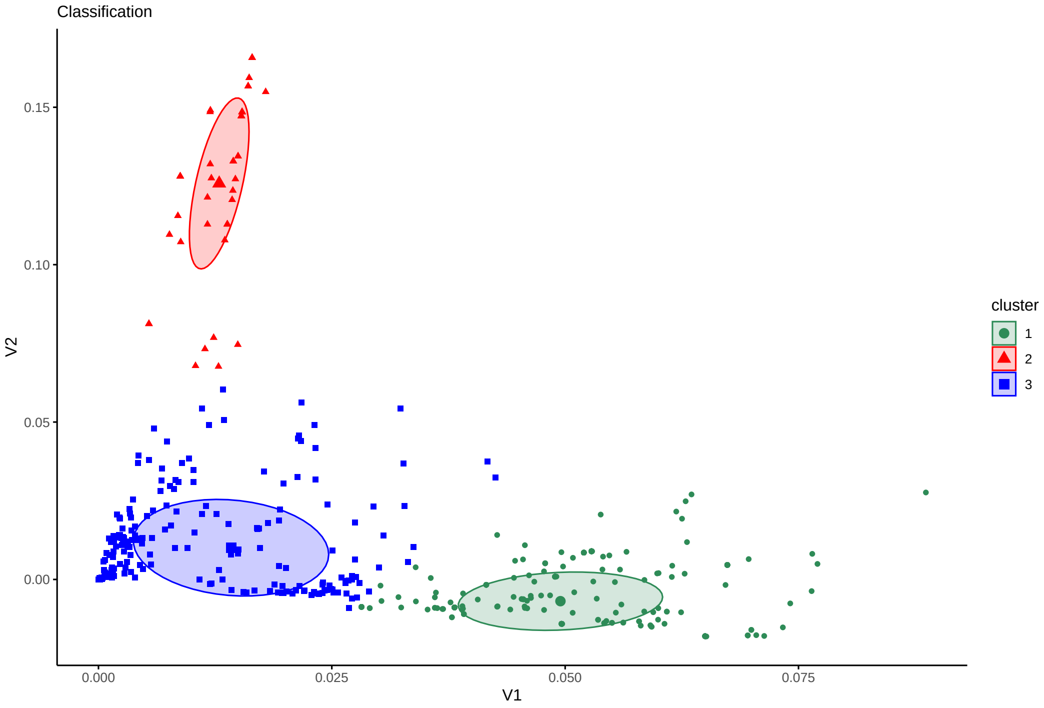


Figure 2.14: Clustering via Gaussian finite mixture models.

- The method of trimmed k-means provides similar accuracy (*i.e.*, 95%) to the Gaussian finite mixture models, again applying 1000 different starting points. Different settings have been tested for this method, and the best one (in terms of accuracy) resulted in a proportion of $\alpha = 0.02$ trimmed observations (Figure 2.15, where the black circles represent the trimmed data, which are not assigned to any cluster).

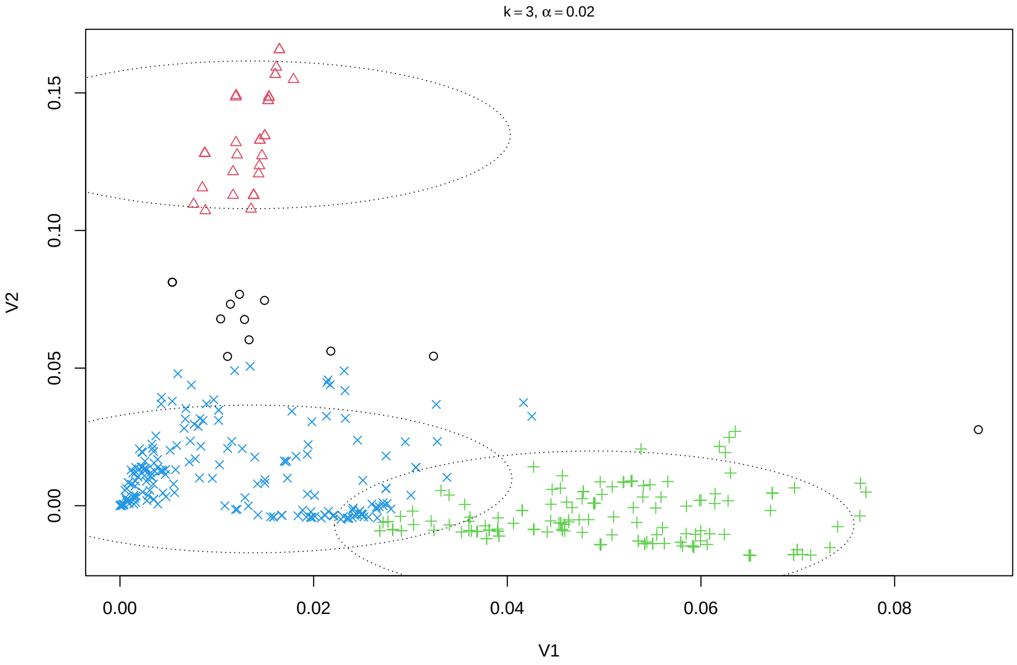


Figure 2.15: Trimmed k-means clustering with $\alpha = 0.02$.

To motivate the choice $\alpha = 0.02$, we calculated the accuracy for values of α between 0.01 and 0.1.

Table 2.6: Accuracy indexes.

α	Accuracy (%)
0.01	91.30
0.02	95.03
0.05	92.39
0.1	77.80

The results, reported in Table 2.6, confirm that the highest accuracy value is obtained for $\alpha = 0.02$.

- Mixtures of Unigrams, Deep Mixtures of Unigrams, and Dirichlet-Multinomial Mixtures are applied directly to the document-by-term matrix, adjusted for the semantic similarity, composed of 644 documents and 1037 terms. In fact, it is not possible to apply these methodologies to the LSA-based 2D representation, since the LSA can generate negative values, and the three methodologies require as input a matrix composed of positive values. Mixtures of Unigrams and Dirichlet-Multinomial Mixtures have been applied using a multi-starting strategy to prevent the local maxima issue, where, for each iteration, the initial assignment to the clusters has been randomly defined. Among all performed iterations, the one having the lowest Bayesian information criterion (BIC) index has been selected, assuring that we have approximately obtained a global maximum value for parameter estimation. The implementation details are as follows:

- For Mixtures of Unigrams, 1000 iterations (*i.e.*, 10 times the default setting of the function `mou_EM`) and a tolerance of 10^{-7} (equal to the default setting) have been applied. For the multi-starting strategy, 100 different random starting points have been considered.
- For Deep Mixtures of Unigrams, based on Gibbs sampling, 1000 iterations have been used with a burn-in of 500. For the top layer, three clusters have been considered, whereas two clusters have been considered for the hidden bottom layer, since this setting provided the highest accuracy in the simulation studies performed by Viroli and Anderlucci (2021).
- For Dirichlet-Multinomial Mixtures, 100 iterations have been set, combined with 100 different random starting points for the multi-starting strategy. It is worth mentioning that Dirichlet-Multinomial Mixtures are much more computationally intensive than the Mixtures of Unigrams and the Deep Mixtures of Unigrams, with calculations taking several hours.

These three methods, implemented in the R package `DeepMOU`, show accuracies between 70% and 76%. For these approaches, better performances can perhaps be obtained by trying different settings and, in particular, increasing the iterations at the price of higher computational costs.

- For the same reason as for the previous methods, the LDA has also been directly applied to the document-by-term matrix, adjusted for semantic similarity. The applied LDA setting considers three topics (since the analysts identified two clusters, besides the residual data), and 10,000 iterations with a burn-in of 5000. The prior parameters for topics over documents and for words over topics have been set to $\alpha = 0.1$ and $\beta = 0.05$ (*i.e.*, the default values of the function `FitLdaModel`). To obtain the clustering, we assign each description to the topic showing the highest probability. The LDA shows an accuracy of around 85%, which could be likely improved by trying different settings, such as increasing the number of iterations and fine-tuning the values for prior parameters α and β. However, note that setting α to 50 divided by the number of topics (*i.e.*, 50/3 for this application), as suggested by Grün and Hornik (2011), does not increase the accuracy.

2.5 Concluding remarks

To the best of our knowledge, the present work is among the first ones that have addressed the application of text analysis techniques to OpRisk event descriptions and is the first one that has provided a structured general workflow for such analyses. Furthermore, we have complemented the established frameworks of currently applied statistical methods for quantitative data, hence contributing to the construction of a holistic OpRisk management framework. Indeed, our ultimate goal is to provide an analytical and measurement framework that considers OpRisk information in its entirety in order to acquire a common language and a unified understanding of risk.

We have applied several statistical approaches and models to analyze and cluster OpRisk event descriptions using text analysis techniques, in order to identify the main root causes of such a risk. We have enriched the standard text analysis techniques by considering a semantic adjustment capable of dealing with similar concepts expressed by different words. The semantic adjustment can be based on word embedding methods, such as `word2vec`. We have used clustering and topic-modelling techniques (*e.g.*, *k*-means, and LDA) to validate and support the clustering performed by the analysts. Conversely, the information provided by analysts (such as the number of clusters to be considered) can serve as useful guidance for statistical methods.

We focused on the UniCredit CoRep data set when applying the described text analysis methods and several clustering approaches, thus providing a useful comparison that highlights their advantages and limitations. Our results have allowed us to identify two homogeneous clusters of events within the event type "Clients, Products & Business Practices" concerning "disputes related to irregularities in the interest rate calculations", "disputes related to mortgages in foreign currency", and a residual cluster containing other events within the same event type. Such results have been validated by statistical indices. Notably, the indexes were consistent with the judgments and knowledge of skilled analysts in the field. The k-means clustering method provided the highest accuracy relative to the clusters identified by the analysts. However, further analysis of more extended data sets should be performed before drawing conclusions on the best methodologies for these purposes. The proposed framework constitutes a starting point for analyzing OpRisk event descriptions. It could be improved and extended by focusing on several aspects:

- Including the procedure, described in Section 2.3.5, in an analytical loop. At each iteration, event descriptions belonging to the identified clusters can be labeled and then removed from the data set. A tag deduction activity can be performed to infer tags of new events from tagged events with similar descriptions by using, *e.g.*, a k-nearest neighbors approach.

- Systematically applying clustering and topic modelling techniques to partially automate the identification of the clusters on large data sets.

- Employing techniques to drive the selection of the number of relevant clusters or topics (*e.g.*, identifying the number of clusters that maximizes the average silhouette index).

- Adopting multidimensional projection techniques, such as Least Square Projection (LSP) (Paulovich *et al.*, 2008), or self-organizing maps (SOM) (Pacella *et al.*, 2016), to preserve neighborhood relations and improve cluster identification.

- Adopting other word embedding techniques, such as GloVe (Pennington *et al.*, 2014) or BERT (Kaliyar, 2020) and training them on large OpRisk data sets.

Chapter 3

An approach for detecting emerging Operational Risks from textual data

Based on:

Di Vincenzo D., Greselin F., Piacenza F., & Zitikis R. (2024).
An approach for detecting emerging Operational Risks from textual data.
Forthcoming.

Di Vincenzo D., Greselin F., Piacenza F., & Zitikis R. (2024).
A tweet data analysis for detecting emerging Operational Risks.
Submitted to *11th International Conference MATHEMATICAL AND STATISTICAL METHODS FOR ACTUARIAL SCIENCES AND FINANCE - MAF2024* and the related book edited by Springer.
https://sites.google.com/unisa.it/maf-2024/home-page
https://sites.google.com/unisa.it/maf-2024/conference-publications?authuser=0

3.1 Introduction

The operational risk (or OpRisk) is related to the risk of losses resulting from events such as frauds, sanctions, physical damage, IT issues, cyberattacks, and errors (refer to European Parliament and Council of the European Union, 2013 for the official definition). International financial institutions have OpRisk management functions, performing regulatory activities, such as loss data collection, capital requirement calculations, and reporting. To perform loss data collection, financial institutions have databases to collect and store the necessary information for each OpRisk event, such as loss amounts, reference dates, Basel loss event type (selected among Internal Fraud; External Fraud; Employment Practices and Workplace Safety; Clients, Products & Business Practices; Damage to Physical Assets; Business Disruption and System Failures; Execution, Delivery and Process Management), and event description (for example, refer to Soprano *et al.*, 2010). The present work addresses the application of text analysis techniques to the OpRisk event descriptions and other data sources, such as web data, proposing a fully integrated workflow. Text analysis is one of the main tasks of Natural Language Processing (NLP), which is a branch of Artificial Intelligence (AI).

The proposed workflow includes the following steps for OpRisk event descriptions:

1. Description cleaning (*e.g.*, identifying English-written descriptions, removing stop-words, reducing words to their lemmas).

2. Text vectorization (building a document-by-term matrix).

3. Semantic adjustment (enriching the document-by-term matrix, considering the semantic similarity among words).

4. Dimensionality reduction (building a 2D representation of the data, where each point is an event description, and similar ones are represented as clusters of points).

5. Cluster selection (points are automatically clustered, according to the evidence that emerges from the 2D representation).

For web data, and in particular tweets, the same steps are considered, and integrated by two final steps:

6. Observe the trend of OpRisk related topics.

7. Detect emerging OpRisk related topics.

The entire integrated workflow is represented in Figure 3.1. In this work, Section 3.2 gives a literature review of the text analysis applied to OpRisk. Sections 3.3 and 3.4 describe data sets related

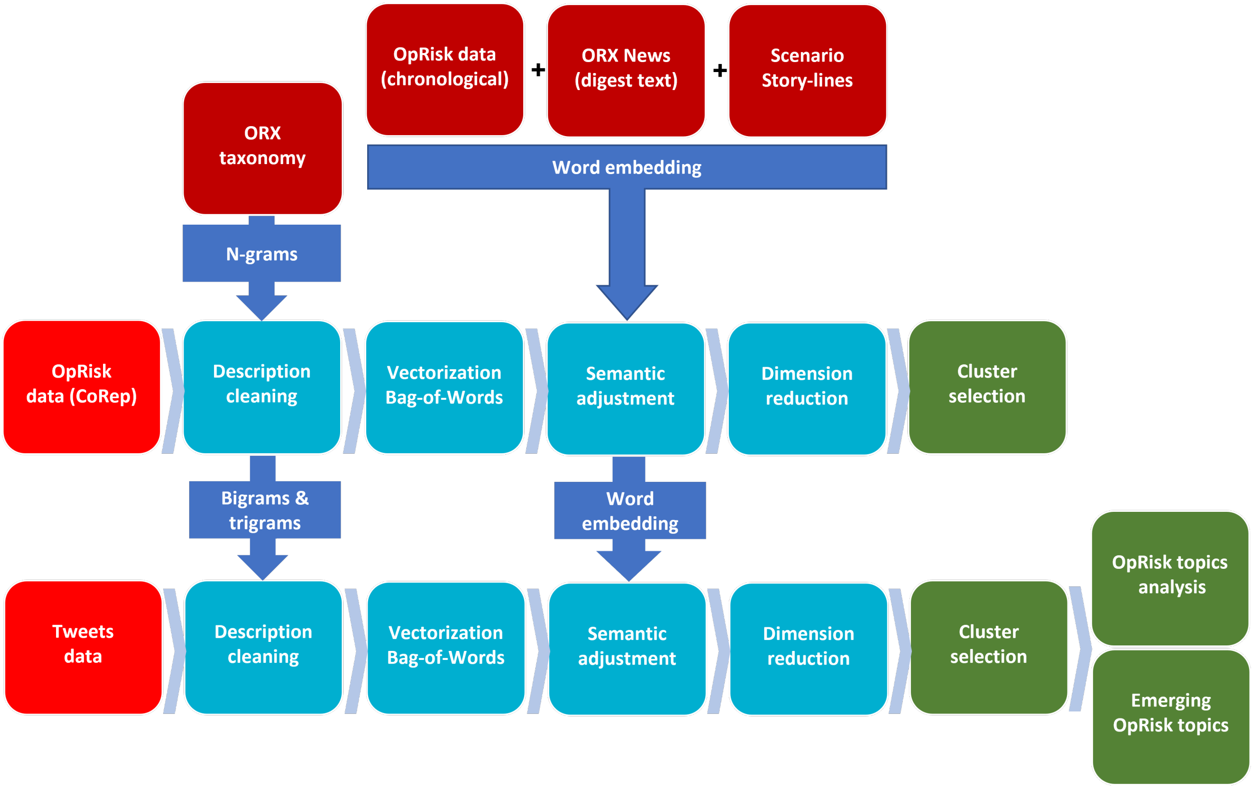


Figure 3.1: Workflow for OpRisk description and tweet analyses.

to OpRisk event descriptions and tweets, mentioning their sources and structures. Section 3.5 describes in detail the steps of the proposed workflow for OpRisk event descriptions. Section 3.6 describes the steps of the proposed workflow for web data sources. Section 3.7 reports an application of the proposed workflow to the descriptions of the CoRep OpRisk data set for the UniCredit banking group, while Section 3.8 describes an application on tweets. All the data elaborations and analyses have been performed using the software R (R Core Team, 2023). Finally, Section 3.9 summarizes the main achievements and results of this work, discussing possible extensions in several directions.

3.2 Literature review

Some approaches for the analysis of textual data related to OpRisk loss event descriptions are available in the literature.

Pakhchanyan *et al.* (2022) apply machine learning techniques to OpRisk descriptions, to automatically classify events into Basel event types. The classification of OpRisk events is also discussed by Zhou *et al.* (2021), who propose semi-supervised methods to include unlabeled data in the training stage.

Wang *et al.* (2018) and Wang *et al.* (2022) investigate the main OpRisk factors, applying the Latent Dirichlet Allocation (LDA).

Data Study Group team (2019) provide a preliminary proof-of-concept for the potential usefulness of statistical and NLP approaches in OpRisk modelling, applying LDA and long short-term memory neural networks (LSTM).

Carrivick and Westphal (2019) suggest that text analysis methodologies can be useful to gain deeper insights into the OpRisk data.

Ji *et al.* (2023) use BILSTM-CRF, *i.e.*, a text mining method that combines long short-term memory (LSTM) and conditional random field (CRF), for safety record analysis.

Di Vincenzo *et al.* (2023) define a structured workflow to perform text analysis of OpRisk event descriptions, comparing several clustering and topic modelling methods to be applied within each Basel event type to detect their main root causes. The approach has been applied to a clean selection of OpRisk data.

In general, natural language processing (NLP) is often used to gain insights from unstructured risk data (Leidner and Schilder, 2010). Applications include identifying key risks (*e.g.*, Chu *et al.*, 2020) and data pre-processing to extract relevant factors from free-text reports (*e.g.*, Pence *et al.*, 2020). Arumugam *et al.* (2016) perform descriptive analytics with k-means clustering on risk phrases extracted from reports of offset wells, using NLP to streamline well drilling planning and execution.

To the best of our knowledge, however, no attempt has been made, up to now, to integrate different data sources (*i.e.*, not OpRisk-specific) to retrieve information that could be used as early warnings by the financial institutions.

3.3 OpRisk data

By OpRisk data analysis we mean the analysis of data on the OpRisks owned by the financial institution (*e.g.* UniCredit). We can differentiate the data sources among the following types:

- Internal data, *i.e*, the OpRisk events registered by the financial institution. For each OpRisk event of the internal data, we have two versions of the descriptions, *i.e.*, short and long descriptions:

  - the CoRep description (CoRep stands for Common Reporting, *i.e*, the set of data that the financial institutions have to periodically report to their Supervisory Authorities), with maximum length of 250 characters and is typically written in English (surely written in English if the loss amount is higher than or equal to the threshold € 100,000 and collected since 2018);
  - the chronological description with variable length (without a cap) and written in different languages (the same description can include parts written in different languages). The maximum observed chronological description length is around 4000 characters.

- External data, *i.e*, the OpRisk events registered by other financial institutions. It is possible to access these data by joining the ORX association (Operational Risk eXchange, which is the largest OpRisk management association in the financial services sector, ORX, 2023) and, in particular, accessing the ORX News Service (ORX News, 2023) to get the descriptions of the publicly reported OpRisk events from around the world. For each event of the ORX News, we have two versions of the descriptions, *i.e.*, short and long descriptions:

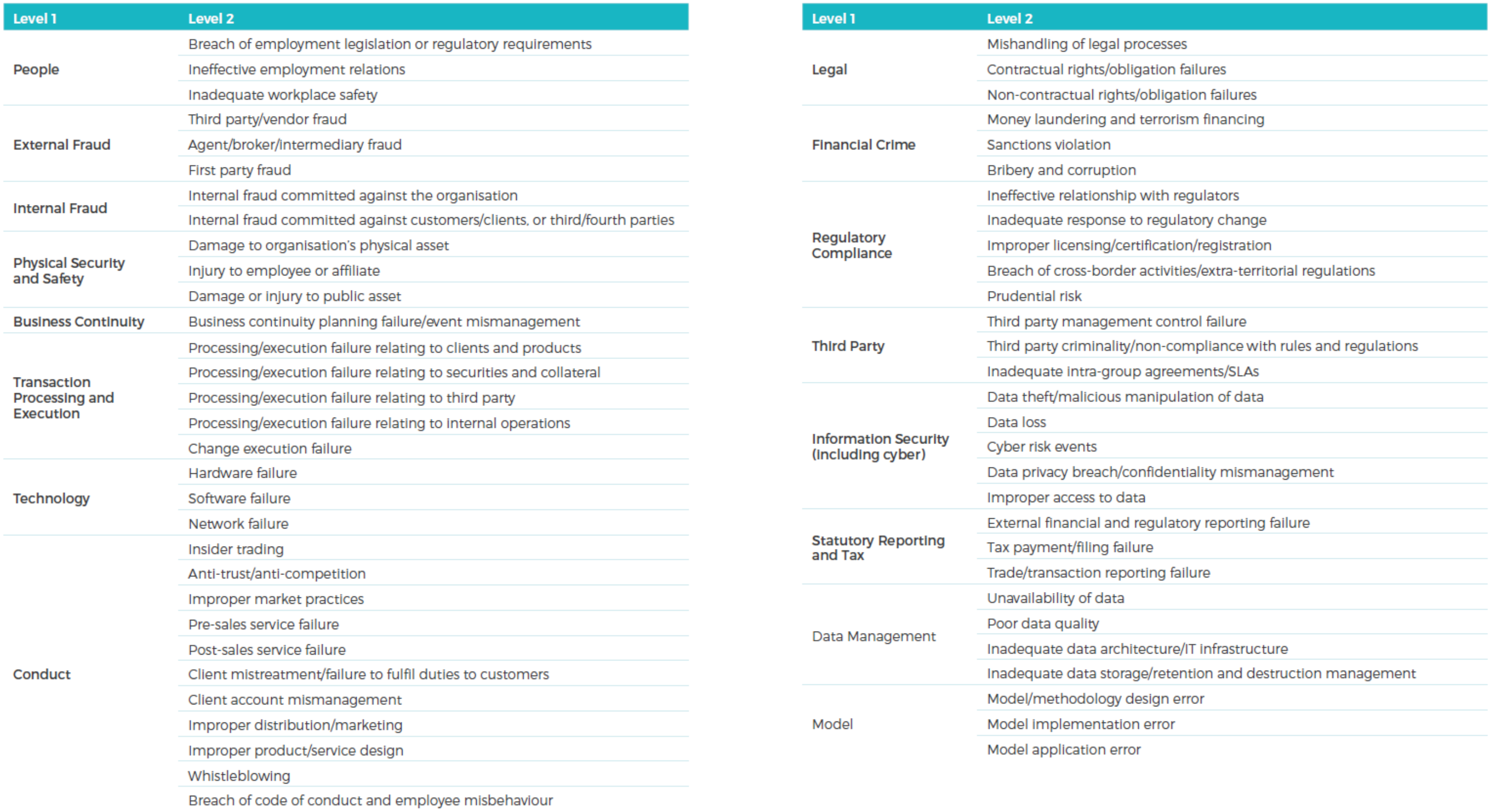
  - a brief description, named "Headline", which never exceeds 200 characters;
  - the full content of the news, named "Digest Text", with variable (typically high) length. The maximum observed Digest Text length is around 24,000 characters.

- Scenario analysis, *i.e*, fictitious OpRisk events that could impact the financial institution. For each scenario, a storyline describing the potential event is produced by OpRisk analysts. For the scenario analysis, only a storyline is typically available, where, *e.g*, in UniCredit, the length ranges between 100 and 4000 characters.

3.3.1 The ORX taxonomy

In recent years, the landscape of OpRisks has seen a drastic evolution with new entries into the risk vocabulary, *e.g.*, cyber risk and conduct risk. Considering that the Basel loss event type taxonomy was defined around 20 years ago (Basel Committee on Banking Supervision 2004), some of these recently emerging themes are not explicitly captured or addressed in it. Therefore, several financial institutions have either adapted the Basel event types, or developed internal taxonomies, leading to some divergence among organisations. For this reason, ORX, in collaboration with Oliver Wyman and their members, has developed a new reference OpRisk taxonomy ("the ORX taxonomy") to be used as a new standard to categorize OpRisks (ORX and Oliver Wyman, 2020; ORX and Oliver Wyman, 2023). The ORX taxonomy, composed of Level 1 and Level 2, is represented in Figure 3.1. The presented work makes extensive use of the ORX taxonomy for several purposes.

Table 3.1: ORX taxonomy.



Level 1	Level 2
People	Breach of employment legislation or regulatory requirements
	Ineffective employment relations
	Inadequate workplace safety
External Fraud	Third party/vendor fraud
	Agent/broker/intermediary fraud
	First party fraud
Internal Fraud	Internal fraud committed against the organisation
	Internal fraud committed against customers/clients, or third/fourth parties
Physical Security and Safety	Damage to organisation's physical asset
	Injury to employee or affiliate
	Damage or injury to public asset
Business Continuity	Business continuity planning failure/event mismanagement
Transaction Processing and Execution	Processing/execution failure relating to clients and products
	Processing/execution failure relating to securities and collateral
	Processing/execution failure relating to third party
	Processing/execution failure relating to internal operations
	Change execution failure
Technology	Hardware failure
	Software failure
	Network failure
Conduct	Insider trading
	Anti-trust/anti-competition
	Improper market practices
	Pre-sales service failure
	Post-sales service failure
	Client mistreatment/failure to fulfil duties to customers
	Client account mismanagement
	Improper distribution/marketing
	Improper product/service design
	Whistleblowing
	Breach of code of conduct and employee misbehaviour
Legal	Mishandling of legal processes
	Contractual rights/obligation failures
	Non-contractual rights/obligation failures
Financial Crime	Money laundering and terrorism financing
	Sanctions violation
	Bribery and corruption
Regulatory Compliance	Ineffective relationship with regulators
	Inadequate response to regulatory change
	Improper licensing/certification/registration
	Breach of cross-border activities/extra-territorial regulations
	Prudential risk
Third Party	Third party management control failure
	Third party criminality/non-compliance with rules and regulations
	Inadequate intra-group agreements/SLAs
Information Security (including cyber)	Data theft/malicious manipulation of data
	Data loss
	Cyber risk events
	Data privacy breach/confidentiality mismanagement
	Improper access to data
Statutory Reporting and Tax	External financial and regulatory reporting failure
	Tax payment/filing failure
	Trade/transaction reporting failure
Data Management	Unavailability of data
	Poor data quality
	Inadequate data architecture/IT infrastructure
	Inadequate data storage/retention and destruction management
Model	Model/methodology design error
	Model implementation error
	Model application error

3.4 Tweets data

X (formerly known as Twitter) has been selected as a relevant web data source to capture early warnings on potential OpRisk events, for three main reasons:

- Each tweet has a maximum length of 280 characters, making tweets comparable with the CoRep descriptions of UniCredit OpRisk data.
- It was feasible (until July 12th, 2023) to access and store the tweets using a specific API freely available for the development of research activities (after Twitter approval).
- Considering the widespread use of Twitter, it is likely that some tweets are promptly written, when some relevant event occurs (having or not an OpRisk nature).

