PhD-FSTC-2025-005
Faculty of Science, Technology and Medicine

DISSERTATION

Defense held on 10 February 2025 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

BRAULIO C. BLANCO LAMBRUSCHINI
Born on 19th July 1986 in Huanuco (Peru)

CAPITALIZING ON TEXTUAL DATA TO PREDICT SME's MULTIDIMENSIONAL RISK

Dissertation defense committee

Dedicated to
my mother
Jovita Lambruschini

and
my father
Jorge Blanco

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor and supervisor, Dr. Mats Brorsson, for his invaluable guidance and constant encouragement throughout this research journey. I am also profoundly thankful to my committee members, Dr. Jacques Klein, Dr. Manxing Du, Dr. Zsofia Kräussl, and Dr. Radu State, for their thoughtful questions and valuable feedback during my thesis defense.

I am truly grateful to the friends I have met along this journey, as well as my lifelong friends, whose unwavering support and companionship have made this challenging process both manageable and fulfilling. To my colleagues, I extend my sincere appreciation for their camaraderie and stimulating discussions, which have been a constant source of motivation and inspiration.

I would like to express my heartfelt gratitude to the friends that I find during this journey and my long time friends, who have been by my side throughout this journey. Their support and friendship have made this challenging process not only manageable but also rewarding. To my colleagues, for their camaraderie and stimulating discussions that have motivated and inspired me.

I dedicate this work to my family, especially my parents and my brother, who have stood by me through every challenge and triumph. Their unwavering support, sacrifices, and love have been my greatest source of strength, and for that, I am forever grateful.

Lastly, I acknowledge the financial and institutional support from Yoba and its CEO, Jarno Partenen, without which this research would not have been possible.

This doctoral thesis was conducted at "The Interdisciplinary Centre for Security, Reliability, and Trust (SnT)" at the "University of Luxembourg" in partnership with Yoba S.A.

This research was funded in whole or in part by the Luxembourg National Research Fund (FNR), grant reference 15403349. For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

ABSTRACT

This thesis examines how various data sources impact the risk profiles of small and medium-sized enterprises (SMEs) in Luxembourg. Effective risk analysis and prediction is crucial for societal development, as it enhances decision-making processes related to the allocation of financial resources. By identifying and managing risks, businesses can make informed investment choices, fostering growth and optimizing returns. This contributes not only to the success of individual enterprises but also to broader economic stability and development. Most current risk prediction methods rely on financial statements, ratios, and other numerical data. Some researchers also analyze short text pieces, such as tweets, news articles, and headlines. Additionally, certain studies focus on specific sections of annual reports that discuss risk evaluation. However, these approaches are often limited to large companies.

Financial data is often seen as providing a limited perspective on a company's risk level. Therefore, it is essential that additional variables influencing risk be considered. Many factors within a business can contribute to risk. In this context, based on the data that can be collected, extracted, and generated, potential risk information will be derived from text-based insights found in annual reports, people networks, and geographic location. The proposed multidimensional risk model incorporates diverse information from reliable sources, such as the Luxembourgish Business Registry (LBR).

The current work is presented following a data pipeline process. Information is extracted using data provided by the partner company, with textual content obtained from various official company documents through OCR and PDF reading tools. Relationships between companies, audit firms, auditors, and notaries are created using extracted information from textual sources and additional datasources. The proposed Long-Text BERT model is applied to predict the risk of bankruptcy based on the annexes from annual accounts, while also categorizing pages for subsequent information extraction using a fine-tuned GPT-based model. With a proposed autoclustering algorithm, clusters of hidden accountants or consultancy firms were identified and added in the company people's network. Geolocation is performed using addresses found through information extraction and those registered in the company profile within the LBR, from which latitude and longitude coordinates are obtained. This information is integrated into a graph network, where companies relationships are analyzed to identify various risk factors and complement the text-based risk assessment.

As a first outcome of this thesis, a dataset containing both financial and non-financial information was created. This existing data was enriched using NLP tools, and a network of companies and individuals was established. Additionally, valuable insights were extracted from the textual information, achieving approximately 80% precision in risk prediction based solely on the textual data from financial annexes. Companies were also clustered based on the “hidden accountant” concept. Information from various data sources is integrated using graphs to calculate dimensional risk for each company. An initial user interface has been proposed to enable users to navigate and explore some data more effectively.

In conclusion, this thesis successfully developed a data pipeline to process information from Luxembourgish SMEs, leveraging publicly available information. The data was enriched using advanced Machine Learning and Deep Learning techniques to assess company risk from multiple dimensions. This approach provides decision-makers with deeper insights, enabling more informed and strategic decisions. The findings suggest that these models can be adapted for application in other countries or scaled to analyze larger enterprises. Furthermore, the analysis could be significantly enhanced by integrating additional datasources, such as social networks, and employing more sophisticated methods like Graph Neural Networks for data integration.

Keywords: Finance, Risk, NLP, GPT, BERT, Embeddings, Graphs, SMEs

RESEARCH OBJECTIVES

The primary objective of this doctoral thesis is to analyze and evaluate the influence of textual information in assessing various dimensions of risk within small and medium-sized enterprises (SMEs) in Luxembourg,

The specific objectives of the study are:

- To create a text dataset comprising both financial and non-financial data regarding companies in Luxembourg, ensuring compliance with GDPR regulations.
- To enrich the non-financial information with valuable insights from company reports and other textual sources.
- To enhance risk assessment for companies by integrating non-financial data across various facets to provide a more holistic understanding of risk.
- To provide a user-friendly platform for efficient data management and intuitive display of information.

Contents

Acknowledgments . . . iii
General Abstract . . . iv
Research Objectives . . . vi

1 Introduction **1**
1.1 Motivation and Challenges . . . 1
1.2 Research Questions . . . 5
1.3 Scope and Limitations . . . 5
1.4 Main Contributions . . . 5
1.5 Structure of the Thesis . . . 7
1.6 Published Papers . . . 11

2 Theoretical Background **14**
2.1 Introduction . . . 14
2.2 Company Risk Prediction . . . 15
2.3 Natural Language Processing . . . 17
2.4 Deep Learning Models (DNN) . . . 19
2.5 Attention Mechanism and Transformers . . . 19
2.6 Encoder-based models: BERT . . . 21
2.7 Decoder-based models: GPT . . . 23
2.8 Tokenization and Embeddings . . . 26
2.9 Information Extraction . . . 29
2.10 Document Clustering . . . 32
2.11 Entity Resolution . . . 33
2.12 Graph Analysis . . . 35
2.13 Summary . . . 37

3 Methodology and Proposed Architecture **39**
3.1 Introduction . . . 40
3.2 Research Design . . . 40
3.3 Dataset . . . 40
3.3.1 European Commission: List of NACE codes . . . 41

3.3.2 Luxembourg Business Registers (LBR) 41
3.3.3 Institute of Business Auditors (IRE) 44
3.3.4 Chambre des notaires du Grand-Duche de Luxembourg 45
3.4 General Methodology 45
3.4.1 Data Gathering and Refining 46
3.4.2 Information Enrichment 47
3.4.3 Data Analysis and Insights Generator 47
3.4.4 Presentation and Knowledge Exploitation 47
3.5 Component-Specific Methodologies 48
3.5.1 Basic Software Engineering 48
3.5.2 Data Exploratory 49
3.5.3 Supervised and unsupervised Learning 50
3.5.4 Agile 53
3.6 Technologies and Proposed Architecture 55
3.6.1 Application Architecture View 55
3.6.2 Deployment View 58
3.7 Summary 61

4 Data Gathering and Refining **62**
4.1 Introduction 62
4.2 Data Collection 63
4.2.1 European Commission: List of NACE codes 63
4.2.2 Luxembourg Business Registers (LBR) 65
4.2.3 Institute of Business Auditors (IRE) 66
4.2.4 Chambre des notaires du Grand-Duche de Luxembourg 66
4.3 PDF downloader 67
4.4 OCR, extraction and cleaning 68
4.5 Data ingestion 70
4.6 Summary 71

5 Information Enrichment **72**
5.1 Introduction 72
5.2 Page-level Labeler 74
5.3 Financial Tokenizer 74
5.3.1 Generation of new terms 75
5.3.2 Embeddings training 76
5.4 Page type classifier 76
5.4.1 Long-text BERT Architecture 77
5.4.1.1 Pre-processing 77
5.4.1.2 Segment Analysis 79

5.4.1.3 Integrated prediction 80
5.4.2 Page type classifier using Long-text BERT 83
5.5 Information Extraction 84
5.5.1 Dataset 85
5.5.2 Fine-tuning process 88
5.5.3 Model's Evaluation and model selection 91
5.6 Address Geolocation 95
5.7 Entity Resolution: Embedding Generation and Entity Insertion . . . 96
5.7.1 Record Embeddings 96
5.7.1.1 Embeddings Generator Tool 97
5.7.2 Embeddings strategies 99
5.7.3 Dataset of Companies 100
5.7.4 Dataset of People 102
5.7.5 Testing and Results 103
5.8 Summary 106