Early warning systems are extensively used in finance (a bibliometric analysis can be found in Klopotan *et al.* (2018), while Zhang *et al.* (2019) studied them for stock market crises). Furthermore, tweet analysis is often used for predictive purposes (Cano-Marin *et al.*, 2023; Iacopini and Santagiustina, 2021; Costola *et al.*, 2021).

Two different R scripts based on the package `rtweet` (Kearney, 2019) have been written to access and store tweets, and then scheduled to automatically run using the R package `taskscheduleR` (Wijffels and Belmans, 2023):

- Script to extract tweets related to specific keywords, mainly based on the ORX taxonomy, scheduled to run every hour. The extraction was performed using the function `search_tweets`. The list of the keywords, grouped by generic OpRisk topic (obtained aggregating some Level 1 categories of ORX taxonomy), is reported in Table 3.2.
- Script to extract tweets related to specific accounts, selected among the main ones reporting financial news (*e.g.*, Financial Times, Bloomberg, Reuters), scheduled to automatically run every day. The extraction was performed using the function `get_timeline`. The list of accounts is reported in Table 3.3.

The different frequencies of extraction are motivated by the number of tweets expected for each type of search, and considering that each extraction cannot exceed around 100,000 tweets. We empirically observed this limitation, even if we are not aware of any specific reference motivating

Table 3.2: Tweet keywords.

Topic	Keywords
Fraud	fraud, rubbery, theft
Physical security	damage, injury, terrorism
Execution	error, model error, implementation error
Technology	hardware failure, software failure, IT failure, business continuity
Conduct and Legal	sanction, breach, compliance, regulators, fines
Financial Crime	bribery, corruption, money laundering
Third Party	third party, outsourcing
Information Security	cyber
General	operational risk

Table 3.3: Tweet accounts.

Source	Accounts
News agencies	@FinancialNews, @CBSNews, @cnnbrk, @FoxNews
BBC news	@BBCWorld, @BBCNews, @bbcworldservice, @BBCBreaking
Financial Times	@FinancialTimes, @FT, @ftlive
Bloomberg	@business, @Bloomberg, @markets, @BloombergTV @BloombergUK, @opinion, @BloombergLive
Reuters	@Reuters", @ReutersWorld
Risk.net	@RiskDotNet, @RiskNet_REG, @RiskNet_RM, @RiskNet_COM @RiskNet_AM, @RiskNet_DER, @RiskQuantum
UK newspapers	@guardian, @Independent, @DailyMirror, @TheEconomist
US newspapers	@nytimes, @washingtonpost, @WSJ

it. The extraction of tweets requires specific criteria (*e.g.*, only English tweets, discarding re-tweets, and deleting hashtags and web links). Extracting by keywords leads to including also tweets from personal accounts, but it is useful to obtain much more relevant tweets, and even earlier than the financial accounts for particular events (*e.g.*, damages for earthquake or extreme weather conditions). Considering that the free API was dismissed on July 12th, 2023 (when Twitter became X), the available data sets are:

- From May 5th to July 12th for keyword related tweets.
- From May 11th to July 11th for account related tweets.

3.5 Workflow for OpRisk data analysis

This section describes the workflow applied to OpRisk event descriptions.

3.5.1 Descriptions cleaning

Descriptions have to be prepared for the analysis using some cleaning procedures. The set of all descriptions (or documents) to be analyzed is called a "corpus". Procedures to clean texts include the following ones:

- Data anonymization: to retrieve and delete (or substitute with conventional tags) any personal information and dates from texts, for compliance with GDPR (European Parliament and Council of the European Union, 2016) and for analytical purposes (Francopoulo and Schaub, 2020). Data anonymization is already guaranteed in the OpRisk data by the rules of the data collection (in all OpRisk event descriptions, ORX News, and scenario analysis).

- Languages detection: OpRisk event descriptions can be written in different languages. The same description may include parts written in different languages, and therefore we should split each description into several parts to detect the language of each part. The split can be performed considering specific characters as separators (*e.g.*, "." or ";"), or the carriage return. Afterwards, we can apply different strategies:

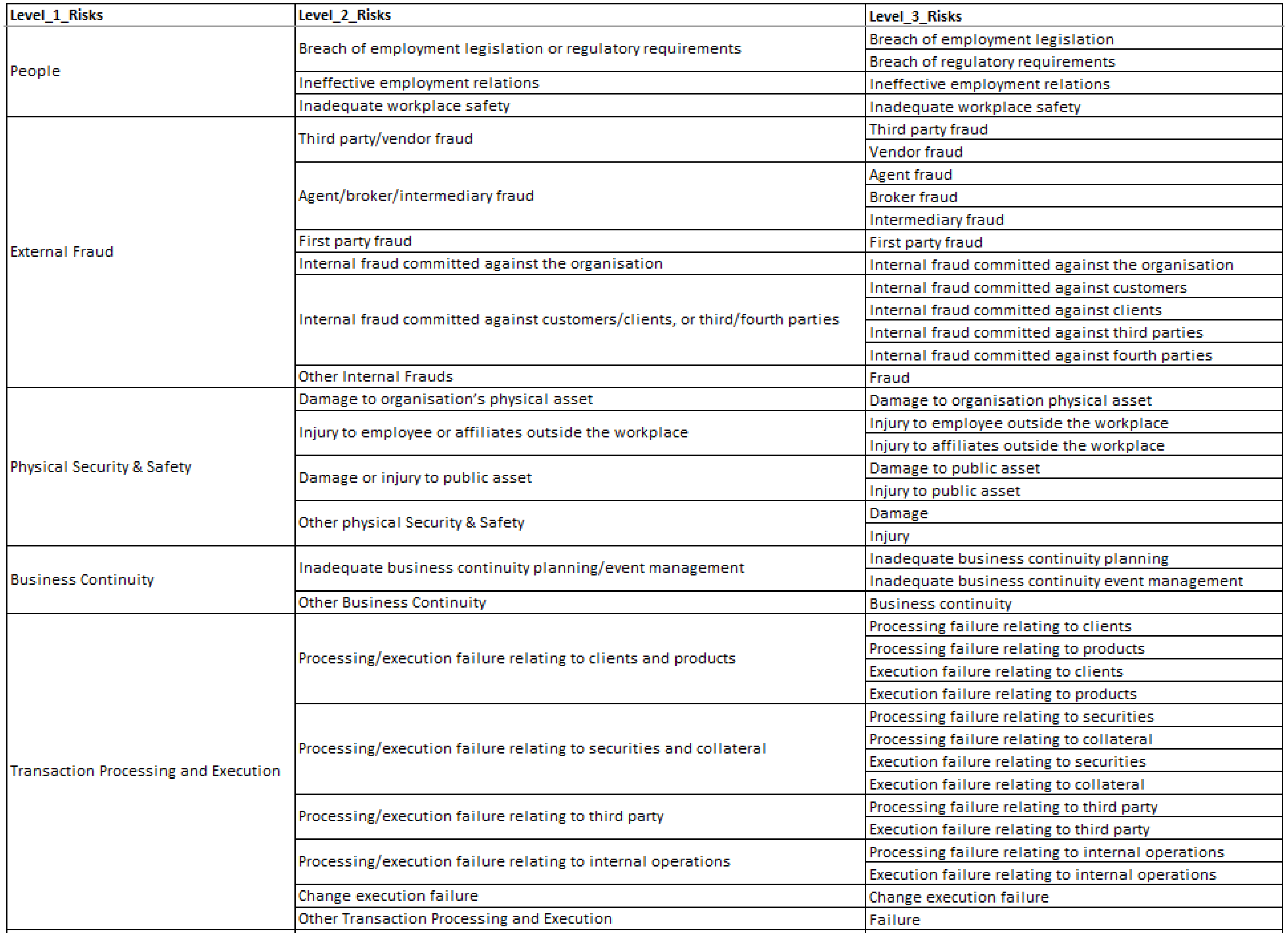
  1. We can select only the events having at least one part of the description written in English. All other events will be excluded from the analysis. This strategy is relatively simple and it guarantees an adequate quality of English descriptions, but it generally decreases the sample size.
  2. The parts of the descriptions detected as not written in English can be translated to English using an automated tool (*e.g.*, an API for Google Translate). This approach guarantees that we use the full data set in the analysis, but the quality of the English descriptions would be heavily affected by the accuracy of the translator tool. Moreover, there are currently no tools that are freely available to be applied to a significant amount of data. As far as we know, there is no available literature for this approach.

3. We can split the parts of descriptions, written in different languages, among different data sets. For each data set related to the main languages (*e.g.* English, Italian, German), we can apply the next steps of the analysis. This approach is consistent with the available literature, where it is usually suggested to apply text analysis techniques to data sets in a single language.

CoRep descriptions of the internal data should be, theoretically, written in English. For events with loss amounts lower than € 100,000, or booking dates earlier than 2018, different languages could appear. Descriptions reported for external data and scenario analysis are all written in English. With this evidence and the fact that, based on our research, there is no free automated tool to translate sentences into accurate English, we decided to adopt the first strategy among the three described above. Descriptions having at least a sentence written in English have been identified, using the function `detect_language` of the R package `cld2` (Ooms, 2022), which is an R wrapper on Compact Language Detector 2 (CLD2, Riesa and Giuliani, 2013), and then the English parts have been selected. CLD2 is a Naïve Bayesian classifier that probabilistically detects over 80 languages. The ORX taxonomy, which is currently composed of two levels (as described in Section 3.3.1 and Figure 3.1), has been considered in language detection, creating a third level (Figures 3.4, 3.5, and 3.6). To include some items in the third level that were not directly linked to anyone in the second level, some additional items (starting with "Other") have been included in the second level. If a sentence contains a character string corresponding to a Level_3_Risk of the ORX taxonomy, then it is forced to be detected as to be expressed in English.

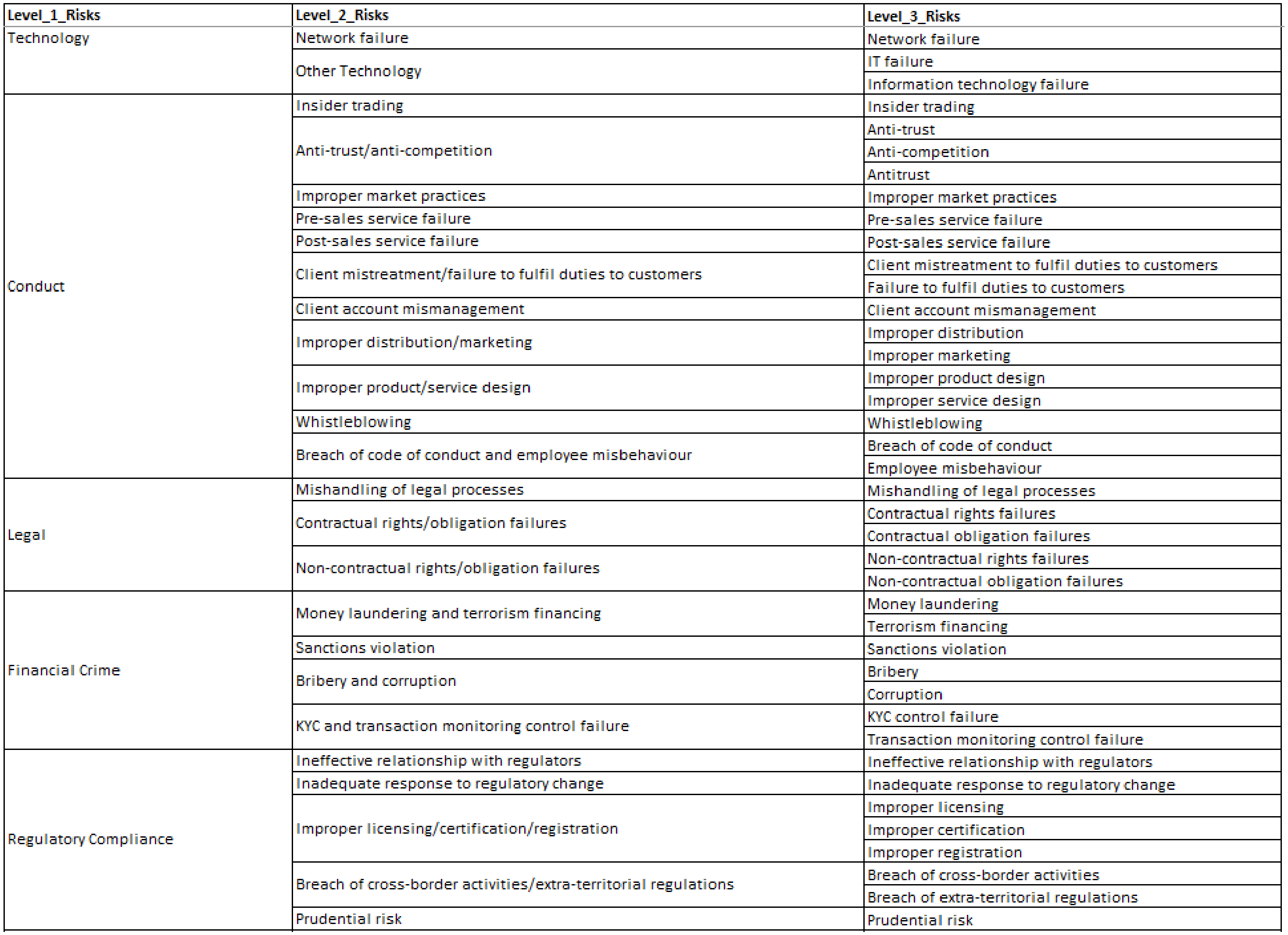
- Ignoring cases, which can be done by case-folding each letter into lowercase.
- Removing punctuations and digits.
- Removing frequent words called stop-words, that do not contain much information, like articles, pronouns, conjunctions, and words like "of", "about", "that", etc. Lists of stop-words are readily available for the main languages. In particular, for the application to be described in Section 3.7, the stop-word list, related to the English language, has been derived from meta::cpan (2021).

Table 3.4: ORX taxonomy Level 3 (part 1)



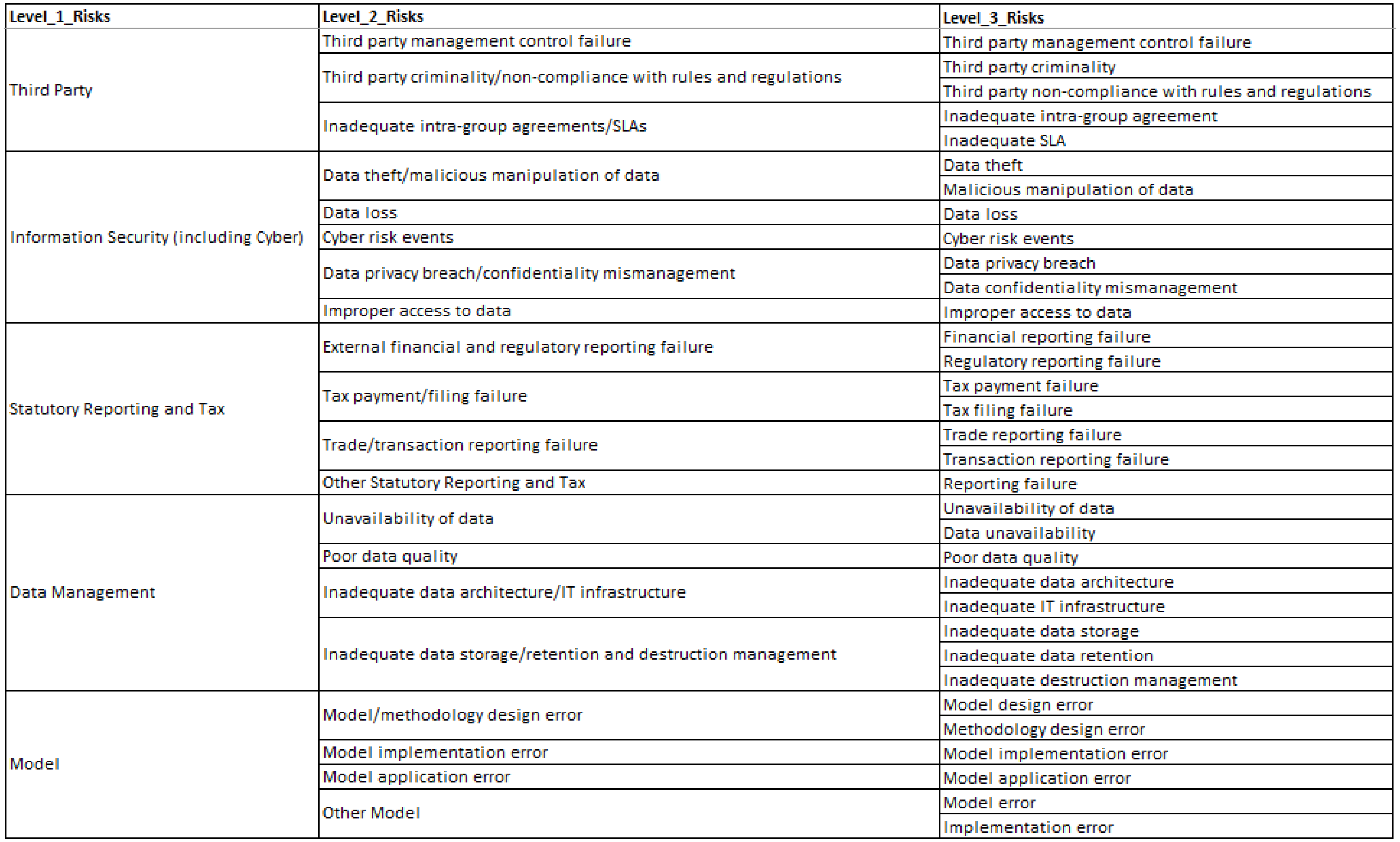
Level_1_Risks	Level_2_Risks	Level_3_Risks
People	Breach of employment legislation or regulatory requirements	Breach of employment legislation
		Breach of regulatory requirements
	Ineffective employment relations	Ineffective employment relations
	Inadequate workplace safety	Inadequate workplace safety
External Fraud	Third party/vendor fraud	Third party fraud
		Vendor fraud
	Agent/broker/intermediary fraud	Agent fraud
		Broker fraud
		Intermediary fraud
	First party fraud	First party fraud
	Internal fraud committed against the organisation	Internal fraud committed against the organisation
	Internal fraud committed against customers/clients, or third/fourth parties	Internal fraud committed against customers
		Internal fraud committed against clients
		Internal fraud committed against third parties
		Internal fraud committed against fourth parties
	Other Internal Frauds	Fraud
Physical Security & Safety	Damage to organisation's physical asset	Damage to organisation physical asset
	Injury to employee or affiliates outside the workplace	Injury to employee outside the workplace
		Injury to affiliates outside the workplace
	Damage or injury to public asset	Damage to public asset
		Injury to public asset
	Other physical Security & Safety	Damage
		Injury
Business Continuity	Inadequate business continuity planning/event management	Inadequate business continuity planning
		Inadequate business continuity event management
	Other Business Continuity	Business continuity
Transaction Processing and Execution	Processing/execution failure relating to clients and products	Processing failure relating to clients
		Processing failure relating to products
		Execution failure relating to clients
		Execution failure relating to products
	Processing/execution failure relating to securities and collateral	Processing failure relating to securities
		Processing failure relating to collateral
		Execution failure relating to securities
		Execution failure relating to collateral
	Processing/execution failure relating to third party	Processing failure relating to third party
		Execution failure relating to third party
	Processing/execution failure relating to internal operations	Processing failure relating to internal operations
		Execution failure relating to internal operations
	Change execution failure	Change execution failure
	Other Transaction Processing and Execution	Failure

Table 3.5: ORX taxonomy Level 3 (part 2)



Level_1_Risks	Level_2_Risks	Level_3_Risks
Technology	Network failure	Network failure
	Other Technology	IT failure
		Information technology failure
Conduct	Insider trading	Insider trading
	Anti-trust/anti-competition	Anti-trust
		Anti-competition
		Antitrust
	Improper market practices	Improper market practices
	Pre-sales service failure	Pre-sales service failure
	Post-sales service failure	Post-sales service failure
	Client mistreatment/failure to fulfil duties to customers	Client mistreatment to fulfil duties to customers
		Failure to fulfil duties to customers
	Client account mismanagement	Client account mismanagement
	Improper distribution/marketing	Improper distribution
		Improper marketing
	Improper product/service design	Improper product design
		Improper service design
	Whistleblowing	Whistleblowing
	Breach of code of conduct and employee misbehaviour	Breach of code of conduct
		Employee misbehaviour
Legal	Mishandling of legal processes	Mishandling of legal processes
	Contractual rights/obligation failures	Contractual rights failures
		Contractual obligation failures
	Non-contractual rights/obligation failures	Non-contractual rights failures
		Non-contractual obligation failures
Financial Crime	Money laundering and terrorism financing	Money laundering
		Terrorism financing
	Sanctions violation	Sanctions violation
	Bribery and corruption	Bribery
		Corruption
	KYC and transaction monitoring control failure	KYC control failure
		Transaction monitoring control failure
Regulatory Compliance	Ineffective relationship with regulators	Ineffective relationship with regulators
	Inadequate response to regulatory change	Inadequate response to regulatory change
	Improper licensing/certification/registration	Improper licensing
		Improper certification
		Improper registration
	Breach of cross-border activities/extra-territorial regulations	Breach of cross-border activities
		Breach of extra-territorial regulations
	Prudential risk	Prudential risk

Table 3.6: ORX taxonomy Level 3 (part 3)



Level_1_Risks	Level_2_Risks	Level_3_Risks
Third Party	Third party management control failure	Third party management control failure
	Third party criminality/non-compliance with rules and regulations	Third party criminality
		Third party non-compliance with rules and regulations
	Inadequate intra-group agreements/SLAs	Inadequate intra-group agreement
		Inadequate SLA
Information Security (including Cyber)	Data theft/malicious manipulation of data	Data theft
		Malicious manipulation of data
	Data loss	Data loss
	Cyber risk events	Cyber risk events
	Data privacy breach/confidentiality mismanagement	Data privacy breach
		Data confidentiality mismanagement
	Improper access to data	Improper access to data
Statutory Reporting and Tax	External financial and regulatory reporting failure	Financial reporting failure
		Regulatory reporting failure
	Tax payment/filing failure	Tax payment failure
		Tax filing failure
	Trade/transaction reporting failure	Trade reporting failure
		Transaction reporting failure
	Other Statutory Reporting and Tax	Reporting failure
Data Management	Unavailability of data	Unavailability of data
		Data unavailability
	Poor data quality	Poor data quality
	Inadequate data architecture/IT infrastructure	Inadequate data architecture
		Inadequate IT infrastructure
	Inadequate data storage/retention and destruction management	Inadequate data storage
		Inadequate data retention
		Inadequate destruction management
Model	Model/methodology design error	Model design error
		Methodology design error
	Model implementation error	Model implementation error
	Model application error	Model application error
	Other Model	Model error
		Implementation error

- Using regular expressions to detect special characters (*e.g.*, “ù”, “ä”, “|”, etc.) and remove them.

- Reducing words to their lemmas (*e.g.*, “pay” from “paying”, “client” from “clients”), substituting each word with the corresponding canonical form. The lemmatization reduces the number of distinct words in a text corpus and increases the frequency of occurrence for some of them. The lemmatization can be preferred to stemming since the latter usually returns trimmed words, which are not always easy to understand (*e.g.* “anatocism” would be converted to “anatoc”), whereas the former always returns complete words (Khyani and Siddhartha, 2021).

- *n*-gramming, which means considering a sequence of two or more words as a unique token since they have a specific meaning together (*e.g.*, “internal fraud”). The *n*-grams from the ORX taxonomy have been selected considering all the items in the Level_3_Risk of Figures 3.4, 3.5, and 3.6. For example, every time that the string “network failure” is observed within a description, the two words are substituted by the unique token “network_failure”. The methodology described by Frigau *et al.* (2021), based on the identification of word sequences appearing much more often than expected, has been applied to unveil relevant bigrams and trigrams. If two words have the probabilities p_1 and p_2 to occur, then the expected probability of the two words to co-occur together, under the independence hypothesis, is p_1p_2. For words having a specific meaning together (*e.g.*, “internal fraud”, “network failure”, etc.), it is plausible that they are significantly dependent and then co-occur with a much higher probability than p_1p_2. Therefore, the relevant bigrams and trigrams are identified as the ones that satisfy both of the following conditions:

  1. There are at least 5 co-occurrences in the corpus.
  2. The standard binomial test, applied to the probability that the observed proportion of co-occurrences exceeds the expected value under independence, has a p-value lower than 0.005.

  This lower-than-usual significance level is based on the following rationales:

  – The independence hypothesis is not fully satisfied for English writing, which could lead to selecting several n-grams that are not relevant. Such significance level favors the selection of n-grams that are meaningful as phrases.

  – Although if it is not explicitly stated by Frigau *et al.* (2021), the low significance level could have been chosen to compensate for the fact that test multiplicity has not been considered within this approach. Applying the method that includes multiplicity, a statistical test is performed on each n-tuple (for n-grams) of consecutive words, where one adopts a more standard significance level, *e.g.* 0.05, and then adjusts it to take the multiplicity into account, using, *e.g.*, Bonferroni correction or the more accurate Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995).

  The n-grams with $n > 3$ have not been considered within this methodology since the most relevant ones are already included in the ORX taxonomy, and to also avoid a further increase of the computational burden.

- Removing all the terms (*i.e.*, tokens) having a total frequency lower than five, considering all the descriptions.

3.5.2 Text vectorization - Bag-of-Words (BoW)

According to the BoW approach (Harris, 1954), the data set is transformed into a document-by-term matrix, where each row represents a document (*e.g.*, an event description), each column represents a term (*e.g.*, a word or an n-gram), and each cell represents the Term Frequency (TF), *i.e.*, the number of times each term appears in each document (Luhn 1957). Di Vincenzo *et al.* (2023) describe in detail the application of the BoW approach in the OpRisk context.

A common approach used in text analysis is known as the "Term Frequency – Inverse Document Frequency" (TF-IDF) method, where each element of the document-by-term is multiplied by the Inverse Document Frequency (Spärck Jones, 1972), which is a scoring of how rare a word is across the documents. Since the analyzed descriptions are short texts (having max 250 characters for OpRisk event descriptions), we do not apply the IDF scaling, as motivated by Anderlucci *et al.* (2019) for their application on similarly structured data. Moreover, as pointed out by Di Vincenzo *et al.* (2023), the main purpose of the IDF scaling is to reduce the weight of terms that are more often used in many documents. IDF assumes that if a word is used in many descriptions, then it is not informative, and neither useful to discriminate different clusters of data. However, most of the non-informative terms have been already excluded by removing the stop-words from the text corpus. Therefore, by applying the IDF scaling to short texts, we risk reducing the weights of some informative words that could characterize the clusters.