6 Data Analysis and Insights Generator 107
6.1 Introduction 107
6.2 Authors fingerprint autoclustering 108
6.2.1 Autoclustering Algorithm 109
6.2.1.1 Phase I: Generation of cluster candidates 110
6.2.1.2 Phase II: Cluster feature's cleaning 113
6.2.1.3 Phase III: Cluster merging 114
6.2.1.4 Phase IV: Sub clustering 115
6.2.1.5 Results and Evaluation 115
6.3 Text-based bankruptcy prediction 116
6.4 Companies' Cluster Analysis 119
6.4.1 Data ingestion 119
6.4.2 Node Graph Centrality 126
6.5 Graph Risk Analysis: Multidimensional Risk Proposed Model 127
6.5.1 Proposed dimensions 128
6.5.1.1 D1: Management Risk 128
6.5.1.2 D2: Partner Relationship Risk 129
6.5.1.3 D3: Environmental and Industry Risk 130
6.5.1.4 D4: Sector Risk 132
6.5.1.5 D5: Financial Reporting Risk 133
6.5.2 Proposed sub-dimensions 133
6.5.3 Evaluation of the Integrated proposed multidimensional Risk Model 134

6.5.3.1 Dataset . . . 135
6.5.3.2 Models and metrics to Evaluate . . . 135
6.5.3.3 Comparison of models performance . . . 136
6.5.3.4 Final weighted proposition . . . 137
6.5.4 Multidimensional Risk Report . . . 138
6.6 Summary . . . 140

7 Data Presentation and exploitation 141
7.1 Introduction . . . 141
7.2 Business Dashboard . . . 142
7.2.1 Proof of Concept . . . 142
7.2.2 Solution Design . . . 146
7.2.3 Implementation, Testing and Deployment . . . 147
7.3 Implemented Dashboard . . . 149
7.4 Summary . . . 152

8 Results and Discussions 153
8.1 Introduction . . . 153
8.2 Results . . . 153
8.2.1 Data Gathering and Refining . . . 153
8.2.2 Information Enrichment . . . 154
8.2.3 Data Analysis and Risk Assessment . . . 154
8.2.4 Data presentation and Exploitation . . . 155
8.3 Discussion . . . 155
8.4 Recommendations for Future Research . . . 157

9 Conclusions 158

Bibliography 165

List of Figures

1.1 Introduction: Thesis Structure 8

2.1 Background: Transformer-model architecture 20
2.2 Background: Transformer-model architecture: Encoder 21
2.3 Background: The Transformer-model architecture: Decoder 24

3.1 Methods: LBR:Number of registered SMEs per year 42
3.2 Methods: Data pipeline process 46
3.3 Methods: Software Engineering methodology 48
3.4 Methods: Data Exploratory Methodology 50
3.5 Methods: Supervised and Unsupervised Learning Methodology 51
3.6 Methods: Agile Methodology 54
3.7 Methods: Applications Architecture Overview 56
3.8 Methods: Deployment View 59

4.1 Data Gathering and Refining: Data Pipeline 63
4.2 Data Gathering and Refining: NACE code list 64
4.3 Data Gathering and Refining: Screenshot LBR website with a company profile 65
4.4 Data Gathering and Refining: Screenshot IRE website 67
4.5 Data Gathering and Refining: Screenshot of Chambre des notaires website 67
4.6 Data Gathering and Refining: OCR Extractor tool workflow 69
4.7 Data Gathering and Refining: Balance Sheet from LBR Example . . 71

5.1 Information Enrichment: Data Pipeline 73
5.2 Information Enrichment: Web Application of Page-Level Labeler . . . 74
5.3 Information Enrichment: Long-Text BERT: General Architecture . . 78
5.4 Information Enrichment: Long-Text BERT: Pre-processing phase . . 78
5.5 Information Enrichment: Long-Text BERT: Segment Analysis 80
5.6 Information Enrichment: Long-Text BERT: Integrated Prediction - 1st Approach 81

5.7 Information Enrichment: Long-Text BERT: Integrated Prediction - 2nd Approach . . . 81
5.8 Information Enrichment: Long-Text BERT: Integrated Prediction - 3rd Approach . . . 82
5.9 Integrated Prediction: Long-Text BERT: Test Configurations . . . 83
5.10 Integrated Prediction: Long-Text BERT: Results for not long sequences of texts . . . 84
5.11 Integrated Prediction: Text Extractor: Fine-tuning and testing process 89
5.12 Information Enrichment: ER Embeddings . . . 98
5.13 Information Enrichment: ER Embeddings Generation Process . . . 99
5.14 Information Enrichment: ER process for companies . . . 101
5.15 Information Enrichment: ER process for people . . . 103
5.16 Information Enrichment: results ER for companies and people datasets 105

6.1 Data Analysis and Insight Generator: Data Pipeline . . . 108
6.2 Data Analysis and Insight Generator: Autoclustering algorithm phases 110
6.3 Data Analysis and Insight Generator: Autoclustering features . . . 112
6.4 Data Analysis and Insight Generator: Long-Text BERT Bankuptcy prediction: Results with different LSTM hidden layers . . . 117
6.5 Data Analysis and Insight Generator: Long-Text BERT Bankuptcy prediction: Results with different Sequence Lenghts . . . 118
6.6 Data Analysis and Insight Generator:Graph Analysis - Database model 119
6.7 Data Analysis and Insight Generator: Graph Analysis - Graph Model 120
6.8 Data Analysis and Insight Generator: Upload workflow model . . . 120
6.9 Data Analysis and Insight Generator: Graph Ingestion server architecture . . . 121
6.10 Data Analysis and Insight Generator: Graph Examples: Industries . . 122
6.11 Data Analysis and Insight Generator: Graph Examples: Addresses . . 123
6.12 Data Analysis and Insight Generator: Graph Examples: Audit Firms 123
6.13 Data Analysis and Insight Generator: Graph Examples: Auditors . . 124
6.14 Data Analysis and Insight Generator: Graph Examples: Notaries . . 124
6.15 Data Analysis and Insight Generator: Graph Examples: Managers/Shareholders . . . 125
6.16 Data Analysis and Insight Generator: Graph Examples: Accountant Clusters . . . 125
6.17 Data Analysis and Insight Generator: Graph Examples: Network example . . . 126
6.18 Data Analysis and Insight Generator: Multidimensional Risk: Proposition . . . 127

6.19 Data Analysis and Insight Generator: Multidimensional Risk: Partner Relationship Risk (Shareholders, Managers) 129
6.20 Data Analysis and Insight Generator: Multidimensional Risk: Partner Relationship Risk(audit firms, auditors, notaries) 130
6.21 Data Analysis and Insight Generator: Multidimensional Risk: Environmental and Industry Risk 131
6.22 Data Analysis and Insight Generator: Multidimensional Risk: Environmental and Industry Risk: Map 132
6.23 Data Analysis and Insight Generator: Multidimensional Risk: Sector Risk 133
6.24 Data Analysis and Insight Generator: Multidimensional Risk: Subdimensions Importance 137
6.25 Data Analysis and Insight Generator: Multidimensional Risk: Proposed Report 1 139
6.26 Data Analysis and Insight Generator: Multidimensional Risk: Proposed Report 2 139
6.27 Data Analysis and Insight Generator: Multidimensional Risk: Proposed Report 3 140

7.1 Data Presentation: Business Dashboard: Main Organization Layout . 144
7.2 Data Presentation: Business Dashboard: main menu Layout 144
7.3 Data Presentation: Business Dashboard: Datasources Layout 145
7.4 Data Presentation: Business Dashboard: Industries Donut 146
7.5 Data Presentation: Business Dashboard: Companies List 147
7.6 Data Presentation: Business Dashboard: Company Profile 148
7.7 Data Presentation: Business Dashboard: Company Documents List . 148
7.8 Data Presentation: Business Dashboard: Company Location's map . 149
7.9 Data Presentation: Business Dashboard: Implemented Industry Map 150
7.10 Data Presentation: Business Dashboard: Implemented Companies List 150
7.11 Data Presentation: Business Dashboard: Implemented Companies List 151
7.12 Data Presentation: Business Dashboard: Implemented Companies List 151

List of Tables

2.1 Background: Number of maximum tokens per GPT-model 26

3.1 Methods: Company statistics per Industry 41
3.2 Methods: SME statistics per Industry 43
3.3 Methods: SME's Documents statistics 43
3.4 Methods: SME's documents statistics per Document Type 44
3.5 Methods: Distribution of Audit Firms and Auditors 44
3.6 Methods: Top Audit Firms in Luxembourg 45
3.7 Methods: Top Localities in Luxembourg by Number of certified notaries 45