The similarity between documents can be evaluated using the "cosine similarity" (Singhal, 2001). Considering the documents d_s and d_t, $s,t \in \{1,\dots,n\}$, represented by the m-length vectors

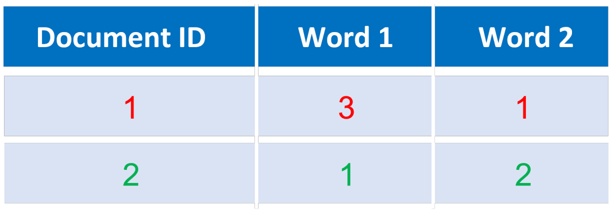
$$\boldsymbol{x}_s = (TF(w_1,d_s),\dots,TF(w_m,d_s)) \text{ and } \boldsymbol{x}_t = (TF(w_1,d_t),\dots,TF(w_m,d_t)),$$

where each component represents the TF of a word of the dictionary (the dictionary size being m), their cosine similarity is given by the cosine of the angle between the two vectors representing the two descriptions:

$$\mathrm{CS}(d_s,d_t) = \frac{\boldsymbol{x}_s \cdot \boldsymbol{x}_t}{\|\boldsymbol{x}_s\|\|\boldsymbol{x}_t\|} \in [0,1].$$

In the following example, two documents are composed of two different words with a TF matrix reported in Figure 3.7 and then represented by the vectors in Figure 3.2.

Table 3.7: TF matrix of the documents d_1 and d_2.



Document ID	Word 1	Word 2
1	3	1
2	1	2

The cosine similarity between the documents d_1 and d_2 is then obtained as follows:

$$\mathrm{CS}(d_1,d_2) = \cos(\theta) = \frac{\boldsymbol{x}_1 \cdot \boldsymbol{x}_2}{\|\boldsymbol{x}_1\|\|\boldsymbol{x}_2\|} = \frac{(3,1)\cdot(1,2)}{\sqrt{3^2+1^2}\sqrt{1^2+2^2}} = \frac{5}{\sqrt{10}\sqrt{5}} = \frac{1}{\sqrt{2}} \cong 0.71.$$

3.5.3 Semantic adjustment

The BoW approach alone cannot capture the semantic similarity between synonyms: even if two documents are almost identical in terms of meaning, their similarity, based on TF, could be low due to scarce word matching. This issue in the OpRisk context has been extensively discussed by

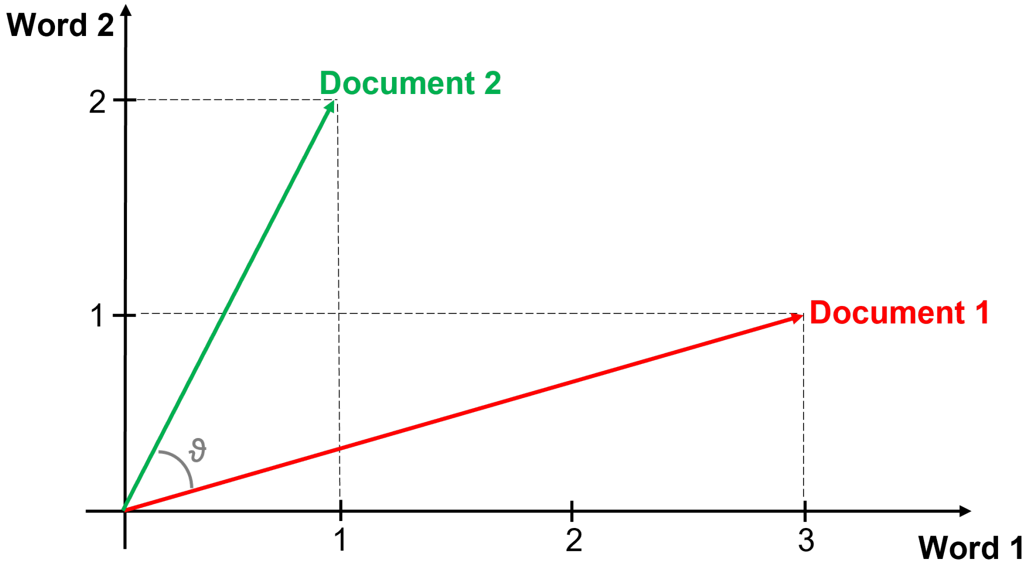


Figure 3.2: Vectors representation of the documents d_1 and d_2.

Di Vincenzo *et al.* (2023).

To consider semantic similarity, an adjustment can be applied to the document-by-term matrix using word embedding techniques, a class of methods allowing to represent words in a vector space so that different words that share a common concept are "close" as measured by the cosine similarity.

One of the most classical and firstly defined word embedding is `word2vec` (Mikolov *et al.*, 2013), composed of two different methods based on neural networks (Figure 3.3):

- Continuous Bag-of-Words model (CBOW), which creates a sliding window around the current word, to predict it from "context", *i.e.*, the surrounding words. After training, the vectors representing the words are obtained.

- Skip-gram model, instead of predicting one word each time, uses one word to predict all surrounding words ("context"). In general, Skip-gram is much slower than CBOW, but it is considered to be more accurate with infrequent words.

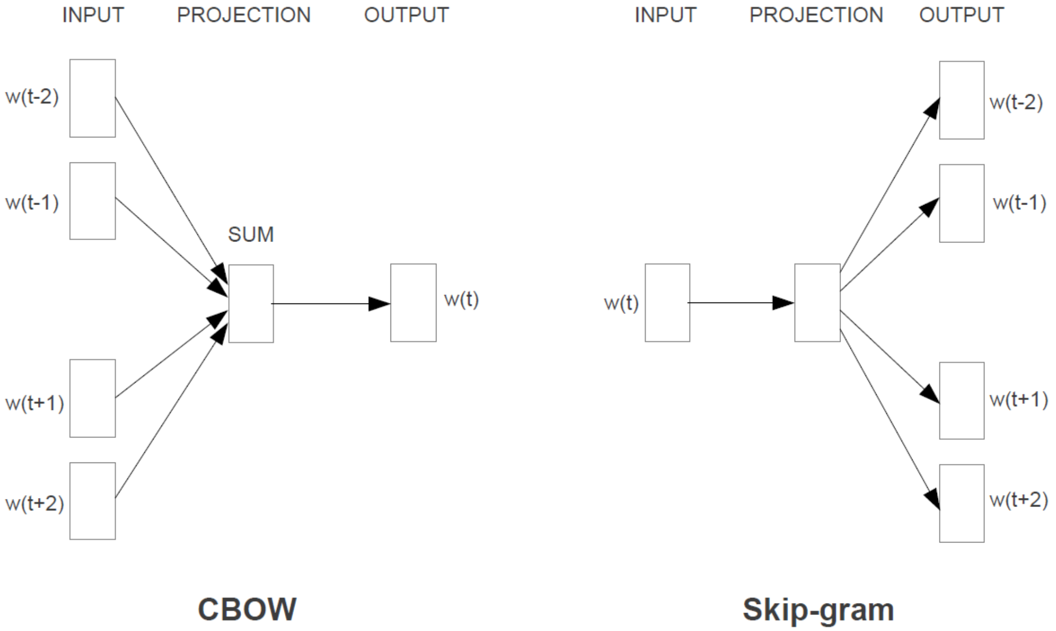


Figure 3.3: Word2vec model architectures (Mikolov *et al.*, 2013): the CBOW architecture predicts the current word based on the context, and the Skip-gram predicts the surrounding words given the current word.

Some alternative word embedding methods are:

- `GloVe` (Global Vectors, Pennington *et al.*, 2014) is a log-bilinear regression model for unsupervised learning of word representations.
- `BERT` (Bidirectional Encoder Representations from Transformers) (Devlin *et al.*, 2019) where each word has a different vector representation according to the different context.
- `FastText` (Bojanowski *et al.*, 2017) which is trained on syllabication, instead of words. Therefore, `fastText` can also represent words that are not present in the dictionary of the training set.

After choosing the word embedding to be used, we can opt for:

- Pre-trained word embeddings, trained on a data set and available on the web. There are several pre-trained word embedding approaches (*e.g.*, `word2vec`, `fastText`, `GloVe`) trained on different corpora (*e.g.*, Wikipedia dumps, CoNLL17, Google News). It would be useful to compare different word embedding approaches to select the most compliant with the purposes of the analysis. For example, we may want to consider a set of highly significant words in the OpRisk context and check whether the most similar terms according to the cosine similarity are appropriate ones. Examples of pre-trained word embeddings are available at the *NLPL word embeddings repository* (2017). In the case of pre-trained word embeddings, specific terms of the analyzed field may not be available (*e.g.*, "anatocism") due to the peculiarity of the OpRisk dictionaries.

- Training word embeddings on the available data set. When we have a large data set, it would be convenient to use it to train the word embeddings. In this case, we obtain field-specific word embeddings, where the specific terms are also included. Since we generated the word embeddings, it is even more important to assess their accuracy. The same method, explained above to compare different pre-trained embedding methods, can be applied to assess the quality of the newly trained word embeddings.

Further methods for word embedding evaluation are described by Wang *et al.* (2019) and Giabelli *et al.* (2022). As introduced in Bakarov (2018), there are two main categories for evaluation methods:

- Extrinsic evaluators consider different word embedding methods (or different settings of the same method) as inputs for downstream tasks and measure changes in performance metrics specific to that task (*e.g.*, accuracy for supervised tasks, or perplexity for unsupervised ones). It is worth mentioning that such evaluators are required to repeat the downstream tasks, *i.e.*, the overall calculation procedures for each different word embedding, multiplying the computational burden by the number of tested methods.

- Intrinsic evaluators test the quality of word embeddings independently of the considered tasks. These evaluators are based on experiments in which word embedding methods are compared with human judgments on word relations. Predefined sets of words are often used to get human assessments, and then these assessments are compared with word embeddings. The main issue for these evaluators is that the available sets of words mainly include generic terms without containing the ones related to specific tasks (such as the case of the OpRisk event descriptions analysis).

Once a word embedding (pre-trained or not) has been selected, the cosine similarity between the vectors representing the terms can be used as a measure of semantic similarity between the terms themselves. It is possible to calculate a word-similarity matrix, which includes the cosine similarity between all the possible couples of terms within the dictionary.

The word similarity matrix allows updating the value of each "zero" of document-by-term matrix with the value of the most similar word included in the same row of the matrix and scaled by the respective word similarity score (Shanavas *et al.*, 2021).

Similarly to Shanavas *et al.* (2021), we use a similarity matrix that only contains similarity values higher than 0.8 to avoid including noise (*i.e.*, medium-low similarity due more to randomness than similar meaning) in the semantic adjustment. Some rationales for the selection of threshold 0.8 are discussed by Di Vincenzo *et al.* (2023).

3.5.4 Dimensionality reduction

After obtaining a document-by-term matrix that considers the semantic similarity between terms, some tools are needed to uncover the possible clusters with related root causes. Therefore, it is useful to represent the data set in a chart by reducing it to two (or maximum to three) dimensions. In this context, dimensionality reduction methods are used to map document vectors from the word space to a space whose number of dimensions is user-defined. For example, Di Vincenzo *et al.* (2023) made use of the Latent Semantic Analysis (LSA) (Dumais *et al.*, 1988) to reduce the data to two dimensions and represent them graphically. The most classical technique for dimensionality reductions is the Principal Components Analysis (PCA) (Pearson, 1901; Hotelling, 1933). PCA and LSA make use of matrix factorization to reduce the dimensionality, whereas most recent methods are based on the data neighborhood, such as:

- t-SNE (t-distributed Stochastic Neighbor Embedding) introduced by Hinton and Roweis (2002), and Van der Maaten and Hinton (2008),
- UMAP (Uniform Manifold Approximation and Projection) proposed by McInnes *et al.* (2018).

The t-SNE method computes the probability that pairs of data points in the high-dimensional space are related and then chooses low-dimensional embeddings that produce a similar distribution. The

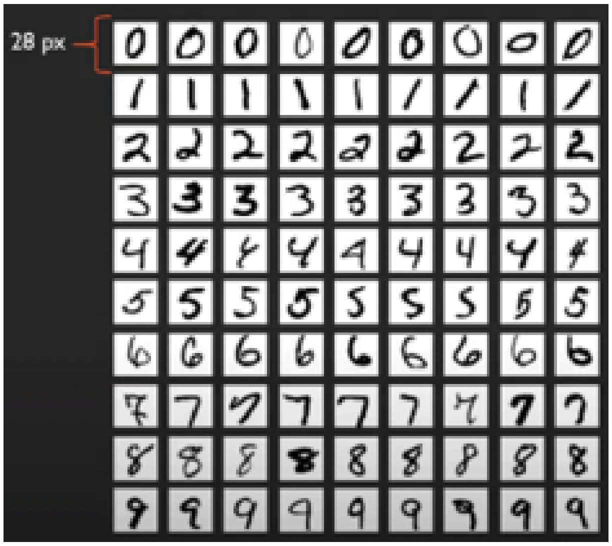


Figure 3.4: A sample from the MNIST data set.

UMAP algorithm is similar to t-SNE for visualization quality but preserves more of the global structure with a lower computational burden. The good performance of UMAP can be appreciated, for instance, when applied on MNIST, a data set of 28×28 pixel grayscale images of handwritten digits. A sample of MNIST is represented in Figure 3.4. There are 10 classes of handwritten digits (0 through 9) of 70,000 total images, 10-digit each being a 784-dimensional vector. The bidimensional UMAP representation of the MNIST data set in Figure 3.5 shows how this projection is able to separate the digits in MNIST. The UMAP algorithm takes the following hyperparameters:

- `n_neighbors`, the number of neighbors to consider when approximating the local metric. The size of local neighborhood (in terms of number of neighboring sample points) used for manifold approximation. Larger values result in more global views of the manifold, while smaller values result in more preserved local data structures. In general, values should be in the range 2 to 100.

- `min_dist`, the desired separation between close points in the embedding. Smaller values generate a more clustered/clumped embedding where nearby points on the manifold are drawn closer together, while larger values result in a more even dispersal of points.

The UMAP representation in Figure 3.5 has been obtained setting `n_neighbors`=10 and `min_dist`=0.001. The impact of varying such hyperparameters can be appreciated in Figure 3.6.

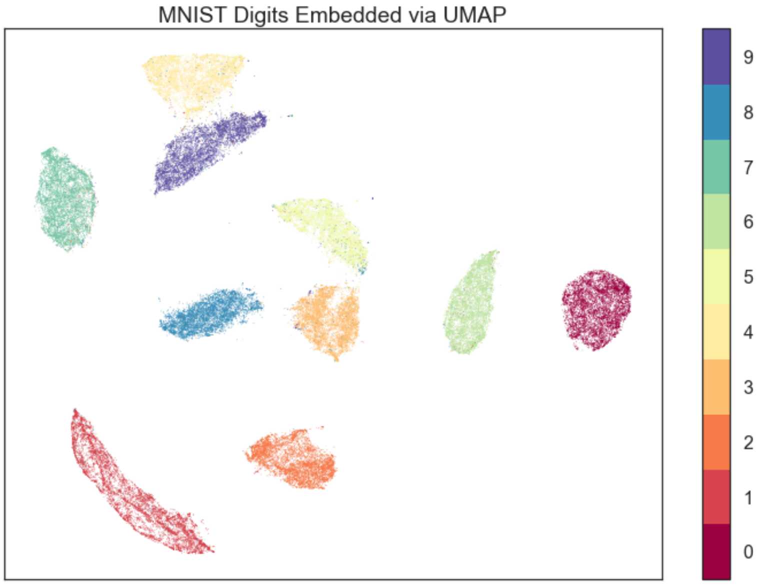


Figure 3.5: A bidimensional UMAP representation of the MNIST data set.

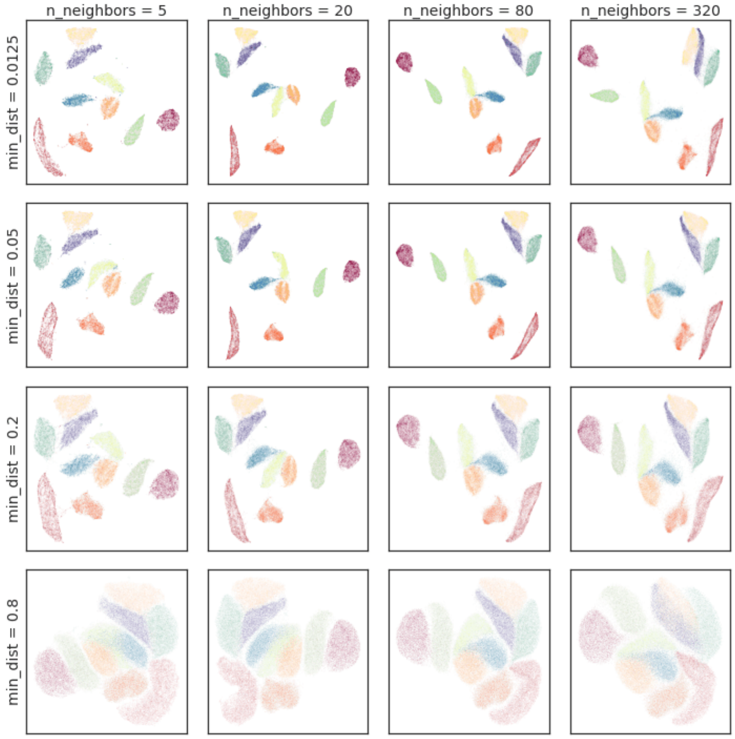


Figure 3.6: Sensitivity of the UMAP representation applied to the MNIST data set (from McInnes *et al.*, 2020) to the choice of hyperparameter values `min_dist` and `n_neighbors`.

UMAP is implemented in the function `umap` of the R package `uwot` (Melville, 2023). As reported in the help of the function `umap`, in the case of data sets with more than 100 columns, it is suggested to reduce the dimensions using PCA, before applying the UMAP, to reduce the computational burden. Moreover, the help of `umap` reports that 50 dimensions are recommended in many t-SNE applications, implicitly suggesting this setting for the UMAP ones. For example, in their applications, Van der Maaten and Hinton (2008) reduce the data set to 30 dimensions, before applying t-SNE.

Through a 2D UMAP representation, it is possible to examine the data using interactive charts produced via the R package `plotly` (Sievert, 2020), which allows to visualize the event description corresponding to each point in the chart, just moving the cursor over it. In particular, in case a set of points in the chart appears to form a cluster (as it occurs in Figure 3.5 for points representing the same digit), we can verify if the points within the cluster (or most of them) are related to a common root cause (*e.g.*, a particular type of fraud, sanction, cyberattack, etc.).

This qualitative analysis can identify the most evident drivers of the data, but (apart from peculiar cases such as Figure 3.5) it is not sufficient to obtain a complete clustering. In particular, when we have several thousands of data, we need a method to assign each data point to a single cluster (*i.e.*, to perform a hard partitioning) and to provide some hints about the content of each cluster. These needs can be accomplished by using topic modeling techniques (to be described in Section 3.5.5), which are usually applied for unsupervised analyses of textual data.

3.5.5 Cluster selection

OpRisk event descriptions can be clustered using classical clustering techniques (*e.g.*, k-means) or topic modelling techniques (Di Vincenzo *et al.*, 2023). The main benefit of the latter techniques, if compared with clustering ones is that they allow for specifying the subjects of the topics to define clusters. One of the most used topic models is Latent Dirichlet Allocation (LDA), which was introduced by Blei *et al.* (2003) in the context of textual analysis. LDA provided the highest accuracy, if compared with other topic models, when applied to a first set of OpRisk event descriptions in Di Vincenzo *et al.* (2023).

LDA is a generative statistical model that explains a set of observations through unobserved (*i.e.*, latent) groups. It assumes that the words (or tokens) in a document are drawn from K topics, and a probability distribution over the available words characterizes each topic.

The tokens of the i^{th} document are supposed to be drawn independently from the k^{th} topic with probability π_{ik}. The distributions corresponding to the topics, and the probabilities $\pi_{i1},\dots,\pi_{iK}$ for all documents, can be estimated via Monte Carlo Markov Chains (MCMC), implemented through Gibbs sampling. In the present application, a document corresponds to an OpRisk event description.

- The classical LDA has a prior Poisson distribution over the number of topics that appear in the corpus but, in practice, the number of topics is fixed to a maximum value to provide interpretable topics (as explained by Frigau *et al.*, 2021).

- The specification of the generative model starts by assuming a Dirichlet distribution with parameter α over the N tokens, and making K draws ϕ_k from such distribution. The draws determine the probabilities that each of the K topics assign to each token, so we have K vectors of length N. The subject of each topic can be inferred from the tokens with a higher probability within the topic. Graphical representations, such as word clouds, can be used for this purpose.

- For the i^{th} document, one draws θ_i from a second Dirichlet distribution with parameter β. This θ_i has length K and determines the extent to which document i participates in each of the K topics. Each document is supposed to contain a certain number of topics with different corresponding probabilities. Supposing that the identified topics are used to define clusters, the clustering can be performed by assigning each document to the topic showing the highest probability within the document.

- To generate tokens in the i^{th} document, one first draws z_j from a one-trial multinomial with parameter θ_i to pick the topic that generates a token. Suppose it is topic k. Then one draws from a one-trial multinomial with the parameter ϕ_k to determine which token within that topic is chosen for the document.

Using this generative model, MCMC is applied to obtain estimates of the K topic distributions and the probability with which each document participates in each topic. As diagnostics, the convergence of MCMC can be assessed through a trace plot of the log-likelihood (Chakrabarti *et al.*, 2023), while the fit can be evaluated based on the interpretability of the topics and quantified via perplexity (Blei *et al.*, 2003).

The classical LDA allows one to discover topics automatically. However, some topics in the data set may be already known or guessed, for example, using the UMAP representation mentioned in Section 3.5.4. Therefore, it is useful to "inform" the LDA that some topics, defined by the corresponding sets of tokens, are expected to be found in the model results, as in the seeded LDA (Jagarlamudi *et al.*, 2012). Seeded LDA enforces some topics to have positive probability only for a restricted set of tokens, *i.e.*, the seed tokens (Frigau *et al.*, 2021). Similarly to LDA, the seeded LDA can automatically discover new topics, by letting a certain number of topics remain unseeded. Seeded LSA is here applied to the rounded semantic-aware document-by-term matrix. Rounding is included because seeded LDA requires count data as input.

To improve the accuracy of this method, we included two additional brand-new features to the seeded LDA:

- We observed that, in general, two or more documents composed of the same tokens can show slightly different probability distributions among topics. This is due to the "generative" nature of the LDA statistical model and could lead to assigning equal documents to different clusters. To prevent this unwanted configuration, we propose to "average" the topic probability distributions of documents composed of the same tokens. In practice, if two or more documents are represented by totally equal rows in the semantic-aware document-by-term matrix, then their topic distributions are all replaced by their average. For example, suppose we have two OpRisk event descriptions composed only (after descriptions cleaning) of the token "fraud" to which the LDA assigns probabilities to three different topics, *e.g.*, $\pi_1 = (0.3, 0.4, 0.3)$ and $\pi_2 = (0.5, 0.3, 0.2)$. Note that even if the two descriptions are equal, the clustering based on the highest probability would assign description 1 to the cluster defined by topic 2, and description 2 to the cluster defined by topic 1. Of course, this cannot be considered an ideal configuration for the results that we are going to obtain. Averaging topic probabilities, we would substitute π_1 and π_2 with their average $\tilde{\pi}_1 = \tilde{\pi}_2 = (0.4, 0.35, 0.25)$. Based on $\tilde{\pi}_1$ and $\tilde{\pi}_2$, both descriptions can be assigned to the cluster represented by topic 1. Therefore, we can ensure that equal descriptions have the same topic probability distribution and, in particular, that they are all assigned to the same cluster.

- In seeded LDA, the seed tokens specifying a seeded topic do not have a positive probability for other seeded topics (unless they have been defined as seed tokens for more topics), but they can have non-negligible probabilities for some unseeded topics. For this reason, especially for longer documents, it happens that a document containing one or more seed tokens presents higher probabilities for unseeded topics. This is an unwanted configuration for our purposes since we consider the presence of seed tokens as a strong signal that the document is related to the corresponding seeded topic. For this reason, instead of assigning each document to the cluster related to the topic with the highest probability, in case a document contains one or more seed tokens, we propose constraining this selection to the topics related to the included seed tokens. This new feature prevents a document, containing seed words related to seeded topics, is then assigned to a cluster that is related to an unseeded topic.

Another method to qualitatively assess the results of seeded LDA, apart from the ones already mentioned above for the LDA, is the representation of the clustered data points in the UMAP bidimensional chart. Whenever the points representing a distinct cluster in the chart are all assigned to the same topic and, in the case of a seeded topic, the assigned descriptions all contain the related seed tokens, this is a clear sign that the results are consistent and adequate. LDA and seeded LDA are implemented in the R package `topicmodels` (Grün and Hornik, 2011; Grün and Hornik, 2023).

3.6 Workflow for tweet data analysis

This workflow mirrors the one presented in Section 3.5 for OpRisk event descriptions, including the adaptations required by tweet data. Therefore, the next sections just focus on the differences if compared with the previous workflow. Considering the amount of extracted tweets (around 100,000-150,000 per day), we decided to separately analyze each daily data set of tweets. Therefore, all the steps described within this section are applied to each daily data set of tweets.

3.6.1 Tweet cleaning

Procedures to clean tweets include the following steps:

- Data anonymization: a pre-defined list of most known names is excluded from tweets. The list of names was obtained from the R package `gender` (Mullen, 2021).
- Languages detection: it is possible to directly select the English-written tweets since the functions `search_tweets` and `get_timeline` (used for tweets extraction, as described in Section 3.4) allow to specify the argument `lang='en'`.
- Hashtags and web links are removed.
- The following steps have been applied as already done for OpRisk event descriptions: ignoring cases, removing punctuations and digits, stop-words, and special characters, and reducing words to their lemmas.
- Removing duplicated tweets.
- Considering *n*-gramming: apart from *n*-grams from ORX taxonomy and bigrams-trigrams already identified within OpRisk event descriptions, also bigrams-trigrams identified within the tweets have been included.
- As already done for OpRisk event descriptions, removing all the terms having a total frequency lower than five, considering all the tweets.

3.6.2 Tweet vectorization (BoW) and semantic adjustment

For OpRisk event descriptions, according to the BoW approach, the tweets data set is transformed into a document-by-term matrix. Here each row represents a document (*i.e.*, a tweet), each column represents a term (*i.e.*, a word or an *n*-gram), and each cell represents the Term Frequency (TF).

Considering the same word embedding selected for OpRisk event descriptions, and the related word-similarity matrix similarity (with values higher than 0.8), the value of each "zero" of the document-by-term matrix is updated with the value of the most similar word included in the same row of the matrix and scaled by the respective word similarity score.

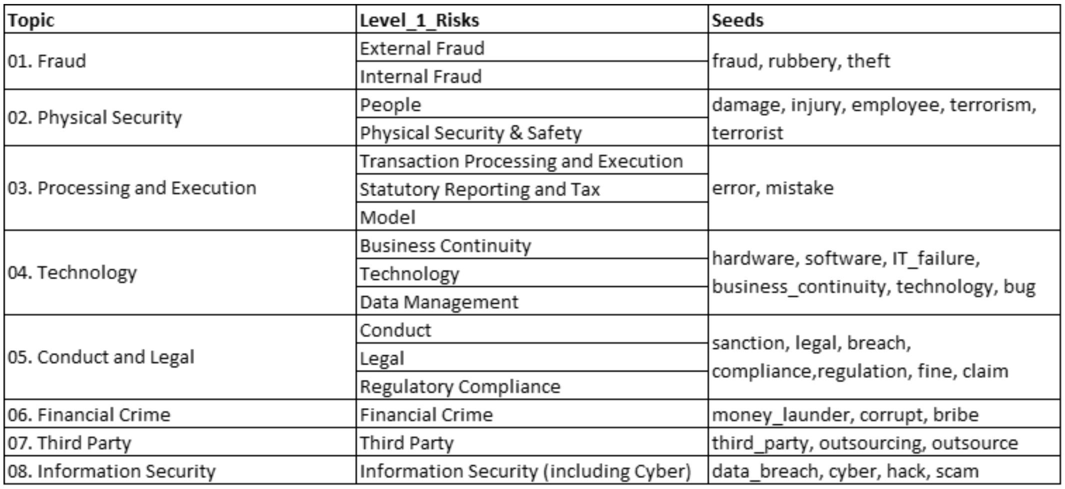
3.6.3 Dimensionality reduction, cluster selection, topics analysis, and emerging topics detection

As for OpRisk event descriptions, we use UMAP for producing the bidimensional data representations, and seeded LDA to identify the topic probability distributions and, consequently, the clustering of the tweets.

Since the tweets are analyzed day by day, it is important (more than checking each specific daily set of topics) to identify a common list of topics, to be able to compare the number of tweets clustered in the topics on different days. This setting allows us to observe the tweet daily frequencies for each topic, and to identify possible peaks in this time series, which can represent particular OpRisk events affecting the financial system, the industry, or the governments. For each topic, as an explorative method to detect the possible peaks, we calculate the 95% quantile of the normal distribution estimated on the time series of tweets, and then we deep dive into all the daily data sets exceeding that value.

We can define the seeded topics based on the ORX taxonomy, aggregating some similar levels of the Level_1_Risks, and assigning the corresponding seed tokens accordingly. Considering all the 16 Level_1_Risks would be a too granular specification, so we made a selection and the list of seeded topics with the corresponding Level_1_Risks and seed tokens is reported in Table 3.8.

Table 3.8: Tweet topics based on ORX taxonomy with related seed tokens.



Topic	Level_1_Risks	Seeds
01. Fraud	External Fraud	fraud, rubbery, theft
	Internal Fraud	
02. Physical Security	People	damage, injury, employee, terrorism, terrorist
	Physical Security & Safety	
03. Processing and Execution	Transaction Processing and Execution	error, mistake
	Statutory Reporting and Tax	
	Model	
04. Technology	Business Continuity	hardware, software, IT_failure, business_continuity, technology, bug
	Technology	
	Data Management	
05. Conduct and Legal	Conduct	sanction, legal, breach, compliance,regulation, fine, claim
	Legal	
	Regulatory Compliance	
06. Financial Crime	Financial Crime	money_launder, corrupt, bribe
07. Third Party	Third Party	third_party, outsourcing, outsource
08. Information Security	Information Security (including Cyber)	data_breach, cyber, hack, scam

Since the tweets can contain several variants of the adopted seed tokens, we aggregated with each seed all the tokens that contain that seed. For example, “cyber_attack” is aggregated to the seed “cyber”, while “hacking” is aggregated to the seed “hack”. This action is performed for the following main reasons:

- it is very difficult to include all the relevant variants of the seed tokens;
- even if we include all the relevant variants, then the weight of each seed token will decrease within the related topic (so losing the relative weight difference between seed tokens and non-seed tokens);
- this aggregation reduces the dimensionality, decreasing the number of columns of the document-by-term matrix.

To detect emerging OpRisk related topics, we consider five unseeded topics in seeded LDA (we did not find any significantly different evidence varying this number). Observing the related word clouds, we can detect OpRisk related topics and deep dive into the corresponding tweets to understand if such topics are relevant as an early warning for the financial institution. Since OpRisk analysts cannot verify every single tweet, we aim to detect the signal related to many tweets related to a specific topic, including the most frequent words that are represented by word clouds. For each day, it is much easier and faster to look at a few word clouds to spot some OpRisk related words among the most frequent ones, than reading all the tweets or websites reporting financial news.

3.7 Application to OpRisk data

This application analyzes the CoRep descriptions of the OpRisk data set for UniCredit banking group using the approaches described in the previous sections. The CoRep is the Common Reporting, which is the set of all data that all financial institutions have to periodically report to their Supervisory Authorities (*e.g.*, European Central Bank). Among the CoRep reports, there is the C17.02 template, which reports information (including the description) on significant OpRisk events. The analyzed data set is composed of the OpRisk data booked between 2005 and 2022 leading to gross loss amounts higher than or equal to € 1000. Each record of this data set represents an OpRisk event, where the related CoRep description is a text field reporting an anonymized description, having a maximum of 250 characters. The obtained data set includes 227,338 OpRisk events. This data set is separately analyzed for each event type.

The analysis is performed using the R packages `quanteda` (Benoit *et al.*, 2018), `word2vec` (Wijffels, 2021), and the ones mentioned in Section 3.5.

First of all, the descriptions are cleaned as described in Section 3.5.1. The events with at least one English-written sentence have been selected, but there is no need for data anonymization since such descriptions are all entered without any personal information. After these steps, the data set has been reduced to 65,974 OpRisk events for CoRep descriptions, and 75,772 events for chronological descriptions. The stop-word list has been obtained through the R package `stopword` (Benoit *et al.*, 2021). The *n*-grams coming from the ORX taxonomy, and the relevant bigrams and trigrams have been included.

While we use the chronological descriptions in a later stage to train the word embedding, we consider the CoRep descriptions to obtain the document-by-term matrix. After having deleted the rows with all zero values, we obtain a document-by-term matrix having 65,032 rows (*i.e.*, the number of descriptions) and 8,535 columns (*i.e.*, the length of the dictionary consisting of all the unique tokens included in the cleaned descriptions).

Note that this data set is considerably more challenging if compared to the previous work by Di Vincenzo *et al.* (2023) because it is much larger (around 100 times larger in terms of rows, and 5 times larger in terms of columns) and not all the descriptions are written in English.

The next step is to generate the semantic-aware document-by-term matrix using the approach described in Section 3.5.3. We trained the word embedding `word2vec` on UniCredit OpRisk event chronological descriptions, ORX News digest text descriptions (around 10,000 data), and storylines of the scenario analysis (around 350 data). We trained both CBOW and Skip-gram models, and assessed them as follows:

- We defined a set of words that are particularly significant in the context of OpRisk in financial institutions. In particular, we selected “bank”, “client”, “anatocism”, and “legal”.
- For each selected word, based on the estimated embedding, we extracted the five most similar words, *i.e.*, the terms showing the highest cosine similarity with the selected word.
- For each selected word, we qualitatively checked if these five words are semantically similar to the selected word itself.

The results are summarized in Tables 3.9 and 3.10. Based on these results, CBOW outperforms

Table 3.9: Most similar words based on CBOW.

Term 1	Term 2	Rank
bank	central_bank	1
bank	national_bank	2
bank	rabobank	3
bank	institution	4
bank	raiffeisen	5
client	client_bank	1
client	client_account	2
client	retail_client	3
client	account_holder	4
client	client_inform	5
anatocism	overdraft_interest_rate	1
anatocism	account_open_start	2
anatocism	overdraft_fee_interest	3
anatocism	rapporti	4
anatocism	account_open_close	5
legal	litigation	1
legal	bring_legal	2
legal	legal_procedure	3
legal	statute	4
legal	legal_action	5

Table 3.10: Most similar words based on Skip-gram.

Term 1	Term 2	Rank
bank	include	1
bank	believe	2
bank	stanchart	3
bank	say	4
bank	techcombank	5
client	provide	1
client	account	2
client	less	3
client	investment	4
client	around	5
anatocism	overdraft_interest_rate	1
anatocism	credit_sentenza_trib	2
anatocism	small_medium_legal	3
anatocism	estinto	4
anatocism	usury_conto_corrente	5
legal	claim	1
legal	allege	2
legal	accuse	3
legal	state	4
legal	action	5

Skip-gram for the selected words “bank” and “client”, while the two embedding approaches appear to be approximately equivalent for the selected terms “anatocism” and “legal”. For this reason, we opted to choose CBOW for the semantic adjustment. The selected word embedding allows us to obtain the word similarity matrix, and then to adjust the document-by-term matrix, as described in Section 3.5.3. Similarly to Shanavas *et al.* (2021) and Di Vincenzo *et al.* (2023), we use a similarity matrix that only contains similarity values higher than 0.8 to avoid including noise (*i.e.*, medium-low similarity due more to randomness than similar meaning) into the semantic adjustment.

We report the results of the event type “Clients, Products & Business Practices”, which includes 38,890 OpRisk events. We apply PCA, reducing the data set to 50 columns, to avoid a subsequent excessive computational burden for UMAP. The percentage of explained variance by the first 50 principal components is 88.4%, confirming that the decrease in accuracy due to this dimensionality reduction can be considered negligible. The proportion of explained variance for each one of the first 50 principal components is represented by the scree plot in Figure 3.7, which confirms that most of the variance is explained by the first five principal components, meaning that, including other principal components over the first 50 ones would not bring any material benefit in terms of explained variance.

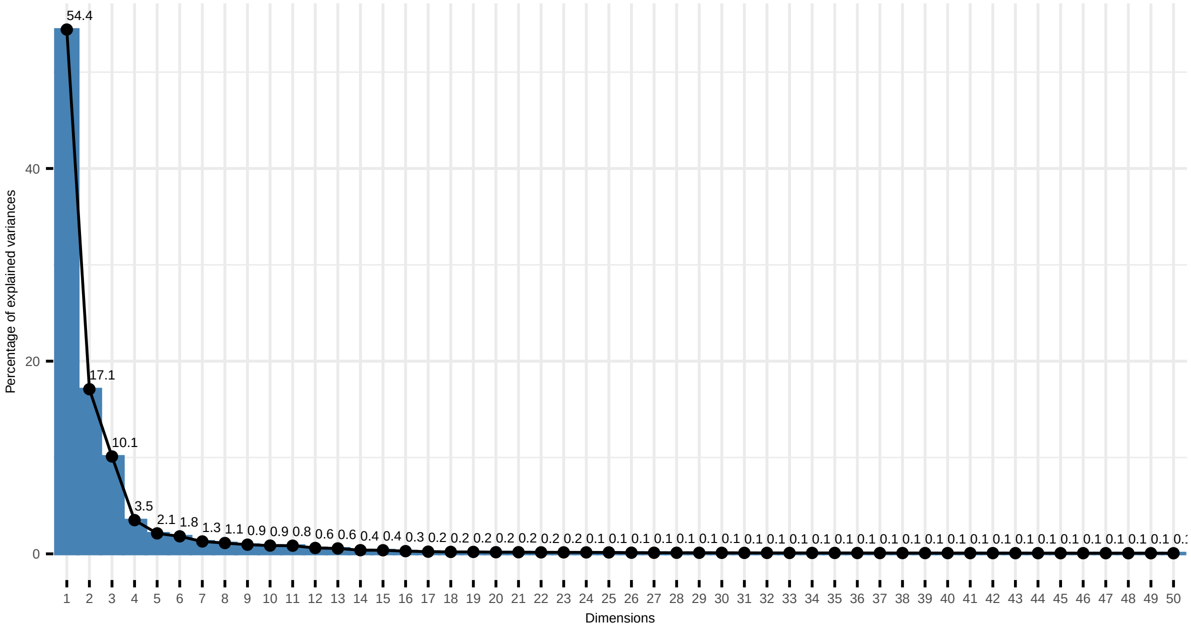


Figure 3.7: Scree plot related to the first 50 principal components for the event type “Clients, Products & Business Practices”.

Starting from the data set of the first 50 principal components, we apply UMAP to obtain a 2D representation, where the axes $V1$ and $V2$ represent the first two dimensions of UMAP. We report the UMAP 2D representation of the event type "Clients, Products & Business Practices" in Figure 3.8.

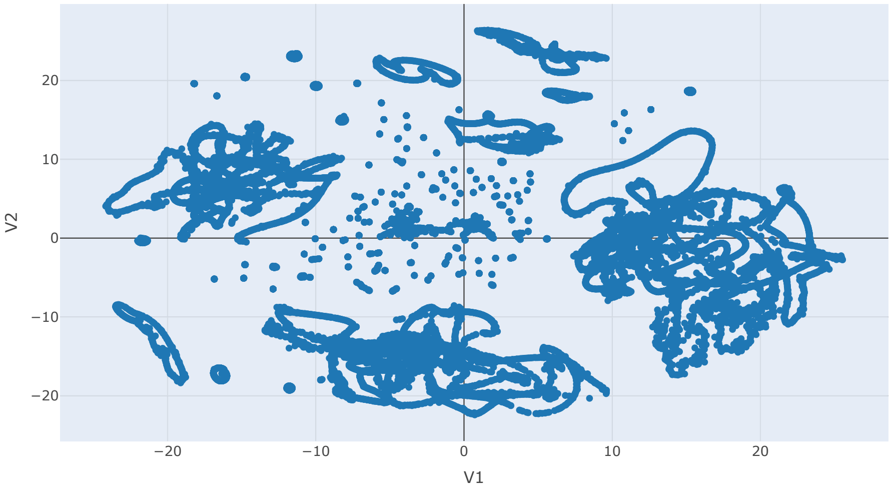


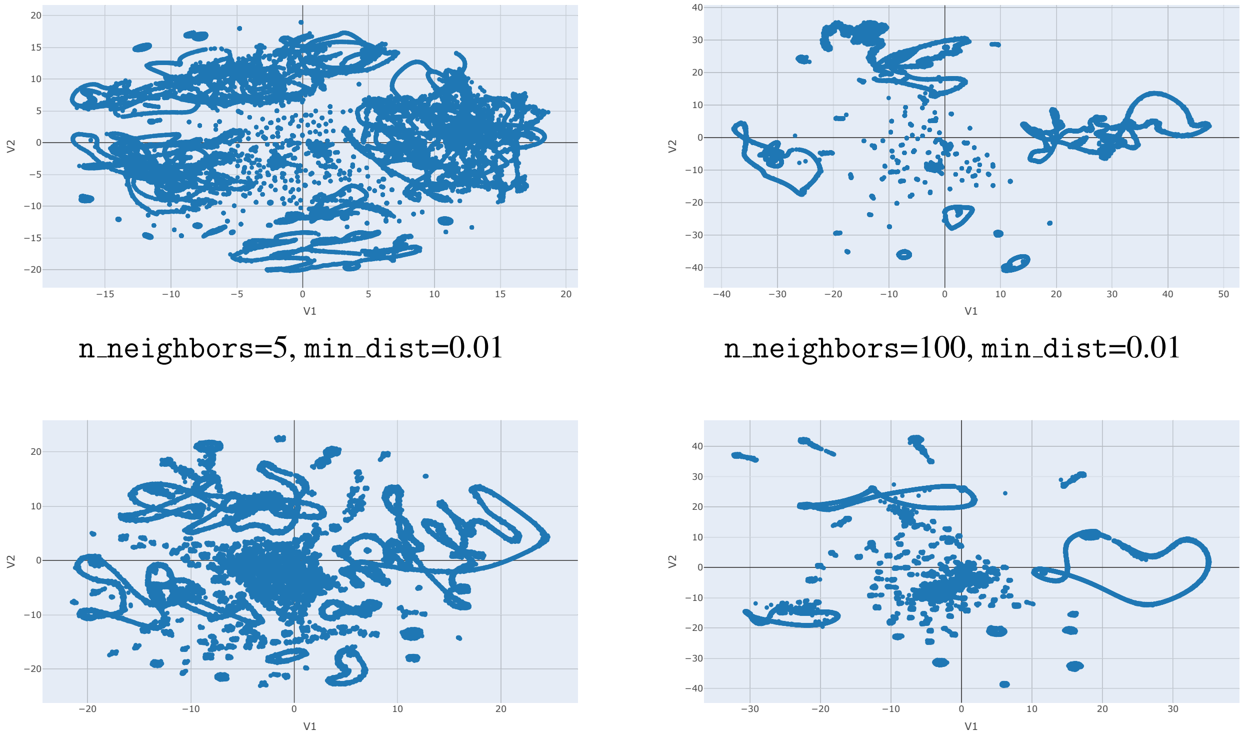
Figure 3.8: UMAP bidimensional representation of the semantic-aware document-by-term matrix for the event type "Clients, Products & Business Practices".

The default hyperparameters of the function `umap` in the package `uwot` have been used, *i.e.*, Euclidean metric, `n_neighbors`=15 and `min_dist`=0.01. Varying the hyperparameters `n_neighbors` from 5 to 100, and `min_dist` from 0.01 to 1, as reported in Figure 3.9, does not provide significantly different evidence. We can observe that, decreasing `n_neighbors` to 5, preserves the local data structure, compacting the clusters among them, whereas increasing it to 100, resulting in more global views of the manifold, leads to more separated clusters. Regarding `min_dist`, we can observe that by increasing it to 1 we have more dispersal points within each cluster. However, no additional or different clusters are highlighted varying these hyperparameters.

The UMAP 2D representation supports the activity of the analysts, who can identify several clusters with the main related tokens. Based on the UMAP representation, it emerges that the following clusters can be identified:

1. CHF Loans Bulk: disputes related to mortgages in Swiss Franch (included in the bulk of the provisions), identified by the tokens "chf_loan", "legal_dispute_client", etc.

2. CHF Loans Other: disputes related to mortgages in Swiss Franch (not included in the bulk



n_neighbors=5, min_dist=0.01 n_neighbors=100, min_dist=0.01

n_neighbors=5, min_dist=1 n_neighbors=100, min_dist=1

Figure 3.9: Sensitivity to hyperparameters of UMAP 2D representation.

of the provisions), identified by the tokens “client_benefit”, “chf_loan”, etc.

3. Anatocism: disputes related to irregularities in the interest rate calculations, identified by the token “anatocism”

4. Personal Loan Reimbursement: disputes related to personal loans, identified by the token “branch_reimbursement”

5. Derivatives Misselling: disputes related to contracts on derivatives, identified by the token “derivatives”

6. Client Account: disputes related to issues on current accounts, identified by the token “client_account”

Taking into account the knowledge of analysts, who identified six clusters with the main related tokens, we obtained the seeded topics and the seed tokens to be used for seeded LDA. We have run a seeded LDA with six seeded topics and one unseeded topic (to include the residual events).

In literature, there are attempts to automate the seed token selection, such as the one performed by Ferner *et al.* (2020). However, this approach would be applicable only when there is a single topic of interest (*e.g.*, the natural disasters in Ferner *et al.*, 2020), and not when there are multiple known and unknown topics to be analyzed as in our work.

We highlight that seeded LDA is directly run on the (rounded) semantic-aware document-by-term matrix, and not on a reduced dimensions data set. The UMAP 2D representation is used to support the activity of the analysts for identifying the clusters and the main related tokens to be used as seeds in the seeded LDA.

Seeded LDA, based on Gibbs sampling, has been run with 1000 iterations over a burn-in of 500. The convergence is confirmed by the trace plot of the log-likelihood function reported in Figure 3.10.

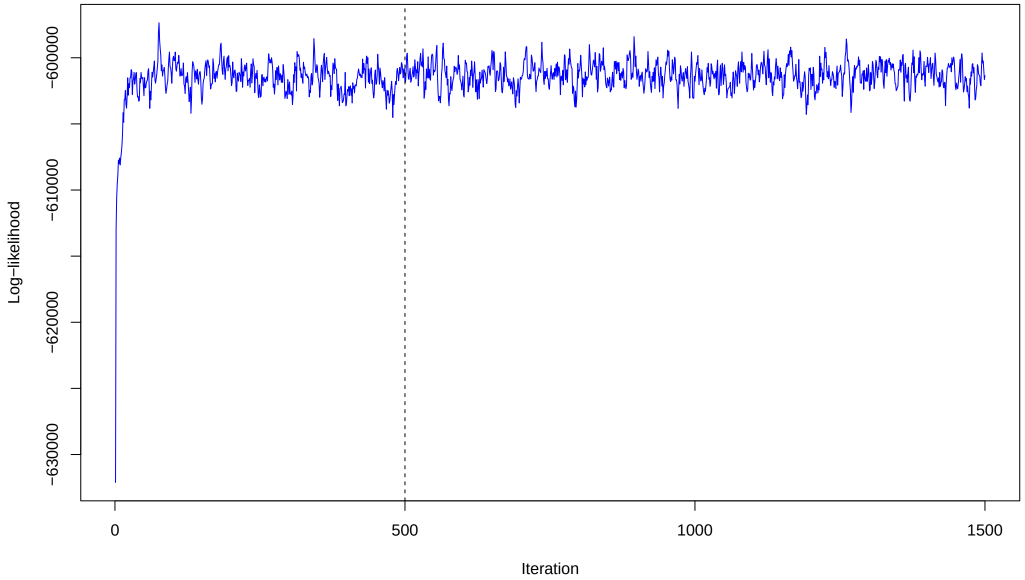


Figure 3.10: Trace plot of the log-likelihood function for the event type "Clients, Products & Business Practices".

The obtained perplexity is equal to 124.3 and has been calculated considering the 90% of the sample as the training set and the remaining 10% as the test set. In general, a lower perplexity score indicates better generalization performance, and we can observe that the obtained value is lower than the best values reported by Blei *et al.* (2003) in their simulation exercises.

The UMAP representation with the clusters identified by the seeded LDA is reported in Figure

3.11, where the group “EL0400 - CP&BP - Other” represents the residual cluster related to the unseeded topic.

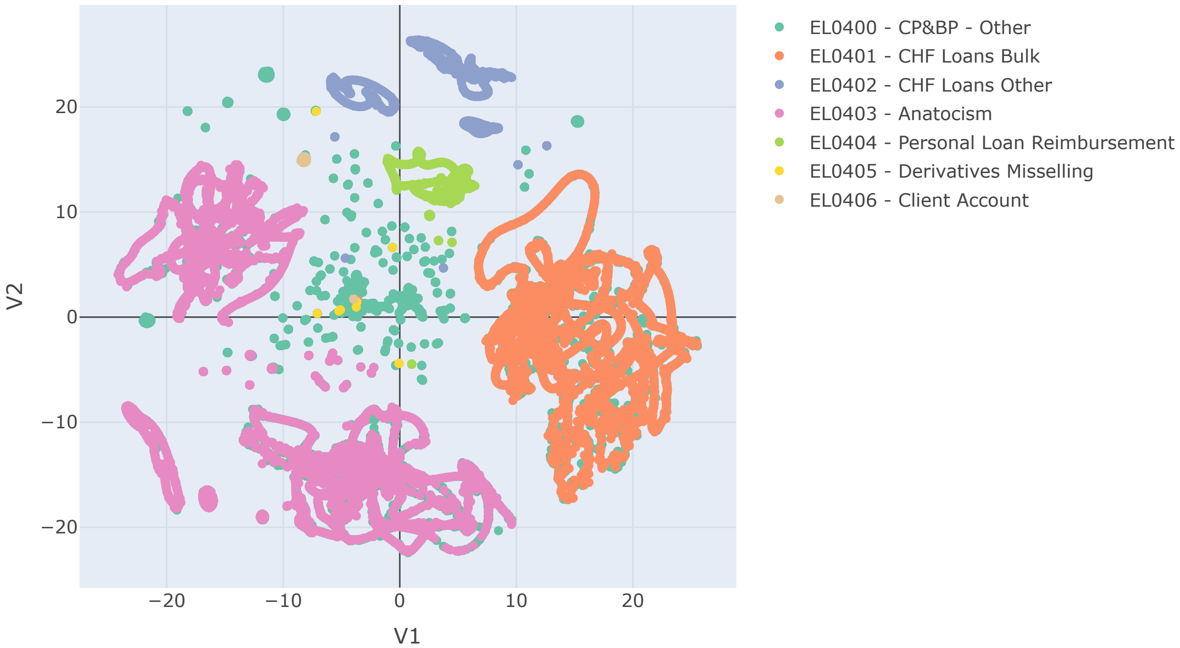


Figure 3.11: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the “basic” seeded LDA for the event type “Clients, Products & Business Practices”.

We can observe in Figure 3.11 that the clusters related to the topics “Anatocism” (*i.e.*, the pink points) and “CHF Loans Other” (*i.e.*, the orange points) are mixed with several data related to the residual cluster (*i.e.*, the green points). For instance, the cluster of pink points in the upper-left part of the UMAP 2D representation is composed of OpRisk events having (after cleaning) descriptions “anatocism”. Mixed with this cluster, there are several OpRisk events assigned to the residual cluster but having (after cleaning) the same description “anatocism”. Moreover, the small cluster of green points, placed at the coordinates $(\text{-}21.5, 0)$ have (after cleaning) descriptions such as “anatocism legal”. From a judgmental point of view, the aforementioned cases shall all be included in the cluster “Anatocism”. To obtain this meaningful result, we consider the averaging of topic probability distributions and the constraint on seed tokens described in Section 3.5.5. We obtain the results reported in Figure 3.12, where the aforementioned cases are no longer present. We can observe in Figure 3.12 that the clusters related to the topics “Anatocism” (*i.e.*, the pink points) and “CHF Loans Other” (*i.e.*, the orange points) are homogeneous and no longer mixed to the green points related to the residual cluster.

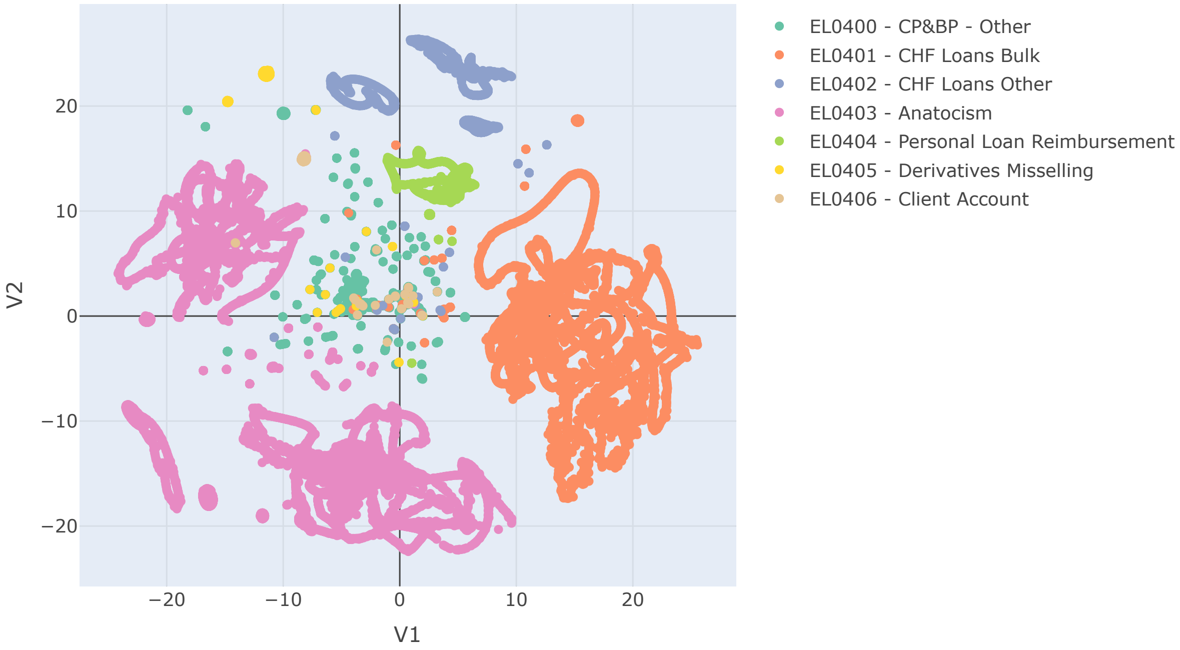


Figure 3.12: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA considering the averaging of topic probability distributions and the constraint on seed tokens for the event type "Clients, Products & Business Practices".

Therefore, by applying seeded LDA with the two brand-new features, we obtain a strong agreement of clusters obtained with this methodology compared to the UMAP 2D representation of the clusters. Other event type results are reported in Appendix A.