4.1 Data Gathering and Refining: Most frequent filed documents in LBR 66

5.1 Information Enrichment: T-MuFIN Tokenizer: Results 76
5.2 Information Enrichment: Long-Text BERT: Tokenization: Distribution per language 77
5.3 Information Enrichment: Long-Text BERT: Integrated Prediction - Value of parameters of models 82
5.4 Information Enrichment: Information Extraction: Distribution of pages per type of document and language 85
5.5 Information Enrichment: Information Extraction: Labeled page type categories 85
5.6 Information Enrichment: Information Extraction: Tokenization: Distribution of labeled pages per type of content 86
5.7 Information Enrichment: Information Extraction: Frequent Entities and attributes from labeled dataset 87
5.8 Information Enrichment: Information Extraction: Tokenization: Dataset configurations for fine-tuning 90
5.9 Information Enrichment: Information Extraction: Distribution of testing pages per type of document 92
5.10 Information Enrichment: Information Extraction: Distribution of testing pages per language and document types 92

5.11 Information Enrichment: Information Extraction: Performance of baseline models and fine-tuned models 93
5.12 Information Enrichment: Information Extraction: Baseline model's performance results by type of content 94
5.13 Information Enrichment: Information Extraction: Baseline models Performance by document type 94
5.14 Information Enrichment: Information Extraction: Performance of the best performed semi-synthetic-based model with other languages and formats 95
5.15 Information Enrichment: Entity Resolution: 97
5.16 Information Enrichment: Information Extraction: Test case Scenarios for Entity Resolution 104
5.17 Information Enrichment: Information Extraction: Results ER for Companies 104
5.18 Information Enrichment: Information Extraction: Results ER for People 105

6.1 Data Analysis and Insight Generator: Autoclustering: Threshold parameters used in the algorithm 109
6.2 Data Analysis and Insight Generator: Autoclustering: Results 115
6.3 Data Analysis and Insight Generator: Autoclustering: Results - Rand Index 115
6.4 Data Analysis and Insight Generator: Risk Prediction: Results of Risk Precision of the models with respect to the different number of hidden layers 117
6.5 Data Analysis and Insight Generator: Graph Analysis: Potential Location Risk Edge types 122
6.6 Data Analysis and Insight Generator: Graph Analysis: Proposed subdimensions 134
6.7 Data Analysis and Insight Generator: Graph Analysis: Results evaluation 136
6.8 Data Analysis and Insight Generator: Graph Analysis: Ablation studies 136
6.9 Data Analysis and Insight Generator: Graph Analysis: Dimension Contribution Analysis 138

7.1 Data Presentation: Business Dashboard Requirements 142
7.2 Data Presentation: Business Dashboard Evaluation of web tools . . . 143

Acronyms

AI Artificial Intelligence. 15–18, 25

API Application Programming Interface. 24, 25

BERT Bidirectional Encoder Representations from Transformers. 6, 9, 21–23, 26, 28, 30, 37, 73, 76, 77, 79, 82, 116, 118

BPE Byte Pair Encoding. 21, 27

DEA Data Envelopment Analysis. 37

DL Deep Learning. 9, 15, 18, 19, 72, 84

DNN Deep Neural Networks. 19, 37

ECDF European Centralized Data Facility. 70

ER Entity Resolution. 33–35, 96, 100–102

EU European Union. 3

GDPR General Data Protection Regulation. vi, 5, 9

GPT Generative Pre-trained Transformer. 6, 23–26, 31, 37, 73, 84, 87, 116

GRU Gated Recurrent Unit. 19, 22

IFRS International Financial Reporting Standarts. 70

IRE Institut des Reviseurs d'Enterprises. 44, 66

IT Information Technology. 8, 9

JSON JavaScript Object Notation. 84

LBR Luxembourg Business Registers. 3, 41–43, 46, 61, 65, 71, 74, 85, 91, 96, 100, 119, 136

LightGBM Light Gradient Boosting Machine. 16, 134

LLM Large Language Model. 31, 88

LoRA Low-Rank Adapters. 31, 89

LSTM Long Short-Term Memory. 19, 23, 34, 83

MDR EU Mandatory Disclosure Rules. 3

MDS Multidimensional Scaling. 37

ML Machine Learning. 16, 18, 72, 134

MLP MultiLayer Perceptron. 134

MVP Minimum Viable Product. 54, 147

NACE Nomenclature of Economic Activities code. 41, 42, 63, 65, 70, 142, 145

NLP Natural Language Processing. 7, 16–18, 22, 37, 38, 116

NPA Non-performing asset prediction. 15

OCR Optical Character Recognition. 5, 33, 46, 47, 68, 69, 87

PCA Principal Component Analysis. 37

PEFT Parameter-Efficient Fine-Tuning. 31, 89

QLORA Quantized Low-Rank-Adapters. 31, 89

RBE Register of Beneficial Owners. 65

RCS Registre de Commerce et des Sociétés. 3, 65

RDBMS Relational Database Management System. 59

REGINSOL Insolvency Register. 65

RESA Electronic Register of Companies and Associations. 65

RNN Recurrent Neural Networks. 19, 34

SEC U.S. Securities and Exchange Commission. 17, 116

SMEs Small and medium-sized enterprises. vi, 2, 3, 5, 6, 15, 32, 42, 43, 48, 61, 95, 108, 135

SVM Supporting Vector Machines. 15, 134

Chapter 1

Introduction

Contents

1.1 Motivation and Challenges 1
1.2 Research Questions 5
1.3 Scope and Limitations 5
1.4 Main Contributions 5
1.5 Structure of the Thesis 7
1.6 Published Papers 11

1.1 Motivation and Challenges

The prosperity of national, regional, and global economies is significantly supported by favorable financial conditions extended to companies. These conditions include access to affordable credit, effective internal risk management, and comprehensive overall risk assessment[1].

Risk can be defined in various ways, but for the purposes of this work, the definition selected is from the Risk Management Framework of the Government of Canada[2]: "Risk refers to the uncertainty that surrounds future events and outcomes. It is the expression of the likelihood and impact of an event with the potential to influence the achievement of an organization's objectives.".

Most academic sources on risk assessment primarily focus on financial, operational, and reputational risks, or concentrate solely on the

[1]https://www.esm.europa.eu/blog/financial-stability-and-risks-growth-euro-area-where-do-we-stand

[2]https://www.tbs-sct.canada.ca/pol/doc-eng.aspx?id=12254

assessment process itself, without providing clear definitions of risk categories or dimensions. ISO 31000[3] provides a framework for Integrated Risk Management, recommending the analysis of various organizational factors in risk management. These factors include social, cultural, political, legal, regulatory, financial, technological, economic, and environmental aspects, as well as local, national, and international contexts. Additionally, the framework emphasizes considering stakeholders' needs, perceptions, values, and expectations, along with contractual relationships, the complexity of networks and dependencies, governance structures and others. Bernard's analysis of ISO 31000 [Bernard et al., 2002] emphasizes categorizing risks based on their origin, distinguishing between endogenous and exogenous risks. Endogenous risks are those where some control factors may be applied, while exogenous risks, by nature, are beyond the organization's ability to modify.

Therefore, it is highly recommended to include diverse sources of risks in the risk evaluation process. Even if frameworks, methodologies, and propositions are initially created for internal management, they can also be applied to external risk analysis, extending beyond financial risk assessments. As stated by Wong [2014], the introduction of non-financial risks, such as social and environmental risks, into the analysis process presents a significant challenge due to the complexity and variety of variables involved. However, incorporating non-financial risk management into the decision-making process can lead to substantial cost savings and new opportunities.

As a consequence, the research will be limited to non-financial variables for which data can be directly obtained or inferred using Machine Learning models, rather than considering all existing non-financial variables. The main risk dimensions considered in this research include environmental risk, the risk associated with the companys location, and social risk, which encompasses stakeholder relationships such as those with managers, shareholders, notaries, auditors, and accountants. This information can assist financial institutions in using these risk insights to offer cheaper credit to Small and medium-sized enterprises (SMEs) and optimize investment returns.

For a robust multidimensional risk analysis, much more information is required than what can be directly found in publicly available data sources. A proper analysis necessitates information in a structured for-

[3]https://www.iso.org/iso-31000-risk-management.html/

mat, yet most available data is unstructured and tends to be massive. In todays world, the substantial volume of collected and generated data is remarkable, with a clear trend towards exponential growth in the coming years. While cutting-edge technology for information processing is readily available to companies of all sizes, the significant investment required in terms of technology, time, and human resources often dissuades firms from leveraging the potential of this vast repository of valuable information.

While a substantial amount of data remains private, the availability of public data relevant to companies is expanding. This includes public posts from companies, employees, consumers, and news outlets, as well as financial reports and other legal documents mandated for disclosure by governments worldwide. For instance, in a bid to enhance transparency, Since July 1st, 2020, the European Union (EU) has mandated certain companies to disclose their annual reports as well as social and environmental risks EU Mandatory Disclosure Rules (MDR)).

In certain countries, such as Luxembourg, a wide range of company information is publicly available through the Registre de Commerce et des Sociétés (RCS), managed by the Luxembourg Business Registers (LBR) [4]. The published information includes annual accounts, articles of association, and details about directors, managers, shareholders, and more.