3.8 Application to tweet data

This application analyses the tweet data that have been extracted as explained in Section 3.4 and have been processed as illustrated in Section 3.6. The time series related to the number of daily tweets is reported in Figure 3.13. We noted that there are six days reporting a number of tweets much lower than the average. Regarding the first day (*i.e.*, May 5th), the lower number of tweets is because we started extracting them at around 9:00 PM, meaning that only around four hours of data were available. For the other five days (*i.e.*, May 21th and 27th, June 7th and 18th, and July 9th), the paucity of data was due to some sporadic downtime in the Twitter API service or the Virtual Machine used to extract the tweets. Note that these days are not considered for the calculation of

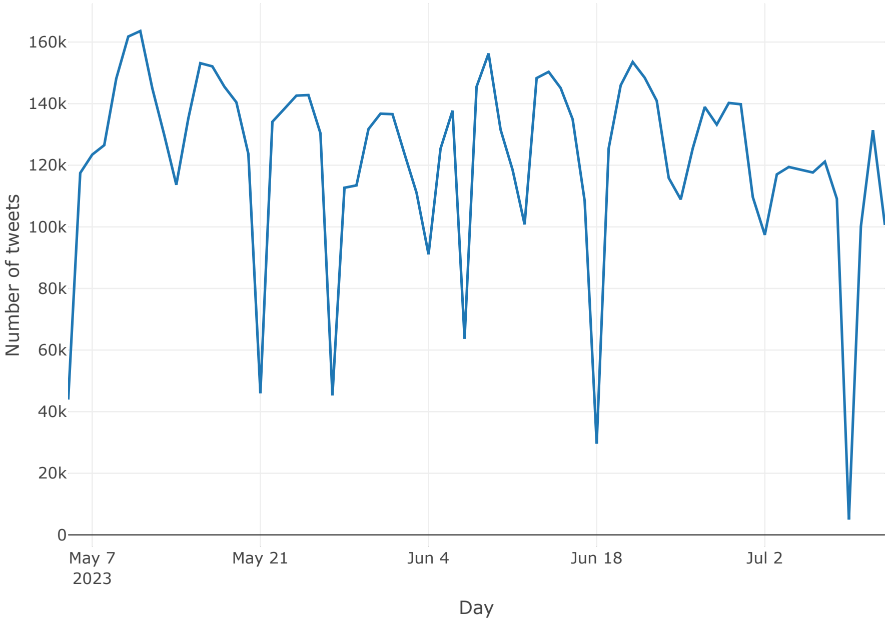


Figure 3.13: Number of daily tweets from May 5th to July 12th 2023.

the 95% quantile of the normal distribution to detect possible peaks in the time series.

We applied seeded LDA to each daily data set, considering the seeded topics and the related seed tokens reported in Figure 3.8. We decided to include five unseeded topics to make room for the identification of unforeseen and possibly relevant OpRisk topics. For each day, we clustered each tweet based on the topic with the highest probability (considering the constraint on seed tokens described in Section 3.5.5) and obtained, for each topic, the time series of the number of daily tweets reported in Figure 3.14 (averaging the topic probability distributions, among tweets having the same tokens, was not strictly necessary because duplicated tweets were already deleted during the cleaning step). Observing the trend of the tweets by topics, it emerges that there are peaks for "06_Financial_Crime" for days May 14th, and June 9th and 14th, as it is even more evident in Figure 3.15, which reports only this topic. The most significant peak is related to the June 9th, which reports around 10,000-15,000 more tweets than on other days. Therefore, we can analyze the UMAP 2D representation including the tweets of that day, reported in Figure 3.16 (calculated starting from the data set of the first 50 principal components). Deep diving into the interactive version of Figure 3.16, we realized that several tweets in the clusters related to "06_Financial_Crime" (*i.e.* the pink ones) are referred to as an alleged bribe accepted by Joe Biden (the current US President) from a Ukrainian energy company (Burisma) between 2015 and 2016, when he was Vice President (refer,

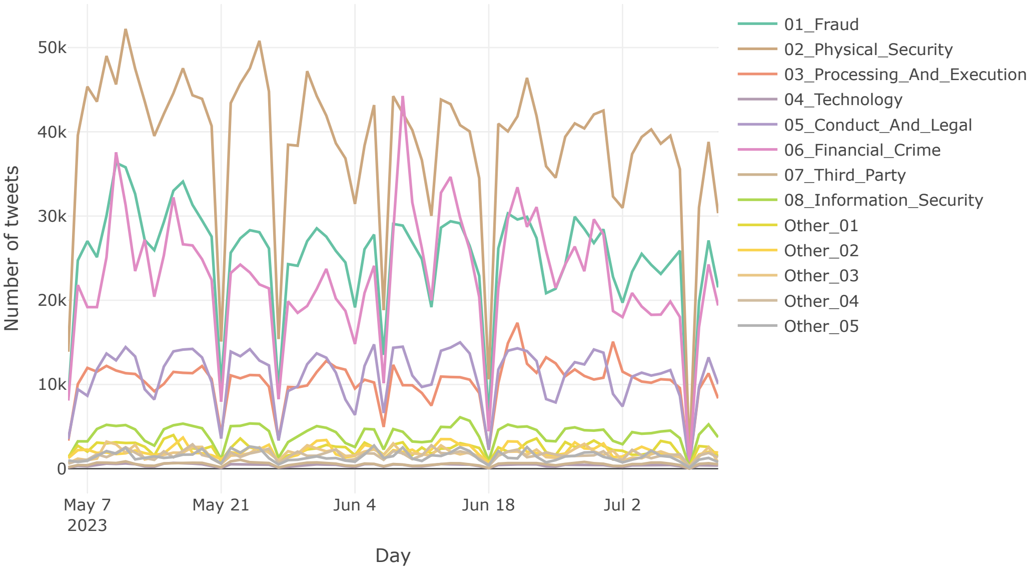


Figure 3.14: Number of daily tweets by cluster from May 5th to July 12th 2023.

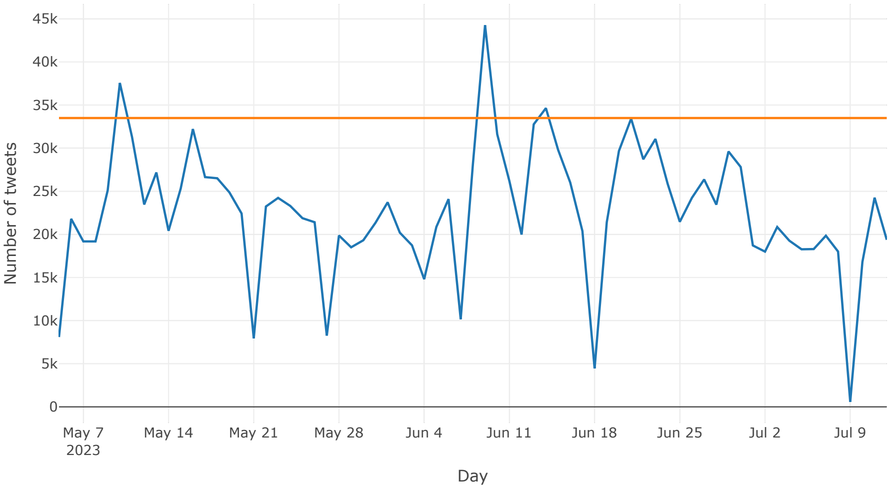


Figure 3.15: Number of daily tweets for 06_Financial_Crime from May 5th to July 12th 2023, where the orange line represents the 95% quantile of the normal distribution estimated on the time series.

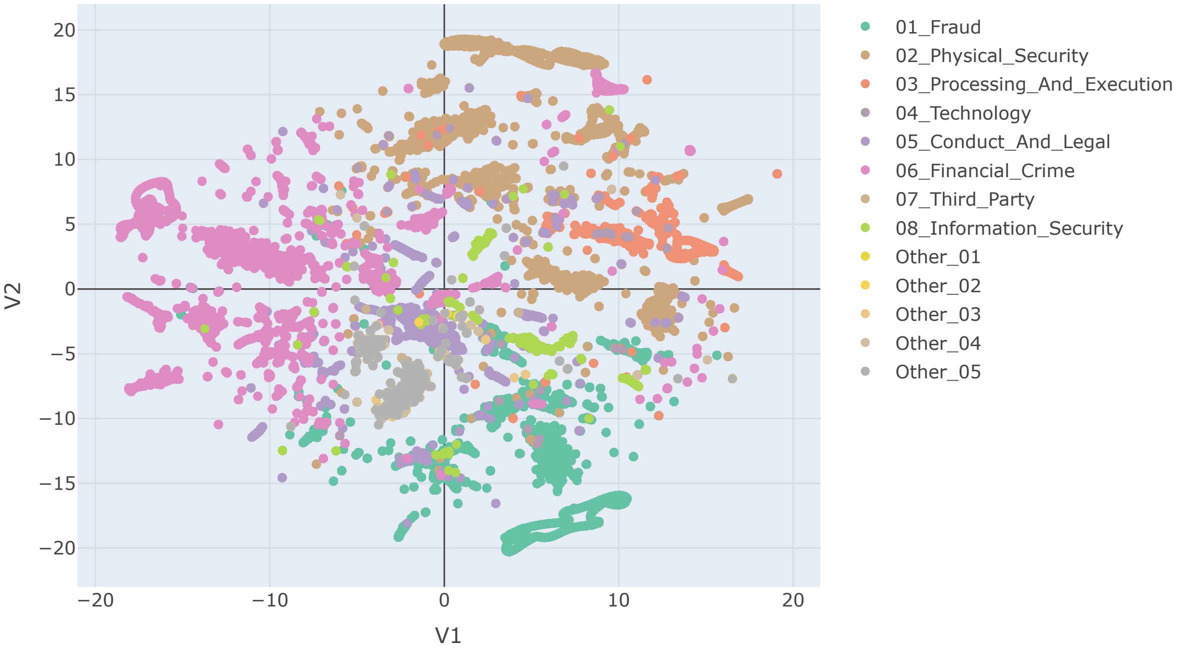


Figure 3.16: UMAP 2D representation of tweets related to June 9th 2023.

for example, to Livemint, 2023).

It is also useful to check if the significant loadings in the PCA 2D representation (*i.e.*, the loadings that are not very close to the origin) are composed of meaningful terms. We observe that all the significant loadings in Figure 3.17 are represented by meaningful terms for OpRisk purposes. Had this not been the case, one should have removed such terms from the analysis as stop-words. In particular, we can observe, among the significant loadings, the terms "bribe" and "corrupt", which are consistent with the mentioned alleged Biden bribery case.

We performed a check on June 9th tweets to verify that the peak is referred to this news. We selected the tweets that contained {"bribe" OR "corrupt"} AND {"biden" OR "burisma"}, and we obtained 12,690 data, confirming our hypothesis. However, this information is not relevant as a possible early warning for a financial institution since it has mainly political implications and refers to an event that occurred 7-8 years ago. Deep diving other peaks on days May 14th and June 14th, related to "06_Financial_Crime", did not highlight any other particular OpRisk event, meaning that these peaks were related to a multitude of several aspects and not due to a few main OpRisk topics.

As a further analysis, we observe that the topic "08_Information_Security" also presents a peak,

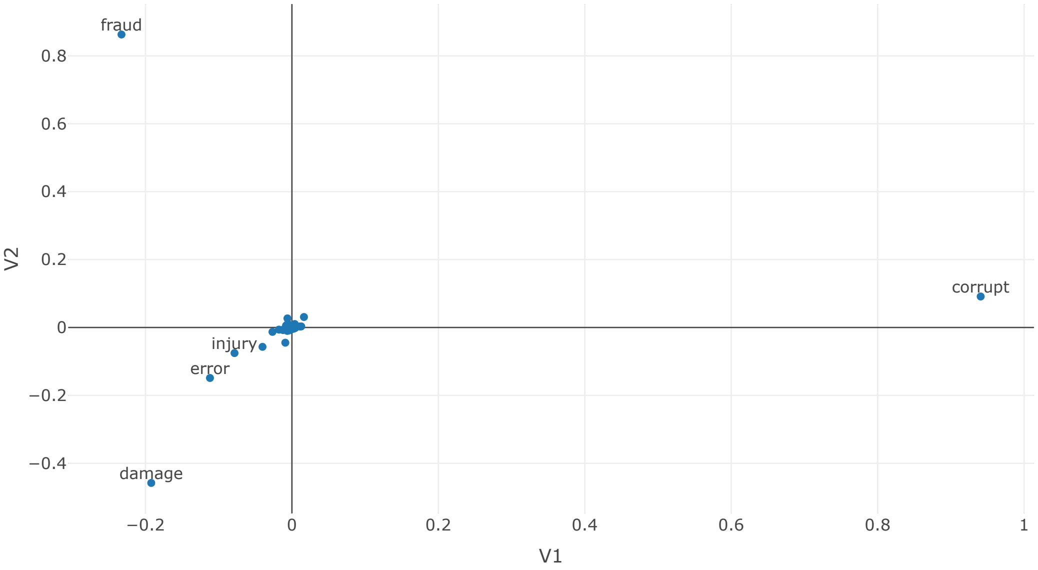


Figure 3.17: Loadings in PCA 2D representation of tweets related to June 9th 2023.

as highlighted in Figure 3.18. The observed peak is related to June 15th and June 16th, which reports around 1000 more tweets than on other days. Therefore, we can analyze the UMAP 2D representation including the tweets of those days, reported in Figures 3.19 and 3.20. Deeply analyzing the interactive version of Figures 3.19 and 3.20, it can be observed that several tweets in the clusters related to "08_Information_Security" (*i.e.* the light green ones) are related to a cyberattack on US government agencies (refer, for example, to CNN, 2023). It is also useful to check if the significant loadings in the PCA 2D representation are composed of meaningful terms. We observe that all the significant loadings in Figures 3.21 and 3.22 are represented by meaningful terms for OpRisk purposes. In particular, among the significant loadings, we can observe the term "scam", which could in some cases be consistent with cyber risk events.

We checked the June 15th-16th tweets to verify that the peak is referred to this news. We selected the tweets that contained {"cyber" OR "attack" OR "hack"} AND {"govern" OR "agencies" OR "america" OR "us" OR "u.s."}, and we obtained 1961 data, confirming our conjecture. This information is much more relevant than the previous one as a possible early warning for a financial institution, since it is related to a global cyberattack perpetrated by Russian cybercriminals occurring during those days. It potentially could have evolved into a cyber pandemic crisis affecting the financial system. Therefore, it would have been very useful for financial institutions to have been informed as soon as possible, to set up the proper prevention initiatives.

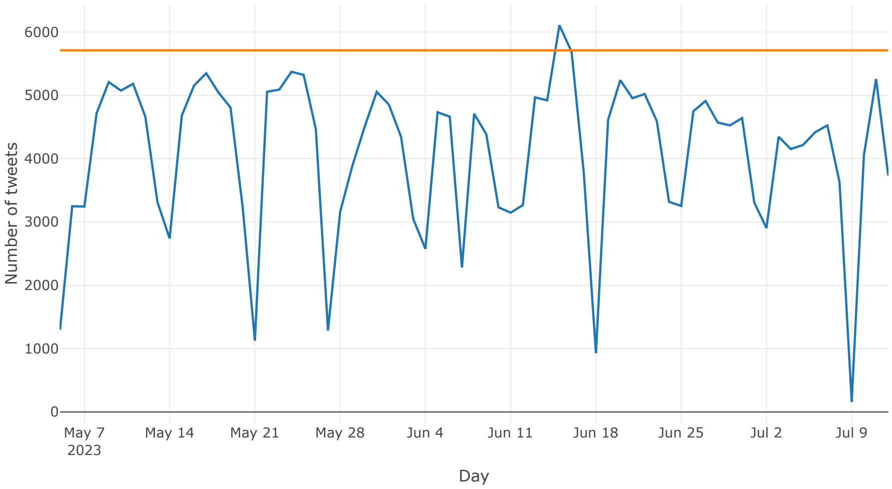


Figure 3.18: Number of daily tweets for 08_Information_Security from May 5th to July 12th 2023, where the orange line represents the 95% quantile of the normal distribution estimated on the time series.

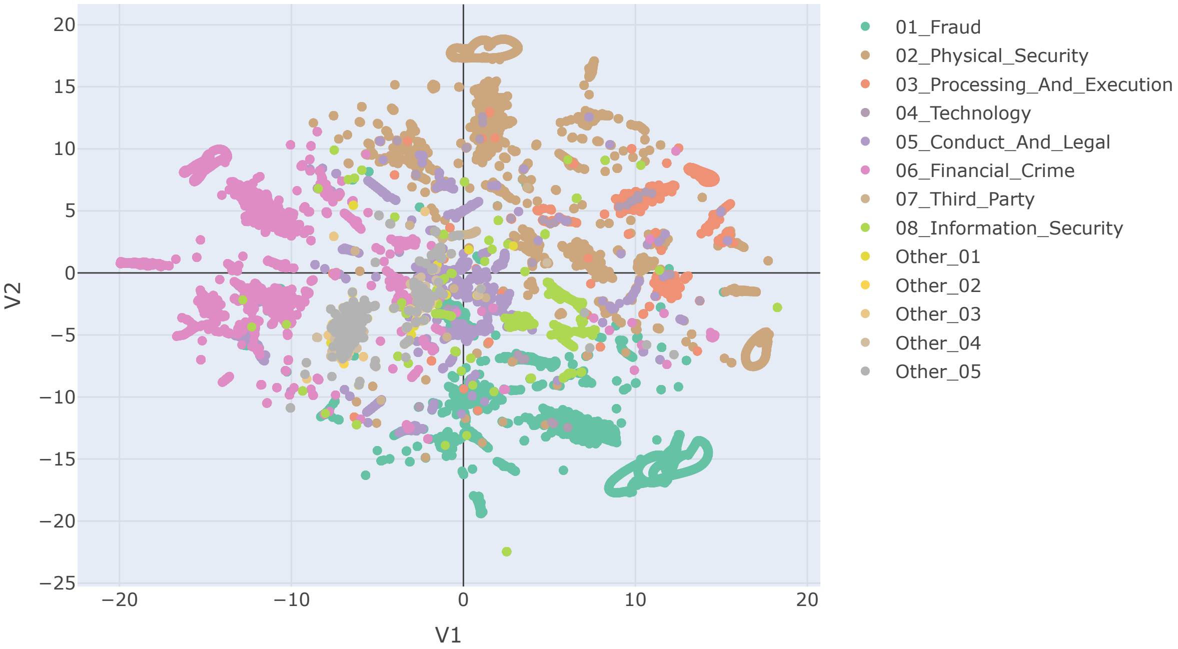


Figure 3.19: UMAP 2D representation of tweets related to June 15th 2023.

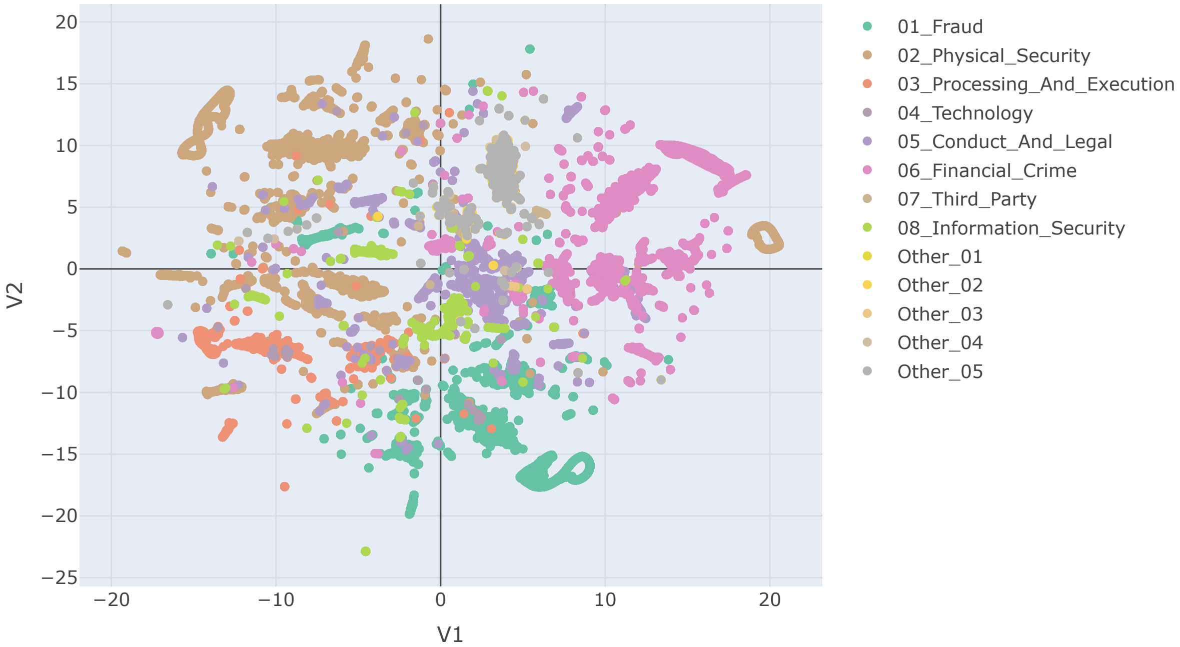


Figure 3.20: UMAP 2D representation of tweets related to June 16th 2023.

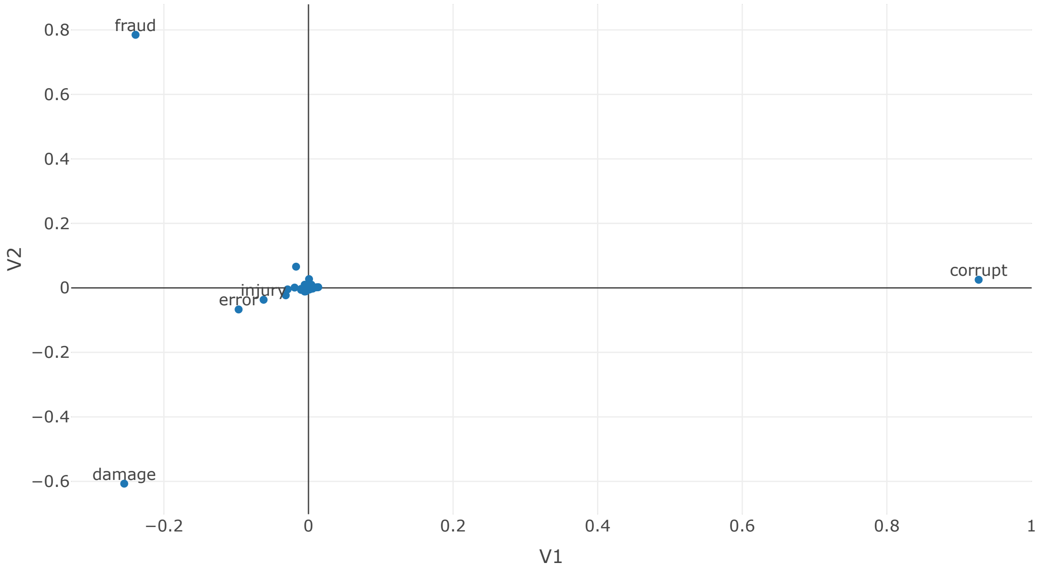


Figure 3.21: Loadings in PCA 2D representation of tweets related to June 15th 2023.

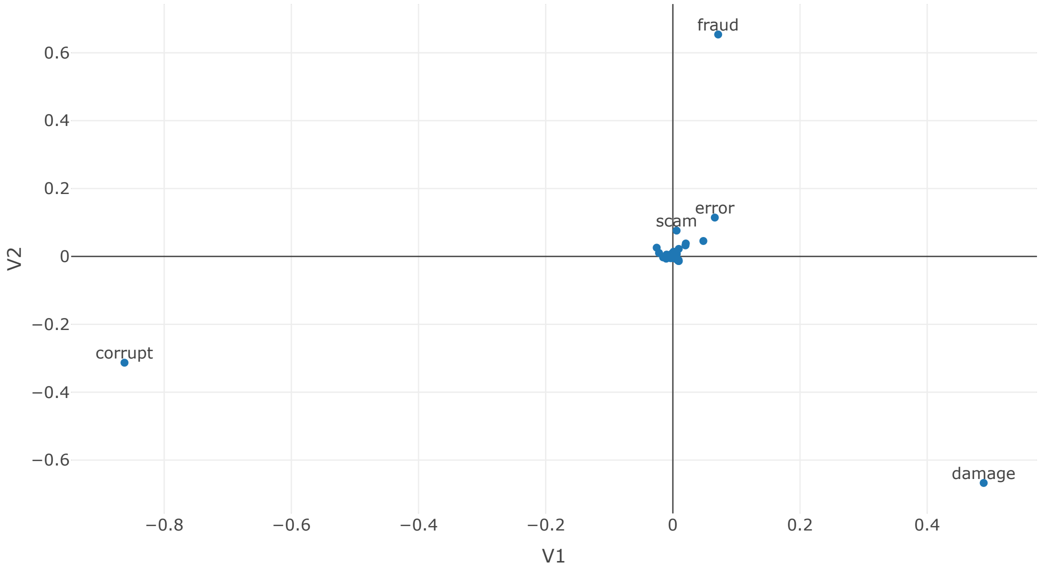


Figure 3.22: Loadings in PCA 2D representation of tweets related to June 16th 2023.

To detect emerging new OpRisk topics, it is useful to analyze the word clouds related to the five unseeded topics for each daily result. As an example, inspecting the word cloud of the 4th unseeded topic related to June 15th, it appears (in the lower part) the token "severe_thunderstorm_warm", that can be seen in Figure 3.23. These warnings were related to problematic weather conditions in the United States, in particular in the Southeast regions (refer, for example, to National Weather Service, 2023, Wikipedia, 2023, and Youtube, 2023). This early warning could have been relevant for financial institutions having branches in the affected regions, allowing them to set up initiatives to prevent or at least mitigate the damages to employees and buildings.

3.9 Concluding remarks

An OpRisk management framework is an approach to mitigating the risks associated with organizational operations. It involves identifying, assessing, monitoring, and controlling risks that could result in adverse outcomes that affect organization's ability to meet its goals and objectives. OpRisk encompasses a wide range of potential threats, including natural disasters, human mistakes, inadequate procedures or technologies, cyberattacks, financial losses due to fraud or theft, and sanctions imposed for regulatory violations. To successfully manage these risks, organizations must have a comprehensive approach that incorporates all aspects of business operations.

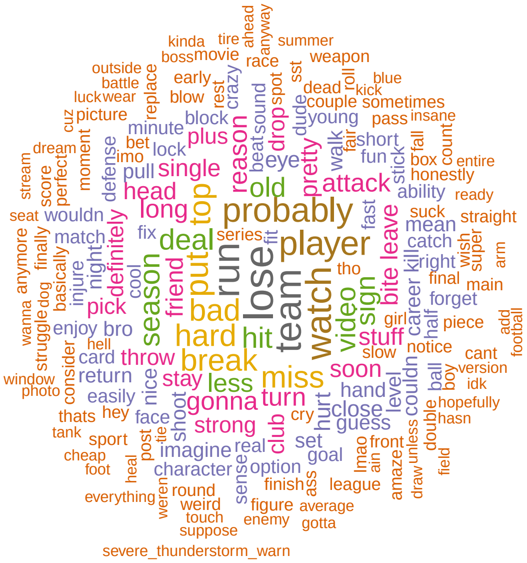


Figure 3.23: Word cloud for the 4th unseeded topic of tweets related to June 15th 2023.