This research focuses on SMEs. According to the European Commission, a small company is defined as having fewer than 50 full-time employees and an annual turnover or balance sheet total of no more than 10 million. A medium-sized company has fewer than 250 full-time employees, with an annual turnover of no more than 50 million, or a balance sheet total not exceeding 43 million.

In Luxembourg, SMEs make up about 99.5% of all companies, employ approximately 66% of the workforce, and contribute around 66.4% of the country's total added value.[5]. In 2023, there were roughly 40,400 SMEs in Luxembourg, with micro firms (fewer than 10 employees) representing about 80% of them.[6].

As most risk assessment methods rely on financial reports and ratios, especially for SMEs, this limited information can lead to loan rejections

[4]https://www.lbr.lu/

[5]https://ec.europa.eu/

[6]https://www.statista.com/statistics/870475/number-of-smes-in-luxembourg-by-size/

and reduced investments, negatively impacting the economy. Therefore, this research aims to introduce additional variables for evaluation, providing decision-makers with more comprehensive information. To achieve this, it is essential to focus on the different phases of the data pipeline, from data collection to data presentation. This involves concentrating on data extraction, improving intermediate models to extract information and generate new insights or predictions, and providing the final risk assessment. The objectives of this thesis are to create a dataset from public government data sources and anonymize it to make it publicly available; train or fine-tune a model that can be implemented on a low-resource server to effectively extract entity information from unstructured and variable text; identify and link entities as the same across different data sources or unstructured data; enhance BERT-based models by feeding them with more information than their inherent limits; analyze text information from reports to identify hidden accountants within clusters of companies; predict the risk of bankruptcy for a company based on their reports without numerical information; and integrate different risk dimensions to provide a single risk report not only based on financial data.

Building on this foundation, the key insights derived from this research are as follows: The dataset can be effectively anonymized using a GPT-based tool, though not with 100% accuracy. Consequently, it is not yet possible to make the dataset publicly available without semi-manual verification. However, a low-resource GPT-based model was fine-tuned, which successfully extracted structured information from unstructured data with ≈78% of no hallucination and 100% of returning JSON structured information. Matching entities from two different sources was achieved with 80% accuracy for companies and individuals. A financial fine-tuned tokenizer was proposed, reducing the average number of tokens generated per word and increasing in almost 50% of data that can be fed into a BERT model. Additionally, two parallel BERT architectures were proposed to handle long sequences of text for categorization tasks. Using this model, the risk of bankruptcy was predicted using only textual information with a precision ≈73%, and the type of content in non-financial documents was predicted with a precision ≈97%. Furthermore, a clustering model was developed that, using content and context features, identified companies using similar templates in their Annual Reports with a Clustering Rand Index ≈87% and identified the hidden accountant with a Clustering Rand Index ≈55%. Finally, the different models were integrated into a single multidimensional risk model, which can be helpful to decision-makers.

1.2 Research Questions

- RQ1: Can financial datasets be effectively anonymized for public availability while adhering to GDPR requirements?
- RQ2: Can a low-resource GPT-based model be trained to efficiently extract structured data from unstructured data?
- RQ3: Can we effectively match entities that appear different due to insufficient information or OCR noise?
- RQ4: Can we effectively reduce the required number of tokens for BERT-based models without compromising their effectiveness?
- RQ5: Can we develop a method to cluster companies that likely share the same authors of financial documents by analyzing both their formatting style and content?
- RQ6: Can we predict bankruptcy risk for companies using only textual information?
- RQ7: Can we effectively identify risky companies by analyzing not only non-financial data?

1.3 Scope and Limitations

The thesis covers only SMEs in Luxembourg and is confined to publicly available data, ensuring compliance with General Data Protection Regulation (GDPR) regulations. Consequently, certain models and information are exclusively developed and employed for research purposes.

1.4 Main Contributions

The primary contributions of this thesis include the following:

- The Optical Character Recognition (OCR) tool extracts useful information from PDF documents using HTML or OCR, including language, page orientation, whether the document is scanned, encoded text, and other relevant details, leveraging various existing Python tools.

- A fine-tuned Generative Pre-trained Transformer (GPT)-based model extracts information from unstructured and diverse text into a two-level JSON format. Five metrics evaluate the performance of generative extractor models, with the best-performing model achieving approximately 78% accuracy in avoiding hallucinations and 100% accuracy in producing structured two-level outputs, while successfully eliminating empty fields.

- A new tokenization method for Entity Resolution matches identical instances from different data sources or information containing OCR errors. Reaching around 80% of accuracy for correctly identifying companies and people.

- Domain-specific tokenizers were proposed to fine-tune the Bidirectional Encoder Representations from Transformers (BERT) model, reducing the number of generated tokens per word and enabling approximately 50% more tokens to be processed by the model.

- An autoclustering algorithm clusters similar documents from the Annexes of the Annual Accounts by leveraging contextual and content information, including format and the authors fingerprint. The algorithm's effectiveness was evaluated using the Rand Index, achieving approximately 87% for first-level template-based clustering and 55% for *hidden author* clustering.

- Two new architectures leverage BERT to handle long or very long text sequences for categorization. These architectures predict bankruptcy risk using only textual information with a precision of approximately 73% and classify the type of page in non-financial documents with a precision of approximately 97%.

- An integrated multidimensional risk assessment framework for SMEs was developed, leveraging information extracted from diverse models. The framework was tested using a simple bankruptcy prediction model, achieving an F1-score of 89.70%. The proposed model incorporates five potential risk dimensions, weighted according to their average importance.

1.5 Structure of the Thesis

This thesis is proposed as a monographic thesis composed of four parts. Part I serves as an introduction, providing an overview of the thesis and contextualizing the state-of-the-art technologies relevant to its domain. Part II outlines the proposed methodology, covering the presentation and explanation of the dataset, the architectural design of the solution, and a thorough exploration of the roles assumed by selected technologies to meet the solution platform's requirements. Part III extensively details the thesis contributions, systematically tracing the progression from data gathering, cleaning, and enrichment, through data analysis and risk assessment, to the presentation of information within a user-friendly platform. Lastly, Part IV consolidates the primary findings from each contribution, engaging in a thorough examination of their impact in relation to the research questions. Concluding reflections are formulated, alongside identification of next steps to further advance the research agenda.

Outlined in Figure 1.1, the manuscript is structured across nine chapters, arranged as follows:

- Chapter 1: Introduction
  This chapter introduces the motivation behind the research and outlines the research questions addressed within this study. It also defines the scope and limitations considered in this research endeavor and delineates the contributions made to both the associated partners and the broader research community.

- Chapter 2: Theoretical Background
  Throughout the chapter, key concepts and relevant literature are introduced to provide a foundation for understanding the research. A concise overview of the evolution of pertinent technologies and commonly used terminology is presented. Specifically, the discussion focuses on technologies associated with Natural Language Processing (NLP), involving the processing of textual information across stages of data gathering, processing, and presentation.

- Chapter 3: Methodology and Proposed Architecture
  In this following chapter, the methodology employed in this research is detailed. The methodology includes a description of the activities undertaken to address the research questions and achieve