This study represents a significant advancement in the application of text analysis techniques to OpRisk event descriptions. Notably, it pioneers the development of a comprehensive workflow that seamlessly integrates such analysis with data from diverse non-OpRisk-specific sources, particularly web data. The overarching objective is to establish an analytical and measurement framework able to assimilate OpRisk information from various origins, classifying it into topics based on the ORX taxonomy, and detecting emerging topics for early warning. By identifying surging issues, the approach helps to timely inform decision-makers and allows emerging problems to be addressed before they bring about large-scale adverse impacts. Employing recent statistical approaches and models, we utilized UMAP for data representation in a reduced space and seeded LDA for clustering in the analysis of OpRisk event descriptions. Our refinement of standard text analysis techniques for OpRisk descriptions involved the incorporation of n-grams based on the contemporary ORX taxonomy, as well as relevant bigrams and trigrams determined by their frequency of occurrence. The focus of our investigation centered on the extended UniCredit CoRep data set, where the application of the described text analysis methods and clustering techniques proved insightful. A notable illustration is the identification of six root causes within the event type ”Clients, Products & Business Practices,” demonstrating high agreement between clusters defined by seeded LDA and the UMAP representation.

Extending our analysis to approximately two months of daily tweets, we uncovered instances causing peaks in OpRisk related topics. Remarkably, a peak in tweets related to cyberattacks emerged as a potential early warning for financial institutions. Additionally, the detection of an emerging OpRisk topic concerning severe thunderstorms in Southeast U.S. regions suggests preemptive actions for potential damages.

While our proposed framework lays a robust foundation for OpRisk event analysis and the incorporation of web data, further enhancements and extensions are conceivable. Future research directions include:

- Training other word embedding techniques, such as `GloVe`, `BERT`, or `fastText` on large OpRisk data sets.

- Identifying relevant n-grams, with $n > 3$, considering explicitly the multiplicity of the applied statistical tests.

- Incorporating web data sources, other than tweets, on a wider time window.
- Considering more advanced techniques to detect significant peaks and trends in the daily number of tweets for each OpRisk related topic.

In essence, the successful application of our methodology underscores its potential for transforming how financial institutions approach OpRisk management, offering a comprehensive and adaptive tool for anticipating and mitigating potential risks.

Chapter 4

Conclusion and future work

In recent years, financial institutions have increasingly embraced advanced OpRisk analytics, surpassing regulatory mandates to bolster managerial decision-making, as elucidated in Chapter 1. The latest strides in artificial intelligence, spanning natural language programming and machine learning, have facilitated the integration of text analysis into OpRisk textual data. To be prepared for Basel II requirements, major financial institutions commenced the meticulous collection and storage of OpRisk loss event data during the first years of this century. Presently, these institutions possess OpRisk data sets spanning 15-20 years, encompassing not only attributes essential for regulatory quantitative analysis (such as loss amount, date, event type, and business line) but also rich free-text data, including detailed OpRisk event descriptions. Consequently, a natural progression has been the application of text analysis techniques to delve deeper into these event descriptions. However, our study revealed a notable absence in the existing literature regarding a well-structured workflow for conducting such analyses, establishing a definitive best practice amid the myriad options available, and outlining the key steps that warrant consideration. Addressing this gap, our primary objective in this manuscript has been to define a robust workflow, elucidate essential steps, and offer best practices for conducting text analysis on OpRisk event descriptions. Pursuing these goals, we have not only refined prevailing statistical methods for quantitative data but also made substantive contributions to shaping a holistic OpRisk management framework.

In Chapter 2, we defined a first kernel for the workflow, applying various statistical models to analyze and cluster OpRisk event descriptions, using text analysis techniques to identify their main root causes. Moreover, we have enriched the standard text analysis by including a semantic adjustment to deal with different words expressing similar concepts. We have considered several clustering and topic modeling techniques and evaluated their accuracy to the clustering performed by the analysts. We found that, when applied to the UniCredit CoRep dataset with accounting dates from 2018 to 2021 and a minimum loss threshold of €100,000 for the event type "Clients, Products & Business Practices", *k*-means and LDA emerged as the most effective clustering and topic modeling techniques. This application successfully revealed two homogeneous clusters of events.

In Chapter 3, we have extended and improved the workflow leading it to an advanced level of maturity. The text analysis of OpRisk event descriptions has been enriched with several crucial features, such as language detection (to select only one language within multi-language descriptions), relevant *n*-grams recognition (based on ORX taxonomy, and the bigrams and trigrams frequency of occurrence), semantic adjustment based on a word embedding trained on OpRisk data (to include specific subject terms, not usually present in pre-trained word embeddings), UMAP, *i.e.*, one of the most advanced techniques for dimensionality reduction (as a fundamental tool to support data exploration by OpRisk analysts to identify main root causes), and seeded LDA (to automatically cluster OpRisk descriptions among the root causes identified by analysts, and detect other OpRisk topics). Moreover, we improved the accuracy of seeded LDA, applied to OpRisk descriptions, including two additional brand-new features, *i.e.*, averaging the topic probability distributions of identical vectorized descriptions, and constraining the clusters' assignment of a description to one the seeded topics, in case the related seed tokens were present in it.

It is worth mentioning that the improved workflow, compared to the one described in Chapter 2, can be applied to much more challenging and larger data sets. In particular, it was applied to the UniCredit CoRep data set with accounting dates 2005-2022 and a minimum loss threshold of € 1000 for all event types. For the event type "Clients, Products & Business Practices", the improved workflow led to the identification of six relevant root causes with a very high level of agreement between the 2D UMAP representation and the seeded LDA results.

Furthermore, the workflow was extended even beyond the initial goals, integrating the data from the social media X (formerly known as Twitter) within a harmonized framework. The final goal has not been limited to providing financial institutions with standardized tools and best practices to deep-dive the main root cause within their data which has already occurred (*i.e.*, referred to backward-looking view). It was also (and especially) to detect increasing and emerging OpRisk topics (*i.e.*, referred to as forward-looking view), in order to provide OpRisk early warnings. Once the financial institutions are alerted with an early warning, they can set up proper actions to prevent or mitigate such OpRisk issues. Analyzing approximately two months of daily tweets, we detected instances causing peaks in OpRisk-related topics. As a significant example, a peak in tweets related to cyberattacks was identified as a potential early warning for financial institutions. Additionally, the detection of an emerging OpRisk topic concerning severe thunderstorms in Southeast U.S. regions could lead financial institutions to perform actions to prevent or strongly reduce potential damages.

Despite that the proposed workflow can be considered a strong advancement in the analysis of OpRisk event descriptions and related web data, we recognize there is still room for improvements and extensions regarding several aspects. Future developments can include:

- Training other word embedding techniques, such as `GloVe` (Pennington *et al.*, 2014), or `fastText` (Bojanowski *et al.*,2017), or fine-tuning a pre-trained word embedding, such as `BERT` (Devlin *et al.*, 2019), on OpRisk data sets, using several methods to assess their adequacy and compare them. This would represent an extension of the currently proposed method, where CBOW and Skip-gram were trained and compared using mainly qualitative drivers.

- Identifying relevant n-grams, with $n > 3$, considering explicitly the multiplicity of the applied statistical tests, instead of applying it as a unique binomial test with a lower significance level, using, *e.g.*, Bonferroni correction or the more accurate Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995). This would allow this methodology to start from a more standard significance level (*e.g.*, 5%), adjusting the obtained p-values for the number of performed statistical tests.

- Incorporating other relevant web data sources for a wider time window to more accurately detect emerging OpRisk topics. Data could be extracted from other social media, such as Threads (which is attempting to become a major rival of X (Milmo, 2023)), and news data providers, such as Bloomberg, Reuters, and Talkwalker.

- Considering more advanced techniques to detect significant peaks and trends in the daily number of tweets for each OpRisk-related topic. This element becomes more and more important once several web sources are analyzed for longer periods. For instance, methods to detect change points can be used (Chen and Gupta, 2012), which are implemented in the R package `changepoint` (Killick and Eckley, 2014; Killick *et al.*, 2022).

- Applying extensions of LDA (once adapted to include term seeds) able to model the topics dependence (while LDA assumes topics to be independent), such as Correlated Topic Model (CTM, Blei and Lafferty, 2007) and Structural Topic Model (STM, Roberts *et al.*, 2013). The latter allows also for the inclusion of topical prevalence covariates (metadata that explain topical prevalence) and topical content covariates (variables that explain topical content).

- Extending the analysis to Reputational Risk measurement, since severe OpRisk events (especially, internal frauds) can have a reputational impact, impacting the stock price of the financial institution (Perry and de Fontnouvelle, 2005). The number of tweets for OpRisk topics, referring to the financial institution, can be used as additional covariates in the Reputational Risk measurement.

As a concluding consideration, the escalating complexity and digital transformation within the financial landscape, coupled with the exponential expansion of available information, underscore the pivotal importance of cultivating a profound understanding of OpRisks and the swift detection of signals indicating their potential emergence. This imperative not only serves as a present-day cornerstone for success in financial institutions but also emphasizes the ongoing commitment to fostering a dynamic risk management framework. In light of these considerations, it becomes paramount for financial institutions to continually invest in research, technological innovation, and adaptive strategies, ensuring they not only navigate the current complexities but also remain agile and resilient in the face of future uncertainties in the OpRisk landscape.

Appendix A

Other event type results on OpRisk data

This appendix complements the results reported in Chapter 3.7, which were limited to the event type "Clients, Products & Business Practices". Here we report the UMAP 2D representations for other event types with the clusters identified by the seeded LDA, considering the averaging of topic probability distributions and the constraint on seed tokens.

A.1 Internal fraud

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. Client Account: internal fraud on client accounts, identified by the tokens "client_account", "account_payment", etc.
2. Unfaithfulness: cases of employees' unfaithfulness, identified by the tokens "unfaithfulness", "unfaithful_employee", etc.
3. ATM Fraud: internal fraud on ATM devices, identified by the tokens "atm", "cash_box", etc.

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.1.

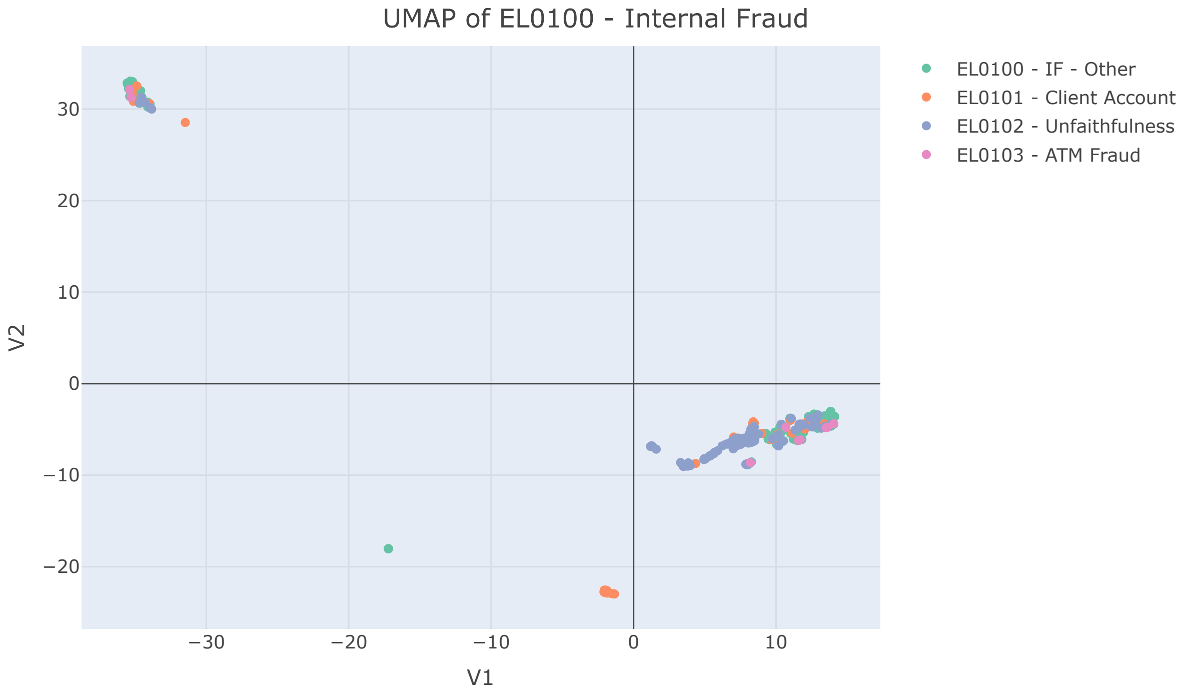


Figure A.1: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type "Internal Fraud".

The cluster "client Account" is well identified in the lower part of the UMAP 2D representation, while other clusters appear to be a bit mixed.

A.2 External fraud

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. Internet Card Fraud: fraud on credit cards related to internet payments, identified by the tokens "card_transaction_steal", "steal_credit_card", etc.

2. Unauthorized Card Transaction EEA: unauthorized card transactions in European Economic Area (EEA), identified by the tokens "unauthorized_card_transaction", "execute_eea_investigation", etc.

3. Card Cloning: cloning of credit cards, identified by the token "card_frad_cloning".

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.2.

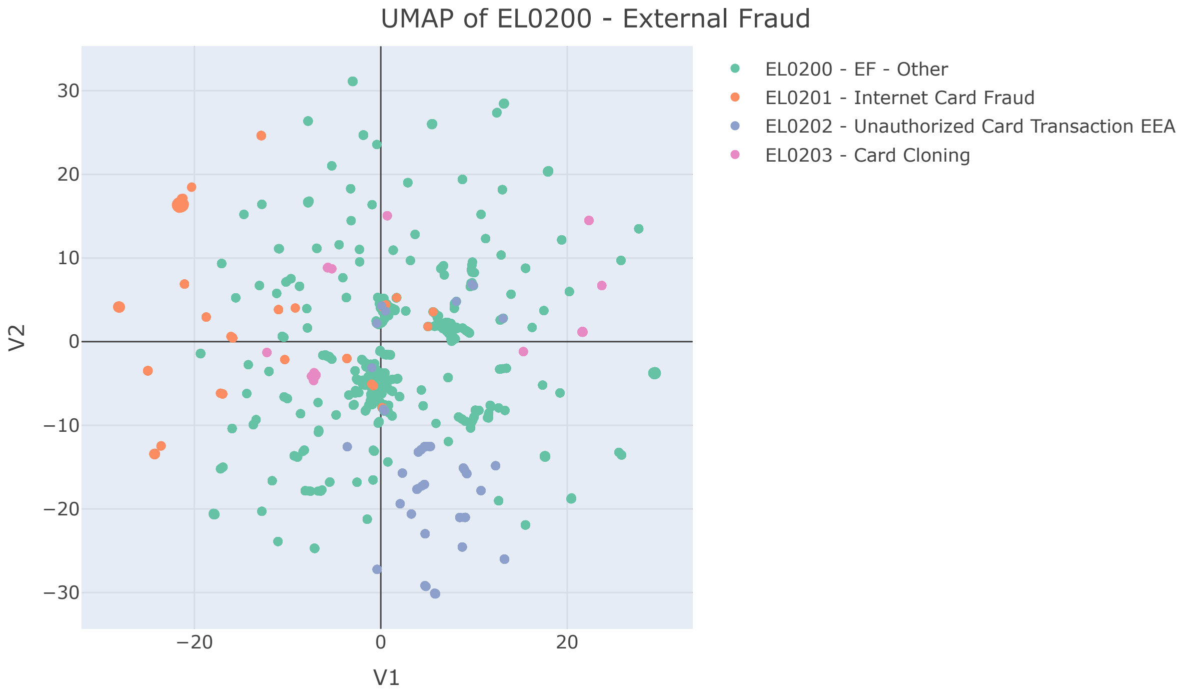


Figure A.2: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type "External Fraud".

The clusters "Internet Card Fraud", "Unauthorized Card Transaction EEA", and "Card Cloning" are well identified in the UMAP 2D representation, but they appear to be separated into several smaller clusters (even if they do not seem to be identified by significant distinctive aspects).

A.3 Employment practices and workplace safety

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. Work Injury: reimbursements to injured employees, identified by the tokens "work_injury" and "injury".

2. Former Employee Litigation: disputes with former employees, identified by the tokens "former_employee_litigation", "legal_procedure_non-competition", etc.

3. Demotion: disputes with employees for claimed demotion, identified by the tokens "demotion" and "compensatory".

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.3.

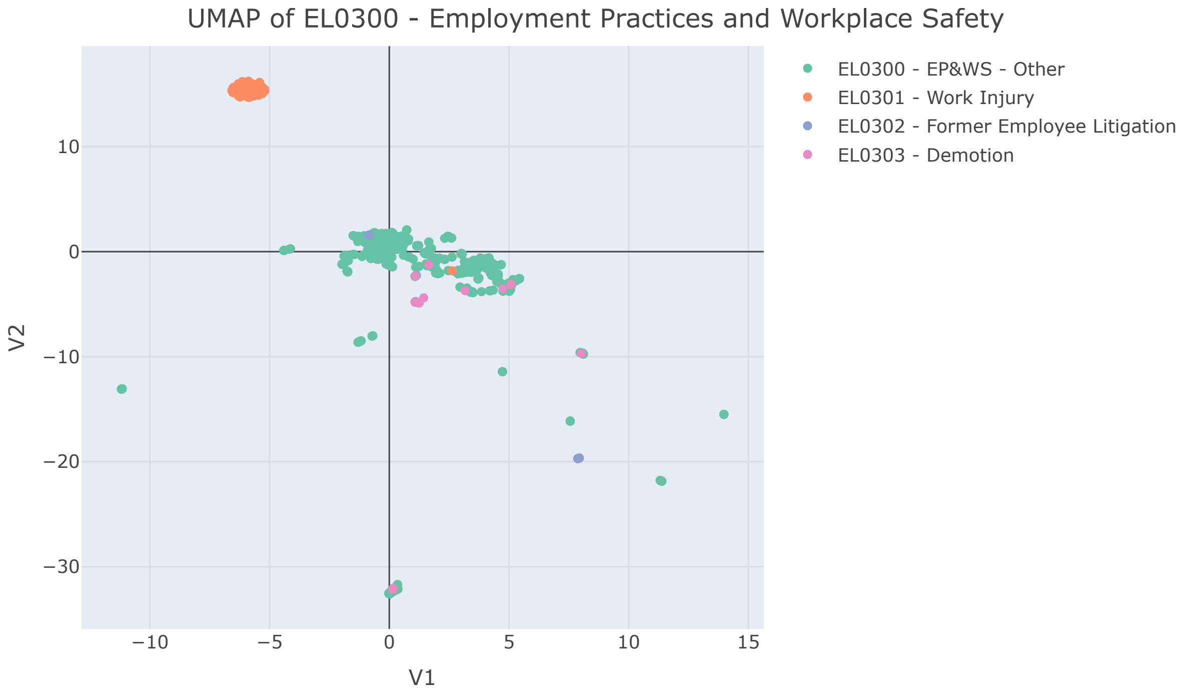


Figure A.3: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type “Employment Practices and Workplace Safety”.

The cluster “Work Injury” is well identified in the upper part of the UMAP 2D representation, while other clusters appear to be a bit mixed.

A.4 Damage to physical assets

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. ATM EE: damages to ATM devices in East Europe (EE), identified by the tokens “atm”, “vandalism_bgn_pay”, etc.

2. Car Damage: damages to company cars, identified by the tokens “car_damage”, “car_accident”, etc.

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.4.

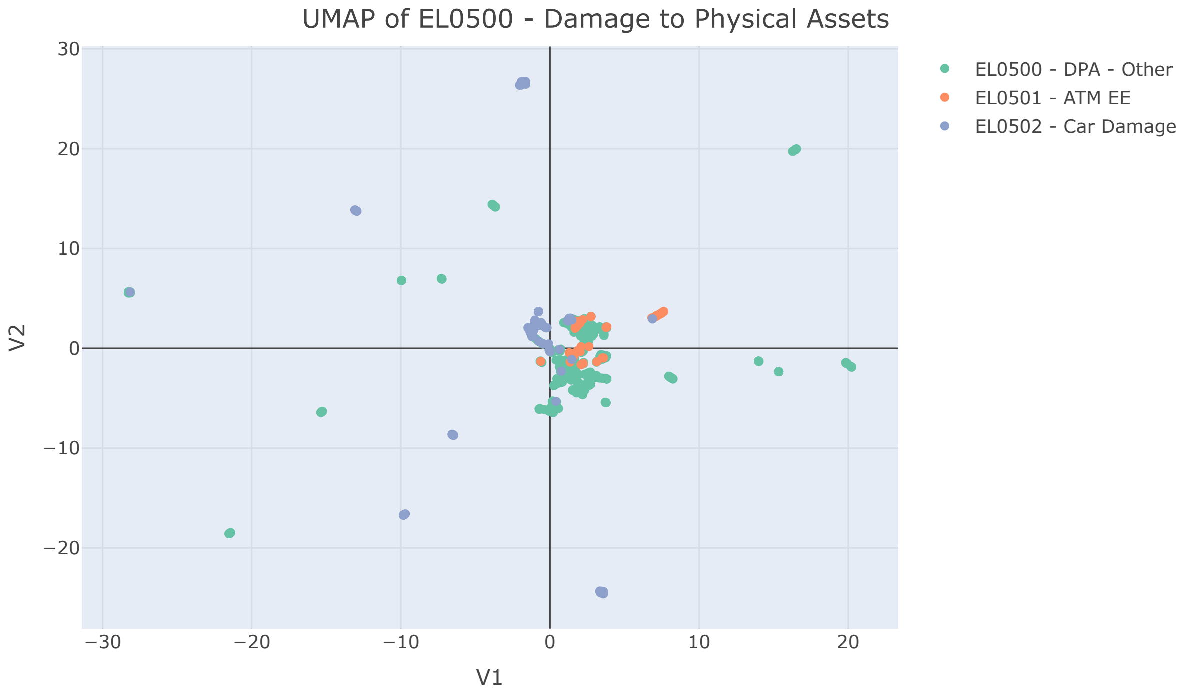


Figure A.4: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type "Damage to Physical Assets".

The cluster "Car Damage" is well identified in the UMAP 2D representation (even if separated into smaller clusters without significant distinctive aspects), while other clusters appear to be a bit mixed.

A.5 Disruption and system failures

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. Software Bugs: software bugs in the IT applications used by the company, identified by the tokens "software_bug", "bug", etc.
2. Loss IT Problem: losses caused by IT issues, identified by the token "loss_problem".
3. Digital Payment Processes: issues in the digital payments, identified by the tokens "card", "digital_payment_procedure", etc.

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.5.

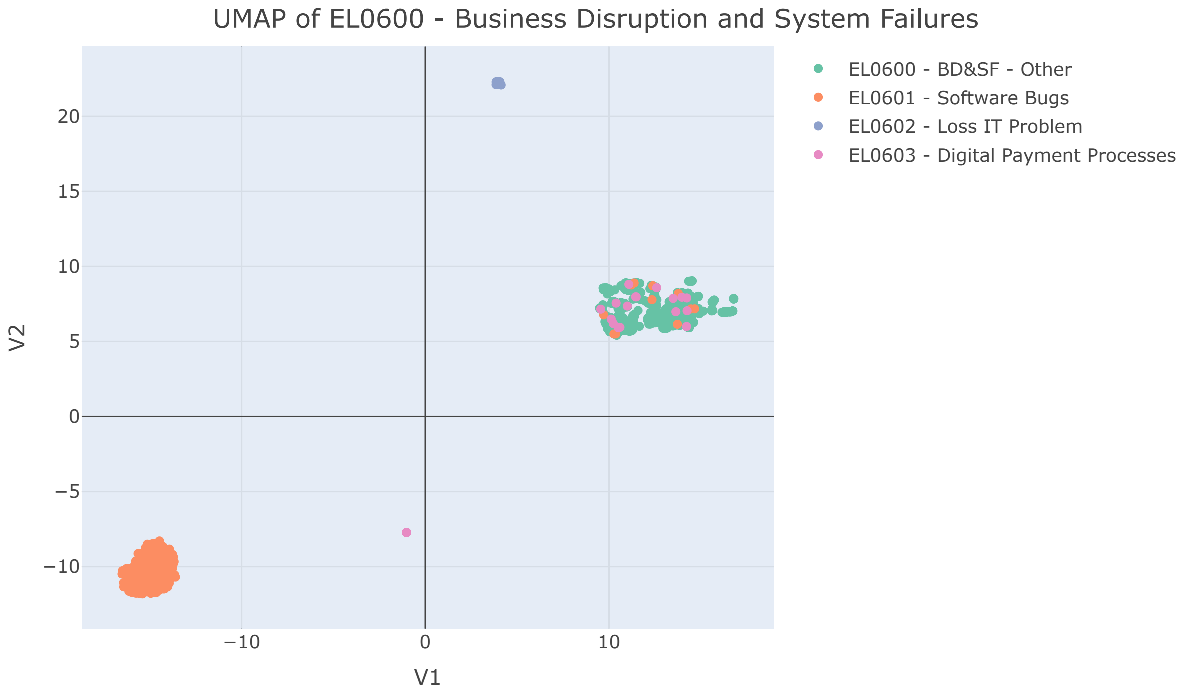


Figure A.5: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type "Disruption and System Failures".

The cluster "Loss IT Problem" is well identified in the upper part of the UMAP 2D representation. The cluster "Software Bugs" is well identified in the left-lower part of the UMAP 2D representation (apart from some sparse points mixed with the residual cluster on the right), while other clusters appear to be a bit mixed.

A.6 Execution, delivery and process management

Based on the UMAP representation, it emerges that the following clusters can be identified:

1. Inadequate Data: issues related to the inadequate recording of data, identified by the tokens "inadequate_data_concern", "return_inadequate_data", etc.
2. Cash Differences: issues related to cash differences detected by reconciliation, identified by the token "cash", "reconcile", etc.
3. Client Account: errors on client accounts, identified by the token "client_account".

4. Error False Notification: errors related to false notifications, identified by the token "error_false_notification".

The UMAP representation with the clusters identified by seeded LDA is reported in Figure A.6.

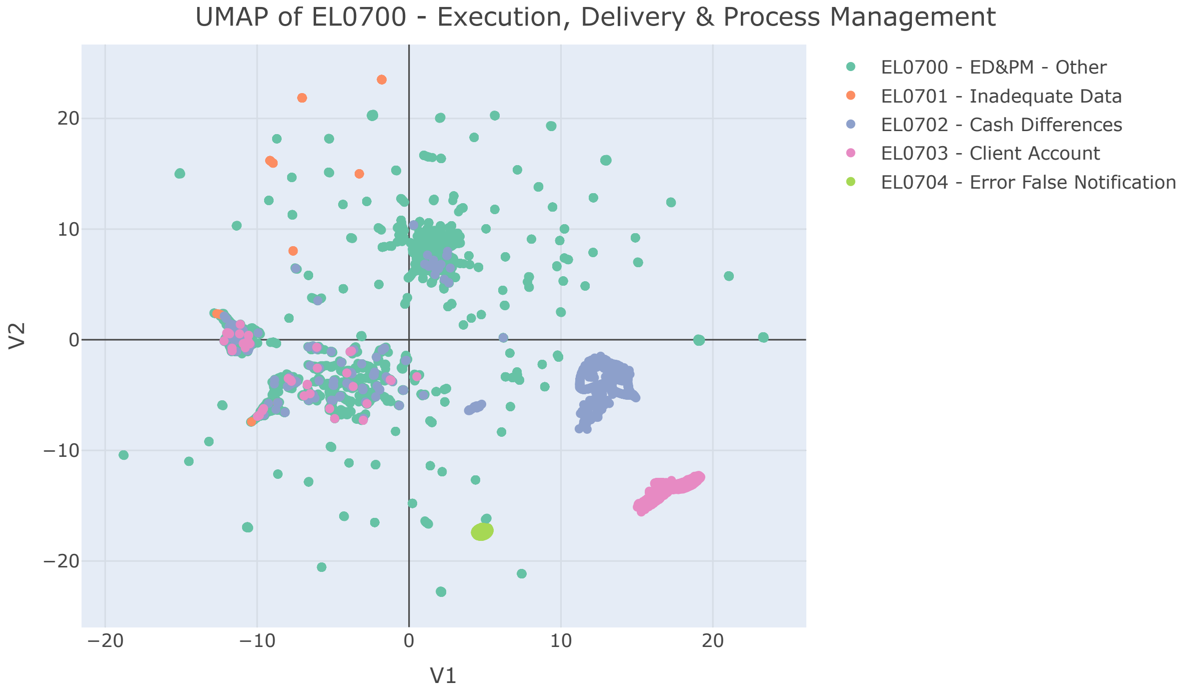


Figure A.6: UMAP 2D representation of the semantic-aware document-by-term matrix with the clusters identified by the seeded LDA for the event type "Execution, Delivery and Process Management".