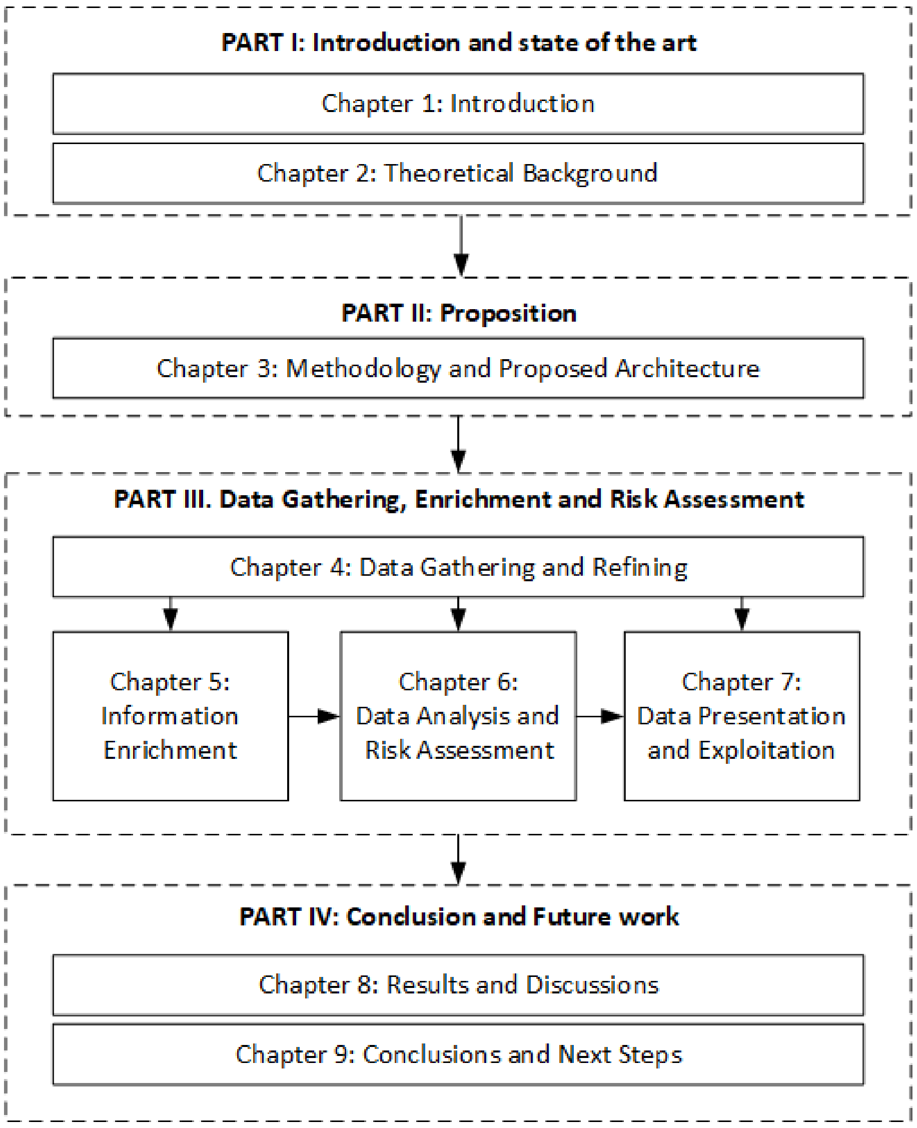

PART I: Introduction and state of the art
Chapter 1: Introduction
Chapter 2: Theoretical Background
PART II: Proposition
Chapter 3: Methodology and Proposed Architecture
PART III. Data Gathering, Enrichment and Risk Assessment
Chapter 4: Data Gathering and Refining
Chapter 5: Information Enrichment
Chapter 6: Data Analysis and Risk Assessment
Chapter 7: Data Presentation and Exploitation
PART IV: Conclusion and Future work
Chapter 8: Results and Discussions
Chapter 9: Conclusions and Next Steps

Figure 1.1: Structure of the Thesis. The thesis is structured into four parts and nine chapters. The first two parts provide a brief introduction to the thesis objectives and goals, followed by a review of the state of the art and the methodologies employed throughout the subsequent sections. The third part outlines the various phases of the data pipeline, from data collection to presentation. Lastly, the final part summarizes the key results and conclusions.

the defined objectives. Furthermore, we provide a detailed description of the datasets used in subsequent chapters and outline the proposed architecture. This architecture encompasses various Information Technology (IT) components and explains the relationships between them.

- Chapter 4: Data Gathering and Refining
  This chapter addresses Research Question RQ1 by detailing the

dataset used in this study. It outlines the processes of gathering company reports and social media information. From these sources, the following steps were performed: extracting text, cleaning the data, and inserting it into a structured database. Additionally, it discusses the challenges encountered during these stages and describes the Deep Learning (DL) components developed to anonymize the data, ensuring its suitability for public release in compliance with GDPR regulations.

- Chapter 5: Information Enrichment
  In this chapter, Research Questions RQ2 and RQ3 are examined by discussing the most suitable models for extracting structured information from unstructured data. The chapter explores how to integrate this diverse information, identifying entities based on their attributes and context. Additionally, the chapter encompasses other engineering activities undertaken, which may not directly pertain to a specific research question or contribute novel insights to the research community. These IT components leverage current technologies to enhance data enrichment and provide additional information utilized in subsequent models, such as text embeddings generation for contextual search and geocoding of addresses.

- Chapter 6: Data Analysis and Risk Assessment
  This chapter delves into exploiting information to identify risky clusters from various perspectives based on non-financial data. Research Questions RQ4, RQ5, RQ6, and RQ7 are explored here. Due to the nature of our dataset, characterized by its considerable length, we require models capable of making predictions utilizing as much information as possible. RQ4 investigates different models and architectures to develop lightweight and high-performing BERT-based models to be used later. Q5 explores the potential of identifying hidden authors of reports and clustering companies based on this information. RQ6 investigates the potential of using the annexes of annual reports alone to predict the risk of bankruptcy, without relying only on numerical data. Lastly, RQ7 examines the possibility of assessing a company's risk based on multiple and multidimensional risk factors.

- Chapter 7: Data Presentation and Exploitation
  In this chapter, we introduce a web interface designed to enhance

user experience. This is intended for end users who seek information and results derived from our models.

- Chapter 8: Results and Discussions
  This chapter presents the results of the main components, followed by an analysis in relation to the research questions and thesis objectives.

- Chapter 9: Conclusions and Next Steps
  This chapter outlines the primary contributions of this doctoral thesis and proposes next steps for enhancing the current solution. Additionally, it identifies open research questions that have emerged from this work.

1.6 Published Papers

- **Paper I: Auto-clustering of Financial Reports Based on Formatting Style and Author's Fingerprint.**
  Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022). MIning DAta for financial applicationS (MIDAS). Grenoble, France, September 1923, 2022.

  Abstract: This paper presents an end-to-end administrative document analysis system. This system uses case-based reasoning in order to process documents from known and unknown classes. For each document, the system retrieves the nearest processing experience in order to analyze and interpret the current document. When a complete analysis is done, this document needs to be added to the document database. This requires an incremental learning process in order to take into account every new information, without losing the previous learnt ones. For this purpose, we proposed an improved version of an already existing neural network called Incremental Growing Neural Gas. Applied on documents learning and classification, this neural network reaches a recognition rate of 97.63%.

- **Paper II: Reducing tokenizer's tokens per word ratio in Financial domain with T-MuFin BERT Tokenizer.**
  32nd International Joint Conference on Artificial Intelligence (IJCAI 2023). Financial Technology and Natural Language Processing (FinNLP). 19th-25th August 2023. Macao, S.A.R

  Abstract: Most domain-specific BERT models are designed to work with short sentences and do not deal with the limitation of 512 tokens in the default BERT tokenizer. This limitation is further exacerbated if the tokenizer has high number of tokens per word ratio (fertility) and thus splits words into several tokens. A term based multilingual Financial (T-MuFin) BERT tokenizer has been proposed to reduce the fertility of the default BERT tokenizer by extending the base dictionary with the most common financial terms instead of word pieces. One key factor of this proposal is to introduce multiword domain-specific terms without affecting the performance of the BERT models. T-MuFin BERT tokenizer reduces at least 40% of the fertility of long text sequences. T-MuFin BERT improves the fine-tuning of a downstream task by approximately 4% compared to a default fine-tuned model. Hence, by reducing the tokenizers fertility, the results of explainable methods are more user-friendly.

- **Paper III: A Novel Architecture for Long-Text Predictions Using BERT-Based Models.**
  10th Intelligent Systems and Applications (IntelliSys 2024). 5th-6th August 2024. Amsterdam, The Netherlands.

  Abstract: The rise in Natural Language Processing usage is driven by the effectiveness of transformer-based models in understanding contextual information. While generative models like GPT have garnered attention, BERT-based models have not received as much focus. Despite their potency, GPT-based models face classification challenges mainly due to hallucinations. Alternatively, BERT-based models prove proficiency in classification, but are restricted to less than 512 words as input. In this context, we propose a BERT-based architecture for long-text predictions, featuring an integration network designed to address the challenge of processing lengthy textual inputs. Within this architecture, an LSTM integration layer demonstrated superior performance for models processing only a few pages, while concatenation layers atop BERT yielded improved results for handling both short and very long text sequences. Two tests were conducted to evaluate the effectiveness of this architecture using company documents sourced from the Luxembourg Business Registers. The first test focused on predicting bankruptcy risk based on extensive text sequences extracted from annexes in Annual Accounts. Our approach exhibited strong predictive abilities in this task, boosting accuracy from 5% to 80% while maintaining a similar precision of 73%. In the second test, which involved analyzing modification reports, the model excelled at predicting page types based on the entire content, achieving a significant increase of approximately 23% in F1 Score compared to the BERT base model. In summary, the proposed BERT-based architecture offers a versatile solution for handling long text sequences in various domains, providing valuable support for categorization and prediction tasks across different industries and applications.

- **Paper IV: Transforming Unstructured Sensitive Information into Structured Knowledge.**
  5th ACM International Conference on AI in Finance (ICAIF ’24). 14th-17th November 2024. Brooklyn, NY, USA.

  Abstract: Information is crucial in today’s context, yet less than 20% of companies utilize their unstructured data due to its complexity. Information Extraction (IE) is vital for effective data use, but current IE models face four major issues. First, they often provide limited information, such as a simple entity-attribute relation. Second, they struggle with multiple languages. Models like GPT, Mistral, and Llama3 show promise but face a third issue: output reliability due to hallucinations. Fourth, there is a challenge in reducing sensitive data leakage after fine-tuning models. This study introduces an enhanced approach for fine-tuning GPT-based models, designed to extract and assess information involving multiple entities and attributes, performing both multientity extraction (MEE) and multirelation extraction (MRE), and presenting results in a JSON format. Our methodology evaluates the impact of using synthetic data for fine-tuning to ensure reliable outcomes. Applied to legal documents from the Luxembourg Business Registers (LBR), our findings show that replacing sensitive data with synthetic data significantly improves the fine-tuning of Llama3-based models, though not for Mistral-based models. Our top models outperform Mistral in various scenarios, requiring only 500 samples for fine-tuning and running efficiently on modest servers. This approach is suitable for multilingual Information Extraction in any domain.