The cluster "Error False Notification" is well identified in the lower part of the UMAP 2D representation. The clusters "Cash Differences" and "Client Account" are well identified in the right-lower part of the UMAP 2D representation (apart from some sparse points mixed with the residual cluster in the center). The cluster "Inadequate Data" is well identified in the left-upper UMAP 2D representation (even if separated into smaller clusters without significant distinctive aspects).

A.7 Perplexity for each event type

The perplexity for each event type, calculated as described in Section 3.7, is reported in Table A.1.

Table A.1: Perplexity for each event type.

Event type	Perplexity
Internal fraud	2539.7888
External fraud	1281.5586
Employment practices and workplace safety	2174.3971
Clients, Products & Business Practices	124.3482
Damage to physical assets	1069.5868
Disruption and system failures	1748.2363
Execution, delivery and process management	6096.6611

The values in Table A.1 confirm the evidence from UMAP 2D representations, where the event type ”Clients, Products & Business Practices” shows its main clusters to be very well separated in Figure 3.12 of Section 3.7.

Appendix B

Code details

This appendix describes the codes written to implement the analyses shown in Chapter 3 (which constitute a generalization of the analyses performed for Chapter 2). First of all, all the codes have been written using the R language, version 4.3.1, (R Core Team, 2023) and leveraging on several R packages available on CRAN (Several authors, 2024). The related files are publicly available on the author's GitHub page:

- https://github.com/FabioPiacenza/OpRiskTextAnalysis for R scripts.
- https://github.com/FabioPiacenza/TweetData for input data.

The analyses have been performed through the following R scripts:

- TextAnalysis2.R: this script contains the code to produce analyses and save results for the UniCredit OpRisk data set.
- ReportTextAnalysis2.R: it loads the results calculated and saved by the previous script, and produces all the reports and the charts.
- TwitterFromR_Scheduler.R: it imports tweets based on specified keywords using the package `rtweet` (Kearney, 2019). It was automatically scheduled to run hourly using the package `taskscheduleR` (Wijffels and Belmans, 2023).
- TwitterFromR_Scheduler2.R: it imports tweets based on specified accounts. It was automatically scheduled to run daily.

- TwitterAnalysis.R: it produces analyses and saves results for tweet data set on a certain time range (*e.g.*, a specific day).

- WrapperTwitterAnalysis.R: it recalls the previous script to perform the tweet data analysis for each day and saves the related results.

- ReportTwitterAnalysis.R: it loads the results calculated and saved by the previous script, and produces all the reports and the charts.

It is worth mentioning that the first two scripts cannot be re-run based on the files available in GitHub, considering that the UniCredit data set of OpRisk events cannot be shared for data sensitivity reasons. These codes are available just for a better understanding of the steps of the analysis. Eventually, they could be applied to another data set with a similar structure. Also the third and the fourth scripts cannot be re-run since X dismissed the used API on July 12th, 2023. However, the imported tweets for June 15th and 16th are available on the author's GitHub page. Based on tweet data, the codes in the last three scripts can be re-run to reproduce the results shown in Chapter 3 for tweets of June 15th and 16th, 2023. Other imported tweets can be provided upon request. The next Sections report more details on the used scripts.

B.1 TextAnalysis2.R

The script is structured as follows:

- Load needed packages.

- Import UniCredit OpRisk data set.

- Clean the descriptions (both chronological and CoRep ones), *e.g.*,

  - remove the spaces at the beginning and at the end of each description;
  - substitute a sequence of two spaces, or a sequence such as “ - ”, “. ”, “, ”, “; ”, with a line feed (basically, causing a carriage return), in order to separate sentences;

- Import the ORX taxonomy to use its 3$^{\text{rd}}$ level, together other special words (*e.g.*, “chf”, “anatocism”, “Covid”), for English language detection (*i.e.*, sentences including the strings related to 3$^{\text{rd}}$ level ORX taxonomy or special words, are forced to be assigned to English).
- Dectect the English sentences (inside both chronological and CoRep descriptions), using the function `detect_language` of the package `cld2` (Riesa and Giuliani, 2013; Ooms, 2022).
- Select the English sentences and discard all others (*i.e.*, select the English sentences within the descriptions, whereas descriptions without English sentences are discarded).
- Reduce the descriptions to lower case.
- Tokenize the descriptions and apply further cleaning (*e.g.*, remove punctuation, numbers, symbols, separators, and URLs).
- Perform lemmatization (*i.e.*, reduce each word to its lemma).
- Define a specific dictionary (*e.g.*, all the words starting with “anat” and “anatocism” are assigned to the word “anatocism”).
- Remove stop-words, *e.g.*, the ones related to the English language have been derived from meta::cpan (2021), together with other specific ones identified for UniCredit data.
- Select relevant bigrams and trigrams.
- Select relevant *n*-grams from 3$^{\text{rd}}$ level ORX taxonomy.
- Integrate previously selected relevant *n*-grams, together with other specific ones (*e.g.*, “chf_loan”).
- Build the document-by-term matrix, excluding all the terms appearing less than 5 times in the corpus.
- Train the word embeddings CBOW and Skip-gram on UniCredit chronological descriptions, ORX News digest texts, and scenario analysis storylines.
- Select the nearest 5 words, based on cosine similarity, for each relevant term (*e.g.*, “bank”, “client”) to qualitatively assess the previously trained word embeddings.

- Compute the word similarity matrix based on the selected word embedding (*i.e.*, CBOW).
- Perform the semantic adjustment of the document-by-term matrix (based on CoRep descriptions), using the word pairs with cosine similarity higher than the significant threshold (*i.e.*, 0.8).
- Calculate the PCA of the semantic-aware document-by-term matrix, selecting the first 50 principal components. Calculate their explained variance, and produce the related scree plot.
- Produce the plot of the first two principal components, and the plot of contribution of terms to the first two principal components. These plots are interactive (being produced with package `plot_ly`) and allow to visualize the description corresponding to each point.
- Calculate the LSA on the first 50 principal components.
- Produce the interactive plot of the first two LSA components, and the interactive plot of contribution of terms to the first two LSA components.
- Calculate the UMAP on the first 50 principal components. Produce the related interactive plot.
- The previous five steps, related to PCA, LSA, and UMAP, are performed also for each event type.
- For each event type, perform the sensitivity analysis of UMAP with respect to the hyperparameters `n_neighbors` and `min_dist`.
- For each event type, perform the seeded LDA based on the specified seeds (allowing one unseeded topic for residual descriptions), produce the related trace plots, and compute the related perplexity.
- For descriptions identified by identical rows in the semantic-aware document-by-term matrix, average the respective topic probabilities.
- Produce the word cloud for each topic.

- Assign each description to the cluster related to the topic showing the highest probability, constraining the assignment to the seeded topic related to the present seed words, if any.
- For each event type, produce the interactive plots for PCA, LSA and UMAP, representing the assignment to the different clusters with points of different colours. Save the results useful to reproduce the plots without re-running all the calculations.

B.2 ReportTextAnalysis2.R

The script is structured as follows:

- load needed packages.
- Specify paths for reading input files `path`, and for writing output files `pathFigures`.
- Specify the desired charts to be produced (*e.g.*, set `plotly_pdf=TRUE` to produce plotly charts in pdf).
- Specify if plot titles have to be included or not.
- Select time range of the input files to be considered. This time range is specified to select the results of `TextAnalysis2.R` among the ones previously produced.
- Import the UniCredit OpRisk data set.
- PCA of Group data without clusters.
- PCA for each event type without clusters.
- PCA term contributions of Group data.
- PCA term contributions for each event type.
- PCA for each event type with clusters.
- Explained variance of PCA for each event type.
- LSA for each event type without clusters.

- LSA term contributions for each event type.

- LSA for each event type with clusters.

- UMAP of Group data without clusters.

- UMAP for each event type without clusters.

- UMAP sensitivity to hyperparameters for each event type without clusters.

- UMAP for each event type with clusters.

- Word clouds for each event type by topic.

- Trace plot for each event type.

B.3 TwitterFromR_Scheduler.R

The script is structured as follows:

- Load needed packages.

- Specify API key (formerly provided by Twitter).

- Create token for API connection.

- Define the string with keywords.

- Define the file name, based on the date time, to save the imported tweets.

- Load the files of tweet IDs already imported.

- Extract the tweets using the function `search_tweets`.

- Save the imported tweets into the previously defined file.

B.4 TwitterFromR_Scheduler2.R

The script is structured as follows:

- Load needed packages.
- Specify API key (formerly provided by Twitter).
- Create token for API connection.
- Define the accounts related to tweets to be imported.
- Define the file name, based on the date time, to save the imported tweets.
- Load the files of tweet IDs already imported.
- Extract the tweets using the function `get_timeline`.
- Save the imported tweets into the previously defined file.

B.5 TwitterAnalysis.R

The script is structured as follows:

- Load needed packages.
- Select paths for input data.
- specify the time range of tweets to be selected.
- Specify tweets to be selected, *i.e.*, tweets related to keywords, accounts, or both.
- Select relevant input files and import the contained tweets.
- If time range includes more than one day, produce the plot related to number of daily tweets.
- Clean the tweet texts (*e.g.*, convert to basic ASCII to avoid strange characters, convert everything to lower case, remove user names, links, tabs, punctuation, and duplicated tweets).

- Tokenize the tweets with further cleaning (*e.g.*, remove numbers, and perform lemmatization).

- Import dictionaries already created for UniCredit OpRisk data set, including, *e.g.*, bigrams, trigrams, and *n*-grams related to ORX taxonomy.

- Import the stop-words defined for UniCredit data, together with other ones identified on tweets (based on the analysis of significant PCA term contributions).

- Import proper names from the package `genderdata` to be excluded from tweets.

- Remove previously mentioned stop-words, proper names, and stop-words related to the English language (meta::cpan, 2021).

- Define relevant bigrams and trigrams for tweets.

- Integrate *n*-grams from ORX taxonomy, bigrams and trigrams from UniCredit data, and bigrams and trigrams identified for tweets.

- Define the seed words for seeded LDA.

- Define dictionary to merge the words containing seed into the related seeds (*e.g.*, "cyberattack" into "cyber").

- Create the document-by-term matrix, excluding terms which appears less than 5 times in the corpus.

- Apply the semantic adjustment to the document-by-term matrix.

- Calculate the PCA of the semantic-aware document-by-term matrix, selecting the first 50 principal components. Calculate their explained variance, and produce the related scree plot.

- Produce the plot of the first two principal components, and the plot of contribution of terms to the first two principal components. These plots are interactive (being produced with package `plot_ly`) and allow to visualize the tweet text corresponding to each point.

- Calculate the LSA on the first 50 principal components.
- Produce the interactive plot of the first two LSA components, and the interactive plot of contribution of terms to the first two LSA components.
- Calculate the UMAP on the first 50 principal components. Produce the related interactive plot.
- Perform the seeded LDA based on the specified seeds (allowing five unseeded topic to discover emerging OpRisk related topics), and produce the related trace plots.
- Produce the word cloud for each topic.
- Assign each tweet to the cluster related to the topic showing the highest probability, constraining the assignment to the seeded topic related to the present seed words, if any.
- Produce the interactive plots for PCA, LSA and UMAP, representing the assignment to the different clusters with points of different colours. Save the results useful to reproduce the plots without re-running all the calculations.

B.6 WrapperTwitterAnalysis.R

The script is structured as follows:

- Specify the path where the script `TwitterAnalysis.R` is saved.
- Specify the path to save output results.
- Specify the time range for the tweets to be analyzed (*e.g.*, from May 5$^{\text{th}}$ to July 12$^{\text{th}}$, 2023).
- Run the loop executing the tweet data analysis (*i.e.*, the script `TwitterAnalysis.R`) for each day within the previously specified time range.

B.7 ReportTwitterAnalysis.R

The script is structured as follows:

- Load needed packages.

- Specify paths for reading input files `pathTwitter`, and for writing output files `pathFigures`.

- Specify the desired charts to be produced (*e.g.*, set `plotly_pdf=TRUE` to produce plotly charts in pdf).

- Specify if plot titles have to be included or not.

- Select time range of the input files to be considered. This time range is specified to select the results of `WrapperTwitterAnalysis.R` among the ones previously produced.

- Select the time range of the reference dates (*e.g.*, from May 5th to July 12th, 2023).

- Select relevant input files and import the contained tweets.

- If time range includes more than one day, produce the plot related to number of daily tweets.

- If time range includes more than one day, produce the plot related to number of daily tweets for each cluster.

- For each topic, calculate the 95% quantile of the estimated normal distribution to identify peaks that, potentially, could be related to significant OpRisks. The six days reporting a number of tweets much lower than the average are excluded from the normal distribution estimation.

- Daily PCA of selected tweets without clusters.

- Daily PCA term contributions of selected tweets without clusters.

- Daily PCA of selected tweets with clusters.

- Daily LSA of selected tweets without clusters.

- Daily LSA term contributions of selected tweets without clusters.

- Daily LSA of selected tweets with clusters.

- Daily UMAP of selected tweets without clusters.

- Daily UMAP of selected tweets with clusters.

- Daily word clouds by topic.

Appendix C

Computational aspects

All the calculations, performed in the thesis, were run on a Virtul Machine with processor AMD EPYC 7H12 64-Core Processor 2.60 GHz (4 processors) with installed RAM 24.0 GB, accessed through “VMWare Horizon Client”. It is worth mentioning that the most critical part, related to calculations, was the RAM memory consumption. Several parts of the calculation code have been optimized to avoid exceeding the available RAM. The next sections report the computational time of the R scripts described in Appendix B. We do not report computational data on `TwitterFromR_Scheduler.R` and `TwitterFromR_Scheduler2.R` since they cannot be run anymore, due to the dismissal of the used API. Computation times have been calculated using the R package `tictoc`.

C.1 TextAnalysis2.R

Details on computational time for the main steps of the script `TextAnalysis2.R` are reported in Tables C.1, C.2, and C.3.

Table C.1: Computation time for `TextAnalysis2.R` in seconds (part 1).

Step	Time (s)
Read data	10.8
Extract English descriptions	1306.62
Tokenizing text, and removing stop-words	15.87
Detect and integrate relevant n-grams	30.89
Creation of the document-by-term matrix	0.55
Training of word embedding	290.29
Perform semantic adjustment	3051.7
Include cleaned descriptions (based on tokenization) and select non-zero rows	4.42
PCA of Group	215.41
LSA of Group	6.69
UMAP of Group	210.97
PCA of EL0100 - Internal Fraud	159.44
LSA of EL0100 - Internal Fraud	0.02
UMAP of EL0100 - Internal Fraud	4.09
PCA of EL0200 - External Fraud	160.62
LSA of EL0200 - External Fraud	0.06
UMAP of EL0200 - External Fraud	50.17
PCA of EL0300 - Employment Practices and Workplace Safety	158.51
LSA of EL0300 - Employment Practices and Workplace Safety	0.01
UMAP of EL0300 - Employment Practices and Workplace Safety	6.69
PCA of EL0400 - Clients, Products & Business Practices	180.62
LSA of EL0400 - Clients, Products & Business Practices	0.25
UMAP of EL0400 - Clients, Products & Business Practices	144.75
PCA of EL0500 - Damage to Physical Assets	166.25
LSA of EL0500 - Damage to Physical Assets	0.03
UMAP of EL0500 - Damage to Physical Assets	7.04
PCA of EL0600 - Business Disruption and System Failures	166.04
LSA of EL0600 - Business Disruption and System Failures	0.03
UMAP of EL0600 - Business Disruption and System Failures	11.2
PCA of EL0700 - Execution, Delivery & Process Management	170.24
LSA of EL0700 - Execution, Delivery & Process Management	0.08
UMAP of EL0700 - Execution, Delivery & Process Management	26.64
Save data to reproduce PCA, LSA and UMAP by Event Type	45.27

Table C.2: Computation time for `TextAnalysis2.R` in seconds (part 2).

Step	Time (s)
Select data and seeds of EL0100 - Internal Fraud	0.08
Estimate seeded LDA of EL0100 - Internal Fraud	7.15
Perplexity of EL0100 - Internal Fraud	1.6
Topics probability of EL0100 - Internal Fraud	0.03
Averaging topic probabilities of EL0100 - Internal Fraud	46.65
Wordcloud of EL0100 - Internal Fraud	12.03
Assign clusters to descriptions of EL0100 - Internal Fraud	0.12
Select data and seeds of EL0200 - External Fraud	0.07
Estimate seeded LDA of EL0200 - External Fraud	10.87
Perplexity of EL0200 - External Fraud	4.98
Topics probability of EL0200 - External Fraud	0.25
Averaging topic probabilities of EL0200 - External Fraud	883.92
Wordcloud of EL0200 - External Fraud	10.03
Assign clusters to descriptions of EL0200 - External Fraud	7
Select data and seeds of EL0300 - Employment Practices and Workplace	0.06
Estimate seeded LDA of EL0300 - Employment Practices and Workplace	7
Perplexity of EL0300 - Employment Practices and Workplace	1.7
Topics probability of EL0300 - Employment Practices and Workplace	0.05
Averaging topic probabilities of EL0300 - Employment Practices and Workplace	94.5
Wordcloud of EL0300 - Employment Practices and Workplace	9.95
Assign clusters to descriptions of EL0300 - Employment Practices and Workplace	0.17
Select data and seeds of EL0400 - Clients, Products & Business Practices	0.11
Estimate seeded LDA of EL0400 - Clients, Products & Business Practices	40.79
Perplexity of EL0400 - Clients, Products & Business Practices	28.39
Topics probability of EL0400 - Clients, Products & Business Practices	1.64
Averaging topic probabilities of EL0400 - Clients, Products & Business Practices	6018.39
Wordcloud of EL0400 - Clients, Products & Business Practices	17.5
Assign clusters to descriptions of EL0400 - Clients, Products & Business Practices	37.99
Select data and seeds of EL0500 - Damage to Physical Assets	0.06
Estimate seeded LDA of EL0500 - Damage to Physical Assets	5.19
Perplexity of EL0500 - Damage to Physical Assets	1.64
Topics probability of EL0500 - Damage to Physical Assets	0.03
Averaging topic probabilities of EL0500 - Damage to Physical Assetsd	97.47
Wordcloud of EL0500 - Damage to Physical Assets	6.14
Assign clusters to descriptions of EL0500 - Damage to Physical Assets	0.18

Table C.3: Computation time for `TextAnalysis2.R` in seconds (part 3).

Step	Time (s)
Select data and seeds of EL0600 - Business Disruption and System Failures	0.06
Estimate seeded LDA of EL0600 - Business Disruption and System Failures	7.2
Perplexity of EL0600 - Business Disruption and System Failures	1.85
Topics probability of EL0600 - Business Disruption and System Failures	0.05
Averaging topic probabilities of EL0600 - Business Disruption and System Failures	125.73
Wordcloud of EL0600 - Business Disruption and System Failures	7.68
Assign clusters to descriptions of EL0600 - Business Disruption and System Failures	0.71
Select data and seeds of EL0700 - Execution, Delivery & Process Management	0.08
Estimate seeded LDA of EL0700 - Execution, Delivery & Process Management	15.82
Perplexity of EL0700 - Execution, Delivery & Process Management	7.94
Topics probability of EL0700 - Execution, Delivery & Process Management	0.43
Averaging topic probabilities of EL0700 - Execution, Delivery & Process Management	1350.52
Wordcloud of EL0700 - Execution, Delivery & Process Management	11.3
Assign clusters to descriptions of EL0700 - Execution, Delivery & Process Management	10.38
Save all results by Event Type	21.89
Plot PCA by ET with clusters	0.86
Plot LSA by ET with clusters	5.1
Plot UMAP by ET with clusters	4.84
Total	15528.43

The total computational time for the script `TextAnalysis2.R` is around 4 hours and 20 minutes. The most computationally intensive parts are related to the semantic adjustment (around 50 minutes), and the averaging topic probabilities of equal document vectors for event type “Clients, Products & Business Practices” (around 1 hour and 40 minutes).

C.2 ReportTextAnalysis2.R

Details on computational time for the main steps of the script `ReportTextAnalysis2.R` are reported in Table C.4.

Table C.4: Computation time for `ReportTextAnalysis2.R` in seconds.

Step	Time (s)
Read data	0.73
PCA of Group	33.64
PCA of ETs	20.59
PCA features of Group	4.2
PCA features of ETs	15.57
PCA of ETs by cluster	22.07
Explained variance for PCA of ETs	13.03
LSA of Group	27.39
LSA of ETs	26.16
LSA features of Group	3.06
LSA features of ETs	4.83
LSA of ETs by cluster	26.06
UMAP of Group	19.48
UMAP of ETs	20.81
UMAP of ETs by cluster	21.07
Wordcloud of ETs by topic	106.33
Traceplots of ETs	1.88
Total	366.90

The total computational time for the script `ReportTextAnalysis2.R` is around 6 minutes.

C.3 TwitterAnalysis.R

Details on computational time for the main steps of the script `TwitterAnalysis.R` are reported in Table C.5. The calculation has been applied to a single day data set, referred to June 15th, which is composed of 146,195 tweets.

Table C.5: Computation time for `TwitterAnalysis.R` in seconds.

Step	Time (s)
Load tweets	39.86
Selection of relevant tweets	1.16
Clean the tweets	11.51
Tokenizing the tweets	11.83
Load dictionaries from OpRisk data	0.15
Exclude stopwords	5.74
Integrate n-grams	174.56
Define seed words	3.49
Document-by-term matrix	0.91
Statistics and wordcloud	2.56
Apply semantic adjustment	2274.31
Include cleaned tweets (based on tokenization) and select non-zero rows	12.59
PCA of tweets	778.7
LSA of tweets	30.06
Clean memory	2.49
UMAP of tweets	222.41
Estimate seeded LDA	313.8
Topics probability	10.55
Wordcloud	55.03
Assign clusters to tweets	4461.48
PCA by cluster	25.91
LSA by cluster	25.27
Save final results	23.56
UMAP by cluster	27.75
Total	8515.68

The total computational time for the script `TwitterAnalysis.R` is around 2 hours and 20 minutes. The most computationally intensive parts are related to the semantic adjustment (around 38 minutes), and the assignments of tweets to clusters, considering the constraint on seed words (around 1 hour and 15 minutes).

C.4 WrapperTwitterAnalysis.R

Since the script `WrapperTwitterAnalysis.R` recall `TwitterAnalysis.R` for each day of the time range, *i.e.*, from May 5th to July 12th, the computation time can be approximately estimated as the one of `TwitterAnalysis.R` (reported in Appendix C.3 for one day) times the number of days within the time range (*i.e.*, 69 days). The resulting approximated computation time is 587,581.92 seconds, equivalent to around 163 hours (*i.e.*, almost 7 days).

C.5 ReportTwitterAnalysis.R

Details on computational time for the main steps of the script `ReportTwitterAnalysis.R` are reported in Table C.6. The calculation has been applied to a data set of two days, referred to June 15th and 16th.

Table C.6: Computation time for `ReportTwitterAnalysis.R` in seconds.

Step	Time (s)
Plots of daily tweets by cluster	12.98
Plots of daily tweets for each cluster with peaks detection	9.14
Daily PCA	31.8
Daily PCA features	2.64
Daily PCA by cluster	28.98
Daily LSA	29.12
Daily LSA features	1.57
Daily LSA by cluster	28.22
Daily UMAP	35.23
Daily UMAP by cluster	27.73
Daily wordcloud by topic	37.81
Total	245.22

The total computational time for the script `ReportTwitterAnalysis.R` is around 4 minutes for two days. For each day of the time range, *i.e.*, from May 5th to July 12th, the computation time can be approximately estimated as the one previously obtained times the number of days within the time range (*i.e.*, 69 days), divided by two. The resulting approximated computation time is 8,460.09 seconds, equivalent to around 2 hours and a half.

Acknowledgment

I am profoundly grateful to my supervisor, Francesca, whose exceptional dedication and unwavering support have been the cornerstones of my doctoral journey. Francesca not only provided expert guidance but also demonstrated a genuine commitment to my academic and personal growth. Her meticulous attention to detail, insightful feedback, and encouragement have shaped this thesis and my overall research experience.

In addition, I extend my sincere appreciation to my co-supervisor, Ričardas, for his valuable contributions and collaborative approach, enriching the depth and breadth of my research, starting from the suggestion of this thesis title and going through all the related methodological aspects. Many thanks also to the Ričardas' Ph.D. student Peiheng for the fruitful sharing of the respective main research topics.

A special acknowledgment goes to my diligent colleague, Davide, who went above and beyond to follow every aspect of my research. His involvement and enthusiasm were truly commendable.

To OpRisk colleagues, in particular, Alberta, Alessandra, Barbara, Chiara, Francesca, Giuseppe, Laura, Marco G., Marco P., Oliver, Paola, Paolo, Ruben, Simona, and Valentina, I am grateful for your consistent support, in creating a collaborative environment that fostered learning and mutual growth.

I express my thanks to my managers, Ennio and Giandomenico, for providing leadership and creating a conducive space for academic exploration.

To UniCredit colleagues, Anna Rita, Fabio and Ivan, thank you for inspiring me to start this academic journey and greatly supporting me during all these years.

To my doctoral companions, Alessandro, Alice, Federico, Luca, Silvio, and Vincenzo, thank you for the shared experiences and camaraderie that made this journey memorable.

I also thank Roberto Boselli, Lorenzo Malandri, Fabio Mercorio, Luca Frigau, Maurizio Romano, and Gianpaolo Zammarchi for the valuable discussions on the topics argued in this thesis.

A special acknowledgment goes to my wife, Erika, whose patience, understanding, and unwavering support sustained me throughout this demanding journey. Her encouragement has been my pillar of strength.

I would like to express my gratitude to my parents and in-laws for their continuous encouragement and belief in my abilities. Their unwavering support has been a source of inspiration.

I am thankful to Grace and Lella for their companionship during academic endeavors. Your friendship added a meaningful dimension to all these years.

In closing, I acknowledge the collective efforts of everyone mentioned and countless others who, directly or indirectly, have contributed to the realization of this thesis. Your support has been invaluable, and I am truly grateful for the collaborative spirit that has defined this academic endeavor.