Chapter 2

Theoretical Background

Contents

2.1 Introduction 14
2.2 Company Risk Prediction 15
2.3 Natural Language Processing 17
2.4 Deep Learning Models (DNN) 19
2.5 Attention Mechanism and Transformers 19
2.6 Encoder-based models: BERT 21
2.7 Decoder-based models: GPT 23
2.8 Tokenization and Embeddings 26
2.9 Information Extraction 29
2.10 Document Clustering 32
2.11 Entity Resolution 33
2.12 Graph Analysis 35
2.13 Summary 37

2.1 Introduction

This chapter explains the key concepts and the current state of the art related to them. Each concept is applied in one or more chapters of this work. The primary focus of the project is to use various types of data sources to predict a company's risk through Machine Learning models, with a particular emphasis on Deep Learning models.

2.2 Company Risk Prediction

As indicated by The European Central Bank [Bank, 2013], SMEs are a keystone of the euro area economy. Their importance stems from their capacity to innovate and create employment [Crovini et al., 2021]. At the time of the European Bank's report, it highlighted the difficulties SMEs face in obtaining bank loans and their frequent obligation to pay higher financial costs.

As also mentioned by Crovini et al. [2021], empirical evidence shows that approximately 50% of SMEs close down around the fifth year of their establishment [ISO-31000, 2015], primarily due to low business sophistication and inadequate management capacities. This underscores the constant efforts by public and private organizations to help SMEs reduce their risks by promoting better practices within companies and encouraging financial institutions to improve their risk analysis. These initiatives aim to reduce losses caused by unreliable companies and provide cheaper loans to trustworthy ones.

Several studies have been conducted to evaluate the risk of SMEs and companies in general, using various methods, technologies, and data sources. As detailed by Bhatore et al. [2020], the main research directions in risk assessment for companies are associated with credit scoring, fraud detection, and Non-performing asset prediction (NPA). NPA can be conceptualized as a loan that is not generating income for the financial institution. Most of the papers analyzed by the authors focused primarily on credit scoring, followed by fraud detection.

The methods employed in these studies range from rule-based mechanisms to the use of Artificial Intelligence (AI). Bhatore et al. [2020] provides a detailed explanation of the different AI approaches used for companies' risk assessment. Some of these approaches include knowledge representation through ontologies, fuzzy logic, and various Machine Learning techniques such as Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Hidden Markov Models (HMM), Supporting Vector Machines (SVM), Genetic Algorithms, Clustering, Bayesian Methods, and Deep Learning models. The authors also found that the most commonly used Machine Learning technique is Neural Networks (DL), followed by hybrid models.

Studies such as Qu et al. [2019] and Clement et al. [2020] analyzed numerous papers in the financial domain and bankruptcy prediction, respectively, and found that the primary data sources used for predictive models were numerical. This numerical information includes financial statements and financial ratios, which are the main sources of information for traditional manual-based risk analysis. In these studies, the number of researchers using text as input data was relatively small. Most of these textual sources referred to short texts, such as company or management-related tweets and news. The main limitation of the researches for taking advantage of other datasources such as long textual information, images or videos were because the limited capacity of the technology.

For example, in financial analysis and risk assessment, Musanovic and Halilbegovic [2021], Geng et al. [2015], Olson et al. [2012], and Chen et al. [2011] utilize numerical information for tasks such as classifying solvent and insolvent firms, stock prediction, and credit scoring. Clement et al. [2020] reviewed 32 research papers published between 2016 and 2020 on bankruptcy prediction, finding that all but one (Mai et al. [2019]) used financial ratios or company indicators as input data. Additionally, Wang et al. [2023] proposed an automatic feature engineering approach for bankruptcy prediction, which calculates various financial ratios from the financial statements of Luxembourgish companies (the same dataset used in this study). Their best-performing model achieved an AUC of 87.13% using a Light Gradient Boosting Machine (LightGBM) model.

In recent years, technologies for analyzing audio, images, videos, and especially text have improved significantly, allowing researchers to utilize this data in various domains, such as finance. Text processing is a component of NLP, a field within AI that focuses on understanding human language in its natural form. According to Fisher et al. [2016], the application of NLP models has increased since 2010 with the advent of Machine Learning (ML) and AI, leading to a decline in earlier manual text analysis studies.

Studies focusing on financial models that utilize text often employ short texts to train their models. For instance, Sawhney et al. [2021] ranks profitable stocks using news and tweets. They analyze tweets to gauge sentiment and investor attention regarding Bitcoin market dynamics. Gao et al. [2018] leverages loan applications to predict credit risk and determine whether a loan will be approved.

Some studies that utilize text for bankruptcy prediction were conducted by Mai et al. [2019]. The authors use financial disclosures as input data. The scope of the study is limited to the use of Annual Filings from public U.S. corporations. The U.S. Securities and Exchange Commission (SEC) requires the inclusion of a risk description in these filings. In the present work, textual information is also used to predict the risk of bankruptcy, drawn from the financial annexes of the Financial Statements. These documents can be 10 pages or more in length, and the risk information is not necessarily explicitly stated. With this part of our research we aim to determine if it is possible for a NLP model to detect some kind of risk based only on the textual information.

2.3 Natural Language Processing

NLP is a field of AI that focuses on the interaction between computers and humans using natural language. The primary goals of NLP are the comprehension, interpretation, and generation of human language in a manner that appears natural. Some of the tasks in NLP are: Sentiment Analysis, Machine Translation, Name Entity Recognition, Text Classification, Text Summarization, Speech Recognition, Question/Answering and so on. These tasks also can be grouped in a upper level such as categorization models and generative models.

- Sentiment Analysis (SA): This task involves assigning a category to the provided text, typically related to sentiments such as joy, anger, positive, negative, or neutral. Alternatively, categories may include distinctions such as formal, informal, or neutral. This is an example of a Categorization model.
- Machine Translation: This task involves two steps: understanding an input text in one language and generating an output text in another language. This corresponds to a Generative model.
- Name Entity Recognition (NER): In this task, parts of the input text, such as words, are classified into key entities like People, Company, Date, or Location. This is an example of a Categorization model.
- Text Classification: This task is similar to Sentiment Analysis but on a broader scale. Classification can be done into specific, well-

defined categories or through automatic clustering based on shared features identified by the model. It can also be extended to document classification or clustering. This corresponds to a Categorization model.

- Text Summarization: Similar to machine translation, but the output text is a concise summary of the input text. This corresponds to a Generative model.
- Speech Recognition: In this task, spoken language is converted into written language. This corresponds to a Generative model.
- Question-Answering (QA): This task involves providing an answer to a question based on the input text. It can be categorized as either a Categorization or Generative model. If it is a Categorization model, the answer will indicate the location in the text where the answer can be found. If it is a Generative model, the answer may be expressed in different words than those present in the input text.

The work of Kang et al. [2020] provides a detailed account of the evolution of NLP, with a particular focus on DL. Computational models, including AI and particularly ML, require numerical inputs, which simplifies the process for numerical models. However, when dealing with text, several challenges arise, such as converting text into meaningful numerical representations. Early approaches to handling textual data involved encoding each word as a specific number and position within a vector, processing these corresponding numbers using techniques like one-hot encoding. These initial models faced significant issues, including high dimensionality and complexity, as well as the inability to account for the varying meanings of the same word depending on context.

To address the challenges of high dimensionality and sparsity, text embeddings became a crucial development. Techniques such as Word2Vec (Mikolov et al. [2013]), GloVe (Pennington et al. [2014]), and ELMo Peters et al. [2018]) were introduced. These approaches represent a single word or part of a word as a multidimensional vector (text embeddings), capturing diverse features with varying levels of activation. This method allows models to retain more meaning and significance, enabling the comparison of words based on their semantic similarity.

2.4 Deep Learning Models (DNN)

After establishing how text can be converted into numerical form using text embeddings, the next challenge was the design of the model itself. As previously discussed, Bhatore et al. [2020] highlight the impact of DL models on the current state of the art in the financial domain. The term *Deep* refers to the complexity of the model, indicating the presence of multiple intermediate layers between the input and output layers. This depth allows the model to learn complex patterns. Initially, classical networks were designed for classification tasks, where the final layer consisted of a fixed number of units representing the desired categories.

Unlike classical Deep Neural Networks (DNN) models, which have fixed input and output sequences, text-based models must handle variable-length sequences of text to generate other variable-length sequences. The first models developed to address this challenge were Recurrent Neural Networks (RNN). These networks are composed of sequences of neurons capable of managing an undetermined number of input tokens. Despite their ability to handle long text sequences, RNN often struggle with retaining previous words or context, a limitation known as the vanishing gradient problem. This issue arises when the derivatives of long sequences diminish, impairing the backpropagation process and causing the model to lose track of earlier information.

As a result, Long Short-Term Memory (LSTM) networks were developed by Hochreiter and Schmidhuber [1997] to address the vanishing gradient problem. These networks are designed to retain some historical context while selectively forgetting information that is no longer deemed necessary by the model. Subsequent advancements, such as the bidirectional implementation (Bi-LSTM) and the Gated Recurrent Unit (GRU) [Chung et al., 2014], were introduced to further enhance LSTM performance. However, LSTM models still struggle to effectively connect distant concepts in lengthy text sequences.

2.5 Attention Mechanism and Transformers

The significant leap in technological development was achieved by Vaswani et al. [2017] with their *Transformer-model architecture* and its multi-head attention or *attention mechanism*. This architecture is composed of two main components as is shown in Figure 2.3. The left part is the Encoder and the left one, before the final Linear and Softmax layers is the Decoder.

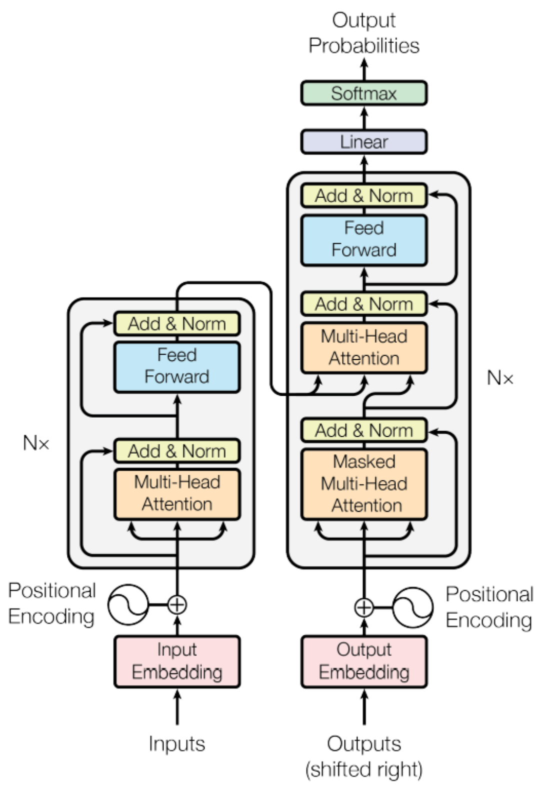


Figure 2.1: Transformer-model architecture. Image taken from: Vaswani et al. [2017]

The embeddings are fed into the network simultaneously, and through multi-head attention, each text input is processed in parallel. This mechanism emphasizes the most critical words necessary for understanding the context of the entire input. Each word can attend to the most relevant words, regardless of their position in the text.

In addition to multi-head attention, the architecture includes an intermediate layer for addition and normalization. This layer combines the intermediate results with the inputs, followed by a feed-forward network that captures complex relationships and transformations. These components are repeated to construct a sophisticated deep learning network.

The Encoder stack is responsible for learning text representations and capturing contextual information. The Decoder stack generates the output sequence based on the learned sequences and previously generated tokens. This iterative process continues to produce words until a termination sequence is reached.

The original model was implemented with six encoder layers and six decoder layers. It was primarily designed for machine translation, with experiments conducted on English-to-German and English-to-French translation tasks. For converting words into numerical representations, the model used Byte Pair Encoding (BPE) for tokenization. Details about the tokenization concepts and processes can be found in section 2.8.

2.6 Encoder-based models: BERT

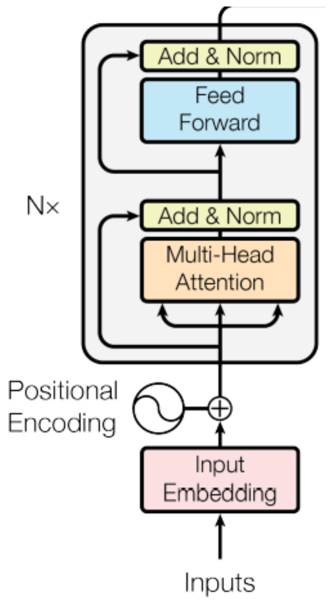


Figure 2.2: The Transformer-model architecture: Encoder. Image taken from: Vaswani et al. [2017]

The first part of the Transformer model is the Decoder, and as explained before is in charge of the language understanding and text representation. For this reason, Devlin et al. [2018] presented their BERT model. This model has the first embedding layer and has 12 encoder layers for the small version (BERT-base) and 24 encoder layers for the large version (BERT-large).

BERT was conceived as a pre-trained model, meaning that the authors trained it on a large dataset for general language understanding tasks and then made the weights of the encoder layers available for others to use. Training models like BERT-base or BERT-large is resource-intensive, but once the model has learned about the language, these pre-trained weights can be fine-tuned on specific tasks with significantly less computational effort and fewer training samples. This fine-tuning process allows the model to adapt to specialized tasks quickly and efficiently. All the models derived from BERT-base or BERT-large are BERT-based models. The main characteristic of BERT-based models are that are focused on classification tasks mainly.

This pre-training approach significantly accelerated the progress of NLP by enabling the rapid development of specialized models with reduced training resources. As of 2024, Hugging Face lists over 1,400 BERT-large-based models and more than 10,800 BERT-Base-based models, covering various domains, languages, and downstream tasks such as text classification, named entity recognition, question answering, and more.

Moreover, BERT-based models have two main limitations: the maximum input size and the language(s) in which the model was pre-trained. The maximum input size for BERT-based models is 512 tokens, which does not necessarily equate to 512 words. A token can represent a full word, a part of a word (word-piece), or a special token used to control the model's behavior. For most NLP tasks, at least two special tokens need to be added to the input: *CLS*, which marks the beginning of the input, and *SEP*, which typically denotes the end of the input text. Additionally, a non-mandatory token, *PAD*, can be used to standardize input size to a fixed maximum length during fine-tuning or batch prediction. Details about the tokenization are explained in section 2.8.

Some notable examples of domain-adapted BERT models include FinBERT for finance [Araci, 2019], BioBERT for biomedicine [Lee et al., 2019], LegalBERT for the legal domain [Chalkidis et al., 2020], and SciBERT for scientific literature [Beltagy et al., 2019]. These models generally avoid BERT's input size limitation because they either work with smaller datasets or truncate the input data to fit within the model's 512-token constraint such as BERTSUM [Liu, 2019] or Kim and Yoon [2021].

In cases where truncating input data is not feasible, and more extensive data needs to be processed by BERT-based models, researchers have explored integrating additional models to complement BERT. A different strategy was proposed by Grail et al. [2021] in their multi-layer transformer architecture. They divide the input into K blocks of tokens, consolidate each transformer's output through a GRU network, and then pass it through a feed-forward network or dense layer. This method effectively manages long sequences of text without adding significant complexity to the network or requiring retraining of the BERT model. However, it still faces challenges with extremely long document sequences.

Similarly to the previous approach, BERTSUM [Liu, 2019] also proposed two other network configurations. The proposed models attach a final LSTM layer or an additional Transformer network on top of BERT instances. This configuration we use as our base model for our own proposed model in subsection 5.4.1. Another example is BERT-AL (BERT for Arbitrarily Long Document Understanding) [Zhang et al., 2020]. In this approach, the authors introduce an LSTM network after each attention layer of BERT to handle long text sequences, similar to BERTSUM. However, to mitigate the vanishing gradient problem often associated with LSTM networks, BERT-AL differs from BERTSUM by providing the full representation of each layer using a multichannel LSTM, rather than relying solely on the *CLS* token to integrate BERT models. While this method proves effective in managing long sequences, it also increases the model's complexity and demands more resources for training.

Another approach is DocBERT [Adhikari et al., 2019], which, despite its name, is not a traditional BERT model. The authors fine-tuned BERT weights and then used knowledge distillation to transfer the learned weights to an LSTM network, creating a lighter model. However, this approach can struggle with very long sequences due to the inherent limitations of LSTM layers.

2.7 Decoder-based models: GPT

Generative models, such as OpenAI's GPT series [Radford, 2018], are derived from the Decoder component of the Transformer architecture. Although the initial versions of these models did not significantly impact the evolution of generative models, their development laid the groundwork for future advancements. As noted by Forbes[1], when OpenAI released an early demo of ChatGPT on November 30, 2022, it quickly went viral on social media due to its impressive ability to understand questions and generate human-like responses. Within just a few days,

[1]https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/

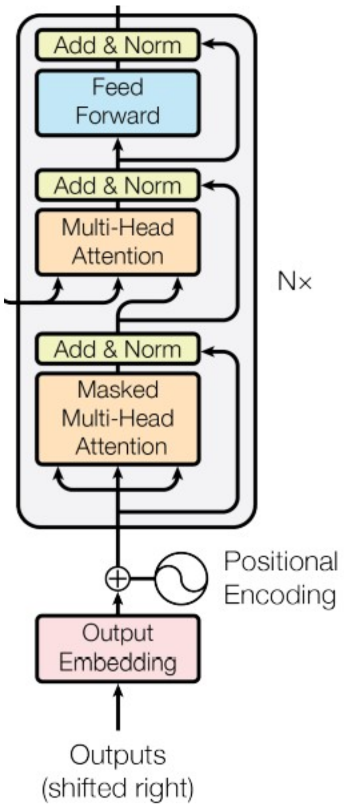


Figure 2.3: The Transformer-model architecture: Decoder. Image taken from: Vaswani et al. [2017]

millions of users began interacting with the GPT model, leveraging it as a tool to assist with daily tasks in both personal and professional settings.