Bibliography

Aggarwal, C. C. (2012). *Mining text data*. Springer Science & Business Media. https://link.springer.com/book/10.1007/978-1-4614-3223-4

Anderlucci, L., Guastadisegni, L., & Viroli, C. (2019). Classifying textual data: shallow, deep and ensemble methods. https://doi.org/10.48550/arXiv.1902.07068

Anderlucci, L., & Viroli, C. (2020). Mixtures of dirichlet-multinomial distributions for supervised and unsupervised classification of short text data. *Advances in Data Analysis and Classification*, *14*, 759–770. https://doi.org/10.1007/s11634-020-00399-3

Arumugam, S., Gupta, S., Patra, B., Rajan, S., & Agarwal, S. (2016). Revealing Patterns within the Drilling Reports Using Text Mining Techniques for Efficient Knowledge Management. *All Days*, SPE-184062–MS. https://onepetro.org/SPEERM/proceedings-abstract/16ERM/All-16ERM/SPE-184062-MS/186896

Bach, M. P., Krsti, Z., Seljan, S., & Turulja, L. (2019). Text Mining for Big Data Analysis in Financial Sector: A Literature Review. *Sustainability*, *11*(5). https://doi.org/10.3390/su11051277

Bakarov, A. (2018). A survey of word embeddings evaluation methods. *CoRR*, *abs/1801.09536*. http://arxiv.org/abs/1801.09536

Basel Committee on Banking Supervision. (1988). *Basel I: International Convergence of Capital Measurement and Capital Standards*. Bank for International Settlement. https://www.bis.org/publ/bcbs04a.htm

Basel Committee on Banking Supervision. (1996). *Amendment to the Capital Accord to Incorporate Market Risks*. Bank for International Settlement. https://www.bis.org/press/p970918a.htm

Basel Committee on Banking Supervision. (2004). *International Convergence of Capital*

Measurement and Capital Standards. A Revised Framework. Bank for International Settlement. https://www.bis.org/publ/bcbs107.htm

Basel Committee on Banking Supervision. (2010). *Basel III: A Global Regulatory Framework for More Resilient Banks and Banking Systems*. Bank for International Settlement. https://www.bis.org/publ/bcbs189_dec2010.htm

Basel Committee on Banking Supervision. (2014). *Operational risk - Revisions to the simpler approaches*. Bank for International Settlement. https://www.bis.org/publ/bcbs291.htm

Basel Committee on Banking Supervision. (2016). *Standardized Measurement Approach for Operational Risk*. Bank for International Settlement. https://www.bis.org/bcbs/publ/d355.htm

Basel Committee on Banking Supervision. (2017). *Basel III: Finalising post-crisis reforms*. Bank for International Settlement. https://www.bis.org/bcbs/publ/d424.htm

Bazzarello, D., Crielaard, B., Piacenza, F., & Soprano, A. (2006). Modeling insurance mitigation on operational risk capital. *Journal of Operational Risk*, *1*(1), 57–65. https://doi.org/10.21314/JOP.2006.004

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., & Ginhoux, F. (2019). Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, *37*(1), 38–44. https://www.nature.com/articles/nbt.4314

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Benoit, K., Muhr, D., & Watanabe, K. (2021). *Stopwords: Multilingual stopword lists* [R package version 2.3]. https://CRAN.R-project.org/package=stopwords

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R Package for the Quantitative Analysis of Textual Data. *Journal of Open Source Software*, *3*(30), 1–4. https://doi.org/10.21105/joss.00774

Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, *1*(1), 17–35. https://projecteuclid.org/journals/annals-of-applied-statistics/

volume-1/issue-1/A-correlated-topic-model-of-Science/10.1214/07-AOAS114.full

Blei, D. M., & Lafferty, J. D. (2009). Topic models. In *Text mining: Classification, clustering, and applications* (pp. 71–94). CRC Press. https://www.taylorfrancis.com/chapters/edit/10.1201/9781420059458-12/topic-models-david-blei-john-la%EF%AC%80erty

Bodur, Z. (2012). Operational risk and operational risk related banking scandals/ large incidents. *Maliye Ve Finans Yazıları*, *1*(97), 64–86. https://dergipark.org.tr/en/pub/mfy/issue/16283/170776#article_cite

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00051/43387/Enriching-Word-Vectors-with-Subword-Information

Bollacker, K., Lawrence, S., & Giles, C. L. (1998). CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications, 116–123. https://doi.org/10.1145/280765.280786

Cano-Marin, E., Mora-Cantallops, M., & Sánchez-Alonso, S. (2023). Twitter as a predictive system: A systematic literature review. *Journal of Business Research*, *157*, 113561. https://www.sciencedirect.com/science/article/pii/S0148296322010268

Carreño, L. V. (2013). *Operational risk modeling in financial services*. John Wiley & Sons. https://www.wiley.com/en-fr/Operational+Risk+Modeling+in+Financial+Services:+The+Exposure,+Occurrence,+Impact+Method-p-9781119508502

Carrivick, L., & Westphal, A. (2019). Machine learning in operational risk Making a business case for its practical implementation. *ORX Association*. https://orx.org/resource/machine-learning-in-op-risk?hsCtaTracking=ee538522-b9da-4225-8322-ef4ada5d3aae%7C4545edb7-cf8a-4984-8569-99d2b2138f62

Chakrabarti, A., Ni, Y., & Mallick, B. (2023). *Bayesian flexible modelling of spatially resolved transcriptomic data*. https://typeset.io/papers/bayesian-flexible-modelling-of-spatially-resolved-8ii6e2dp

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in Neural Information Processing Systems (NIPS)*, 288–296. https://proceedings.neurips.cc/paper_files/paper/2009/file/f92586a25bb3145facd64ab20fd554ff-Paper.pdf

Chen, J., & Gupta, A. K. (2012). *Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance; 2nd ed.* Springer. https://doi.org/10.1007/978-0-8176-4801-5

Chu, C.-Y., Park, K., & Kremer, G. E. (2020). A global supply chain risk management framework: An application of text-mining to identify region-specific supply chain risks. *Advanced Engineering Informatics*, *45*, 101053. https://doi.org/https://doi.org/10.1016/j.aei.2020.101053

CNN. (2023). *Exclusive: US government agencies hit in global cyberattack.* https://edition.cnn.com/2023/06/15/politics/us-government-hit-cybeattack/index.html

Cohen, A. M., Hersh, W. R., & Dubay, C. (2004). A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, *6*(1), 57–71. https://academic.oup.com/bib/article/6/1/57/288914?login=false

Cope, E., Mignola, G., Antonini, G., & Ugoccioni, R. (2009). Challenges in measuring operational risk from loss data. *Journal of Operational Risk*, *4*(4), 3–27. https://doi.org/10.21314/JOP.2009.069

Cornwell, N., Bilson, C., Gepp, A., Stern, S., & Vanstone, B. J. (2023). The role of data analytics within operational risk management: A systematic review from the financial services and energy sectors. *Journal of the Operational Research Society*, *74*(1), 374–402. https://doi.org/10.1080/01605682.2022.2041373

Costola, M., Iacopini, I., & Santagiustina, C. R. (2021). On the "mementum" of meme stocks. *Economics Letters*, *207*, 110021. https://www.sciencedirect.com/science/article/pii/S0165176521002986

Dal Pozzolo, A., & Bontempi, G. (2015). Adaptive machine learning for credit card fraud detection. https://api.semanticscholar.org/CorpusID:111359889

Danesi, I., Piacenza, F., Ruli, E., & Ventura, L. (2016). Optimal B-robust posterior distributions for operational risk. *Journal of Operational Risk*, *11*(4), 1–20. https://doi.org/10.21314/JOP.2016.182

Data Study Group team. (2019). Data study group final report: Global bank. https://doi.org/10.5281/zenodo.2557809

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep

Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:52967399

Dhillon, I., & Modha, D. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, *42*, 143–175. https://doi.org/10.1023/A:1007612920971

Di Vincenzo, D., Greselin, F., Piacenza, F., & Zitikis, R. (2023). A text analysis of operational risk loss descriptions. *Journal of Operational Risk*, *18*(3), 63–90. https://doi.org/10.21314/JOP.2023.003

D'Ippolito, M., Anderlucci, L., & Viroli, C. (2021). *Deepmou: Clustering of short texts by mixture of unigrams and its deep extensions* [R package version 0.1.1]. https://CRAN.R-project.org/package=deepMOU

Dumais, S., Furnas, G., Landauer, T., Deerwester, S., & Harshman, R. (1988). Using Latent Semantic Analysis to Improve Access to Textual Information. *CHI'88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 281–285. https://doi.org/https://dl.acm.org/doi/10.1145/57167.57214

Eler, D. M., Grosa, D., Pola, I., Garcia, R., Correia, R., & Teixeira, J. (2018). Analysis of Document Pre-Processing Effects in Text and Opinion Mining. *Information*, *9*(4). https://doi.org/10.3390/info9040100

European Banking Authority. (2022). *Risk assessment of the european banking system. december 2022*. Publications Office of the European Union. https://doi.org/doi/10.2853/04972

European Parliament and Council of the European Union. (2013). *Capital Requirement Regulations*. Technical Report. https://eur-lex.europa.eu/eli/reg/2013/575/oj

European Parliament and Council of the European Union. (2016). *General Data Protection Regulation*. Technical Report. https://eur-lex.europa.eu/eli/reg/2016/679/oj

European Parliament and Council of the European Union. (2018). *Regulatory technical standards of the specification of the assessment methodology under which competent authorities permit institutions to use AMA for operational risk*. https://eur-lex.europa.eu/eli/reg%7B%5C_%7Ddel/2018/959/oj

European Parliament and Council of the European Union. (2023). *Proposal for a regulation of the european parliament and of the council amending regulation (eu) no 575/2013 as* *regards requirements for credit risk, credit valuation adjustment risk, operational risk, market risk and the output floor*. https://data.consilium.europa.eu/doc/document/ST-15883-2023-INIT/en/pdf

Feldman, R., & Sanger, J. (2006). The text mining handbook: Advanced approaches in analyzing unstructured data. *Cambridge University Press*. https://www.cambridge.org/core/books/text-mining-handbook/0634B1DF14259CB43FCCF28972AE4382

Ferner, C., Havas, C., Birnbacher, E., Wegenkittl, S., & Resch, B. (2020). Automated seeded latent dirichlet allocation for social media based event detection and mapping. *Information*, *11*(8). https://doi.org/10.3390/info11080376

Frachot, A., Georges, P., & Roncalli, T. (2001). Loss Distribution Approach for operational risk. http://www.thierry-roncalli.com/download/lda-practice.pdf

Frachot, A., Moudoulaud, O., & Roncalli, T. (2007). Loss distribution approach in practice. In M. Ong & J. Hashagen (Eds.), *The Basel handbook: A guide for financial practitioners* (pp. 527–555). Risk Books, London. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1032592

Francopoulo, G., & Schaub, L.-P. (2020). Anonymization for the GDPR in the Context of Citizen and Customer Relationship Management and NLP. *Workshop on Legal and Ethical Issues*, 9–14. https://hal.archives-ouvertes.fr/hal-02939437

Fraser, J., & Simkins, B. J. (2002). *Enterprise risk management: Today's leading research and best practices for tomorrow's executives*. John Wiley & Sons. https://wiley.com/en-ie/Enterprise+Risk+Management%3A+Today%27s+Leading+Research+and+Best+Practices+for+Tomorrow%27s+Executives%2C+2nd+Edition-p-9781119741480

Friedman, C., Alderson, P. O., Austin, J. H. M., Cimino, J. J., & Johnson, S. B. (1998). Natural language processing in radiology: A systematic review. *Journal of the American Medical Informatics Association*, *5*(6), 515–525. https://pubs.rsna.org/doi/abs/10.1148/radiol.16142770?journalCode=radiology

Frigau, L., Wu, Q., & Banks, D. (2021). Optimizing the jsm program. *Journal of the American Statistical Association*, *117*(538), 617–626. https://doi.org/10.1080/01621459.2021.1978466

Fritz, H., Garcia-Escudero, L. A., & Mayo-Iscar, A. (2012). tclust: An R package for a trimming approach to cluster analysis. *Journal of Statistical Software*, *47*(12), 1–26. https://doi.org/10.18637/jss.v047.i12

Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Nobani, N. (2022). Embeddings evaluation using a novel measure of semantic similarity. *COGNITIVE COMPUTATION*, *14*(2), 749–763. https://doi.org/10.1007/s12559-021-09987-7

Greselin, F., Piacenza, F., & Zitikis, R. (2019). Practice oriented and monte carlo based estimation of the value-at-risk for operational risk measurement. *Risks*, *7*(2). https://doi.org/10.3390/risks7020050

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*(Suppl 1), 5228–5235. https://www.pnas.org/doi/full/10.1073/pnas.0307752101

Grün, B., & Hornik, K. (2011a). Topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30. https://doi.org/10.18637/jss.v040.i13

Grün, B., & Hornik, K. (2011b). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30. https://doi.org/10.18637/jss.v040.i13

Grün, B., & Hornik, K. (2023). *Topicmodels: Topic models* [R package version 0.2-15]. https://CRAN.R-project.org/package=topicmodels

Gupta, A., & Gupta, D. (2018). Social media text mining and network analysis for decision support in healthcare: A survey. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 108–113. https://www.researchgate.net/publication/264549082_Social_Media_Text_Mining_and_Network_Analysis_for_Decision_Support_in_Natural_Crisis_Management

Harris, Z. (1954). Distributional Structure. *Word*, *10*(2-3), 146–162. https://www.tandfonline.com/doi/pdf/10.1080/00437956.1954.11659520

Hinton, G. E., & Roweis, S. (2002). Stochastic neighbor embedding. In S. Becker, S. Thrun & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15). MIT Press. https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf

Holton, C. (2009). Identifying disgruntled employee systems fraud risk through text mining: A simple solution for a multi-billion dollar problem. *Decision Support Systems*, *46*(4), 853–864. https://doi.org/10.1016/j.dss.2008.11.013

Hong, L., & Davison, B. D. (2010). Empirical study of topic modeling in twitter. *Proceedings of the First Workshop on Social Media Analytics*, 80–88. https://dl.acm.org/doi/10.1145/1964858.1964870

Hornik, K., Feinerer, I., Kober, M., & Buchta, C. (2012). Spherical k-means clustering. *Journal of Statistical Software*, *50*(10), 1–22. https://doi.org/10.18637/jss.v050.i10

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(11), 417–441. https://doi.org/10.1037/h0071325

Iacopini, I., & Santagiustina, C. R. (2021). Filtering the intensity of public concern from social media count data with jumps. *Journal of the Royal Statistical Society Series A: Statistics in Society*, *184*(4), 1283–1302. https://academic.oup.com/jrsssa/article/184/4/1283/7068846

Jagarlamudi, J., Daumé III, H., & Udupa, R. (2012, April). Incorporating lexical priors into topic models. In W. Daelemans (Ed.), *Proceedings of the 13th conference of the European chapter of the association for computational linguistics* (pp. 204–213). Association for Computational Linguistics. https://aclanthology.org/E12-1021

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., & Lindén, K. (2019). Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, *65*, 675–782. https://doi.org/10.48550/arXiv.1804.08186

Ji, Z., Duan, X., Pons, D., Chen, Y., & Pei, Z. (2023). Integrating text mining and analytic hierarchy process risk assessment with knowledge graphs for operational risk analysis. *Journal of Operational Risk*, *18*(3), 31–61. https://doi.org/10.21314/JOP.2023.004

Jones, T. (2021). *Textminer: Functions for text mining and topic modeling* [R package version 3.0.5]. https://CRAN.R-project.org/package=textmineR

Jurafsky, D., & Martin, J. H. (2019). *Speech and language processing*. Pearson. https://web.stanford.edu/~jurafsky/slp3/

Kaliyar, R. K. (2020). A multi-layer bidirectional transformer encoder for pre-trained word embedding: A survey of bert. *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 336–340. https://doi.org/10.1109/Confluence47617.2020.9058044

Kannan, S., & Somasundaram, K. (2017). Autoregressive-based outlier algorithm to detect money laundering activities. *Journal of Money Laundering Control*, *20*(2), 190–202. https://doi.org/10.1108/JMLC-07-2016-0031

Kearney, M. W. (2019). Rtweet: Collecting and analyzing twitter data [R package version 0.7.0]. *Journal of Open Source Software*, *4*(42), 1829. https://doi.org/10.21105/joss.01829

Khrestina, M. P., Dorofeev, D. I., Kachurina, P. A., Usubaliev, T. R., & Dobrotvorskiy, A. S. (2017). Development of algorithms for searching, analyzing and detecting fraudulent activities in the financial sphere. *European Research Studies Journal*, *20*(4B), 484–498. https://ersj.eu/journal/905#

Khyani, D., & Siddhartha, B. S. (2021). An interpretation of lemmatization and stemming in natural language processing. *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, *22*, 350–357. https://jusst.org/an-interpretation-of-lemmatization-and-stemming-in-natural-language-processing/

Killick, R., & Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal of Statistical Software*, *58*(3), 1–19. https://www.jstatsoft.org/article/view/v058i03

Killick, R., Haynes, K., & Eckley, I. A. (2022). *changepoint: An R package for changepoint analysis* [R package version 2.2.4]. https://CRAN.R-project.org/package=changepoint

Klopotan, I., Zoroja, J., & Meško, M. (2018). Early warning system in business, finance, and economics: Bibliometric and topic analysis. *International Journal of Engineering Business Management*, *10*, 1847979018797013. https://journals.sagepub.com/doi/10.1177/1847979018797013

Lam, J. (2003). Enterprise risk management: From incentives to controls. *John Wiley & Sons*. https://www.wiley.com/en-ae/Enterprise+Risk+Management:+From+Incentives+to+Controls,+2nd+Edition-p-9781118413616

Lambrigger, D., Shevchenko, P., & Wüthrich, M. (2007). The quantification of operational risk using internal data, relevant external data and expert opinion. *Journal of Operational Risk*, *2*(3), 3–27. https://doi.org/10.21314/JOP.2007.030

Leidner, J., & Schilder, F. (2010, July). Hunting for the black swan: Risk mining from text. In S. Kübler (Ed.), *Proceedings of the ACL 2010 system demonstrations* (pp. 54–59). Association for Computational Linguistics. https://aclanthology.org/P10-4010

Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. *Risks*, *7*(1). https://doi.org/10.3390/risks7010029

Livemint. (2023). *Biden accused of taking USD 5 million bribe from Ukrainian firm: Report*. https://www.livemint.com/news/world/biden-accused-of-taking-5-million-bribe-from-ukrainian-firm-report-11686412503150.html

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, *1*(4), 309–317. https://doi.org/10.1147/rd.14.0309

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297. https://projecteuclid.org/ebook/Download?urlid=bsmsp%5C%2F1200512992&isFullBook=False

McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv e-prints*. https://doi.org/10.48550/arXiv.1802.03426

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, *3*(29), 861. https://doi.org/10.21105/joss.00861

Melville, J. (2023). *Uwot: The uniform manifold approximation and projection (umap) method for dimensionality reduction* [R package version 0.1.16]. https://CRAN.R-project.org/package=uwot

meta::cpan. (2021). *Lingua::StopWords - Stop words for several languages*. https://metacpan.org/pod/Lingua::StopWords

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013a). Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient Estimation of Word Representations in Vector Space. *In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. https://doi.org/10.48550/arXiv.1301.3781

Milmo, D. (2023). *Threads app: Instagram owner's Twitter rival logs 5 million users in first hours*. https://www.theguardian.com/technology/2023/jul/06/meta-launches-twitter-rivalthreads-in-100-countries

Mullen, L. (2021). *Gender: Predict gender from names using historical data* [R package version 0.6.0]. https://github.com/lmullen/gender

National Weather Service. (2023). *June 15, 2023 Severe Weather Tornadoes*. https://www.weather.gov/cle/event_20230615_severe

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108. https://dl.acm.org/doi/10.5555/1857999.1858011

Nigam, K., Mccallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, *39*, 103–134. https://doi.org/10.1023/A:1007692713085

Nlpl word embeddings repository. (2017). http://vectors.nlpl.eu/repository/

Ooms, J. (2022). *Cld2: Google's compact language detector 2* [R package version 1.2.4]. https://CRAN.R-project.org/package=cld2

ORX. (2023). *Operational Riskdata eXchange Association (ORX)*. Genève, Switzerland. https://orx.org/

ORX and Oliver Wyman. (2020). *ORX Reference Taxonomy for operational and non-financial risk - Causes Impacts - Summary report*. https://www.oliverwyman.com/our-expertise/insights/2020/nov/orx-reference-taxonomy-for-operational-and-non-financial-risk.html

ORX and Oliver Wyman. (2023). *ORX Reference Taxonomy for operational and non-financial risk – Guidance document. Version 2*. https://orx.org/download/orx-reference-taxonomy

ORX News. (2023). *ORX News Service*. Genève, Switzerland. https://orx.org/news

Pacella, M., Grieco, A., & Blaco, M. (2016). On the Use of Self-Organizing Map for Text Clustering in Engineering Change Process Analysis: A Case Study. *Computational Intelligence and Neuroscience*, *2016*, 1–11. https://doi.org/10.1155/2016/5139574

Pakhchanyan, S., Fieberg, C., Metko, D., & Kaspereit, T. (2022). Machine learning for categorization of operational risk events using textual description. *Journal of Operational*

Risk, *17*(4), 37–65. https://doi.org/10.21314/JOP.2022.026

Paulovich, F. V., Nonato, L. G., Minghim, R., & Levkowitz, H. (2008). Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE Transactions on Visualization and Computer Graphics*, *14*(3), 564–575. https://doi.org/10.1109/TVCG.2007.70443

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Pence, J., Farshadmanesh, P., Kim, J., Blake, C., & Mohaghegh, Z. (2020). Data-theoretic approach for socio-technical risk analysis: Text mining licensee event reports of u.s. nuclear power plants. *Safety Science*, *124*, 104574. https://doi.org/https://doi.org/10.1016/j.ssci.2019.104574

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Perry, J., & de Fontnouvelle, P. (2005, October). *Measuring Reputational Risk: The Market Reaction to Operational Loss Announcements* (Working Paper No. Current Draft: October 2005). Federal Reserve Bank of Boston. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=861364

Power, M. (2005). The invention of operational risk. *Review of International Political Economy*, *12*(4), 577–599. Retrieved January 20, 2024, from http://www.jstor.org/stable/25124039

Pun, J., & Lawryshyn, Y. (2012). Improving credit card fraud detection using a meta-classification strategy. *International Journal of Computer Applications*, *56*, 41–46. https://doi.org/10.5120/8930-3007

R Core Team. (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Ramos, M., Schiffer, L., Re, A., Azhar, R., Basunia, A., Rodriguez-Rodriguez, B., Lakhani, V., Bernal-Llinares, M., Bhagwat, N., & Bubier, J. (2020). UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLoS biology*, *18*(3), e3000589. https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1008432

Riesa, J., & Giuliani, I. (2013). Compact language detector 2. https://github.com/CLD2Owners/cld2

Roberts, M. E., Stewart, B. M., Tingley, D., & Airoldi, E. M. (2013). The structural topic model and applied social science. *International Conference on Neural Information Processing*. https://api.semanticscholar.org/CorpusID:59893873

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, *8*(1), 289–317. https://doi.org/10.32614/RJ-2016-021

Several authors. (2024). *The Comprehensive R Archive Network*. R Foundation for Statistical Computing. Vienna, Austria. https://cran.r-project.org/

Shanavas, N., Wang, H., Lin, Z., & Hawe, G. (2021). Knowledge-driven Graph Similarity for Text Classification. *International Journal of Machine Learning and Cybernetics*, *12*, 1067–1081. https://doi.org/10.1007/s13042-020-01221-4

Sharma, S., & Choudhury, A. R. (2021). Fraud analytics: A survey on bank fraud and fraud prediction using unsupervised learning based approach. *International Journal of Innovations in Engineering Research and Technology*, *3*(3), 1–9. https://repo.ijiert.org/index.php/ijiert/article/view/848

Shevchenko, P., & Wüthrich, M. (2006). The structural modelling of operational risk via bayesian inference: Combining loss data with expert opinions. *Journal of Operational Risk*, *1*(3), 3–26. https://doi.org/10.21314/JOP.2006.016

Sievert, C. (2020). *Interactive web-based data visualization with r, plotly, and shiny*. Chapman; Hall/CRC. https://plotly-r.com

Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin*, *24*. http://singhal.info/ieee2001.pdf

Soprano, A., Crielaard, B., Piacenza, F., & Ruspantini, D. (2010). *Measuring operational and reputational risk: A practitioner's approach*. Wiley. https://books.google.it/books?id=NS34Ep8-KAEC

Spärck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, *28*(1), 11–21. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.8343&rep=rep1&type=pdf

Sudjianto, A., Yuan, M., Kern, D., Nair, S., Zhang, A., & Cela-Diaz, F. (2010). Statistical methods for fighting financial crimes. *Technometrics*, *52*(1), 5–19. Retrieved January 14, 2024, from http://www.jstor.org/stable/40586676

Vaidya, A. H., & Mohod, S. W. (2014). Internet banking fraud detection using hmm and blast-ssaha hybridization. *International Journal of Science and Research (IJSR)*, *3*(7), 574–579. https://www.ijsr.net/archive/v3i7/MDgwNzE0MDU=.pdf

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

Viroli, C., & Anderlucci, L. (2021). Deep mixtures of unigrams for uncovering topics in textual data. *Statistics and Computing*, *31*(22), 1–18. https://doi.org/10.1007/s11222-020-09989-9

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C.-C. J. (2019). Evaluating word embedding models: Methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, *8*, e19. https://doi.org/10.1017/ATSIP.2019.12

Wang, C., Blei, D. M., & Li, F.-F. (2011). Simultaneous image classification and annotation. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1905–1912. http://vision.stanford.edu/pdf/WangBleiFei-Fei_CVPR2009.pdf

Wang, Y., Chang, Y., & Li, J. (2022). How does the pandemic change operational risk? Evidence from textual risk disclosures in financial reports. *Journal of Operational Risk*, *17*(3), 1–24. https://doi.org/10.21314/JOP.2022.017

Wang, Y., Li, G., Li, J., & Zhu, X. (2018). Comprehensive identification of operational risk factors based on textual risk disclosures [6th International Conference on Information Technology and Quantitative Management]. *Procedia Computer Science*, *139*, 136–143. https://doi.org/10.1016/j.procs.2018.10.229

Wijffels, J. (2021). *word2vec: Distributed Representations of Words* [R Package Version 0.3.4]. https://CRAN.R-project.org/package=word2vec

Wijffels, J., & Belmans, O. (2023). *Taskscheduler: Schedule r scripts and processes with the* *windows task scheduler* [R package version 1.8]. https://CRAN.R-project.org/package=taskscheduleR

Wikipedia. (2023). *Tornado outbreak sequence of June 14–19, 2023*. https://en.wikipedia.org/wiki/Tornado_outbreak_sequence_of_June_14%E2%80%9319,_2023

Youtube. (2023). *The June 15, 2023 Severe Weather Outbreak, As It Happened...* https://www.youtube.com/watch?v=ThL3-VzlWpc

Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier [International Conference on Computer, Communication and Convergence (ICCC 2015)]. *Procedia Computer Science*, *48*, 679–685. https://doi.org/https://doi.org/10.1016/j.procs.2015.04.201

Zhang, R., Xian, X., & Fang, H. (2019). The early-warning system of stock market crises with investor sentiment: Evidence from china. *International Journal of Finance & Economics*, *24*(1), 361–369. https://onlinelibrary.wiley.com/doi/abs/10.1002/ijfe.1667

Zhou, F., Qi, X., Xiao, C., & Wang, J. (2021). Metarisk: Semi-supervised few-shot operational risk classification in banking industry. *Information Sciences*, *552*, 1–16. https://doi.org/10.1016/j.ins.2020.11.027