In June 2018, the first version of GPT was released, featuring 117 million parameters. This model was trained on a dataset of books using a Next Sentence Prediction task. In February 2019, GPT-2 was introduced, boasting 1.5 billion parameters and significant improvements, though it was not yet ready for public release. GPT-3 followed with 175 billion parameters, marking a substantial evolution in the model's capabilities. Subsequent versions, including GPT-4, introduced progressive enhancements, further advancing the state of generative AI.

While these models are available for free through the ChatGPT[2] web interface with limited features, users can opt for a monthly subscription to access the full range of capabilities. Additionally, it is possible to integrate the models with company data via an Application Programming Interface (API), enabling seamless integration into business operations.

[2]https://chatgpt.com/

Microsoft Copilot also released a similar GPT version in Bing Chat[3]. Google released Gemini or Bard[4]. Anthropic released Claude AI[5]. Mistral AI released Mistral[6]. All these companies have a chat web application and provide access through their own API.

The main characteristic of these models is that they are accessed through the internet. Consequently, the information has to be sent to external servers in unknown locations, which could risk the privacy and confidentiality of the data during transmission or server processing. Among all the previously mentioned models, Mistral's weights and models can be downloaded to be deployed locally and also fine-tuned for specific needs.

In March 2021 EleutherAI released GPT-Neo[7], representing one of the first models to be an open-source alternative to GPT. This model was able to be downloaded and fine-tuned. This was considered as the first step to democratize the generative AI. During the following months many propietary and open source versions were released. Is important to mention that in February 2023 Meta (formerly Facebook) released LLaMA (Large Language Model Meta AI) [Touvron et al., 2023].

LLaMA models have been available in various sizes, with one of the lighter versions designed to be fine-tuned using minimal computational resources. The initial release included models with 7B, 13B, 30B, and 65B parameters. In July 2023, LLaMA 2 was introduced, enhancing conversational capabilities and offering models in three sizes: 7B, 13B, and 70B. The 70B model competed closely with similar proprietary models.

In February 2024, LLaMA 3 was released with updated versions in 7B, 13B, 30B, and 70B sizes. In August 2024, LLaMA 3.1 was introduced with significant improvements. While it retains the same size options as LLaMA 3, LLaMA 3.1 offers enhanced performance, rivaling or surpassing the latest versions of Claude and Mistral, and competing closely with GPT-3 and GPT-4. LLaMA 3.1 is available in 8B and 70B sizes and was trained on 15 trillion tokens. The latest version, LLaMA 3.2, was released in September 2024. While we successfully tested its out-of-the-box performance, which exceeded that of its predecessor, fine-tuning results were less promising. This may be due to

[3]https://www.bing.com/chat
[4]https://gemini.google.com/
[5]https://claude.ai/
[6]https://chat.mistral.ai/chat
[7]https://github.com/EleutherAI/gpt-neo

the need for updates in prompting or fine-tuning parameters to achieve optimal results.

The ability to use models like LLaMA and Mistral (along with other open-source GPT models) within companies' internal network significantly reduces the risk of data leakage. Additionally, these models can be fine-tuned to meet specific needs with relatively low computational resources, making them both secure and cost-effective for businesses.

Like BERT models, which are typically limited to 512 tokens, GPT models have a limitation known as context size. This refers to the maximum number of tokens that can be processed in a single sequence, including both input tokens and generated output tokens. The context size varies by model as shown in Table 2.1.

Table 2.1: Number of maximum tokens per GPT-model

Model	**Tokens**
GPT-2	1,024
GPT-3	2,049
GPT-3.5	4,096
GPT-4	8,192 & 32,768
LLaMA 1	2,049
LLaMA 2	4,096
LLaMA 3 & 3.1	8,192

One of the primary challenges with generative models is the issue of hallucination. Hallucination refers to the generation of text that sounds plausible but is factually incorrect or lacks grounding [Alkaissi H, 2023, Sovrano et al., 2023]. Despite this issue, the versatility of these models allows them to be used effectively in various applications, including question-answering, summarization, and conversational agents.

In this work, we explore two use cases of Generative Models. First, we apply them to Information Extraction by fine-tuning a LLaMA 3.1 model and comparing its performance with the best and most widely used GPT models. Second, we leverage the generated contextualized embeddings to create an embedding database for information retrieval.

2.8 Tokenization and Embeddings

Tokenization is the process of breaking down text into smaller units, known as tokens, that can be associated with numerical representations.

These tokens may represent characters, subwords, symbols, or complete words. The tool or method used to perform tokenization is called a tokenizer. The tokenizer determines how to split the text into tokens based on a dictionary or list of known words, subwords, characters, or symbols.

To function effectively, a tokenizer first needs to create its own dictionary. Once this dictionary is established, the tokenizer can convert text into tokens according to its predefined rules. For example, if a tokenizer's dictionary includes the English word *table*, then the word *table* will be tokenized into a single token. However, if the tokenizer is used to tokenize the French word *tableau*, which is not in the dictionary, it may split it into two tokens: *table* and *##au* (or use an *UNK* token to represent unknown or out-of-vocabulary words).

There are several methods for generating a tokenizer's dictionary, each with its advantages. The most commonly used approaches include:

- Word tokenization: : In this method, the dictionary is built from the most frequent words in a dataset. The size of the dictionary is capped at a specified limit N, and the dictionary includes the top N words. Words that fall outside this frequency threshold are replaced with a single *UNK* (unknown) token. This approach is employed in Mikolov et al. [2013], Pennington et al. [2014], and Peters et al. [2018], among others.

- Sub word tokenization: This technique involves breaking words into smaller units, which helps manage out-of-vocabulary words and reduces the dictionary size. One prevalent method is BPE, which iteratively merges the most frequent pairs of characters or subwords. BPE is utilized in Vaswani et al. [2017], and variations of BPE are employed in GPT-2 and GPT-3. Another common method is WordPiece, which starts with a corpus of single characters and iteratively creates new subwords by combining existing tokens. The most frequent subwords are selected until a predefined dictionary size is reached. WordPiece is used in Devlin et al. [2018]. Other subword-based methods may generate dictionaries based on subword probabilities.

- Character tokenization: This method uses individual characters, including letters, digits, punctuation marks, and special symbols, as the base for tokenization. Character tokenization is highly flexible and can adapt well to diverse datasets. For instance, Peters et al. [2018] uses character-level tokenization to enhance word tokenization with additional morphological details.

As previously mentioned, the tokenizer's dictionary is influenced by the dataset on which it is trained. Consequently, different BERT models are trained on various datasets and languages. It is also important to note that the tokenizer's behavior regarding case sensitivity varies; some models treat words with different cases as distinct, while others do not. This distinction is why BERT models are available in both cased and uncased versions. In this work, we use the multilingual BERT uncased model, as it is suitable for documents in English, French, and German, all of which are supported by this tokenizer[8].

Depending on the language, the case, the dataset and the tokenization method a word can generate one or more tokens using different tokenizers. The measure of the capacity of the tokenizer to generate certain number per tokens in average is called *tokenizer's fertility*. The concept of *tokenizer fertility* is defined by Rust et al. [2020] as the average number of subwords generated per tokenized word. A fertility of one means that, on average, each word generates one token. A higher fertility value indicates that each word produces more tokens. Typically, monolingual tokenizers have lower fertility compared to multilingual ones. Higher fertility can lead to a reduction in the total amount of information that can be effectively processed by deep learning models.

As previously discussed, embeddings are numerical representations of text. Each embedding vector corresponds to a token, and its dimensionality is determined by the model. BERT has a embedding vector size of 768. The position of a token in the vocabulary dictates the index of its vector in the embedding layer. Choosing the appropriate tokenizer for the target language is crucial. For example, as illustrated earlier in this section, tokenizing the word *tableau* with an English tokenizer may yield significantly different interpretations than those intended.

Some authors such as Kim and Yoon [2021], adapt the existing dictionary to add domain-specific new terms and then fine-tune the new embeddings. The fine-tuning of the new weights with or without adding new terms to the dictionary should be done with a Masked Language

[8]`google-bert/bert-base-multilingual-uncased`

Model (MLM) or Next Sentence prediction (NSP). These fine-tuning tasks forces the model to calculate o infer a token or groups of tokens based on the contexts.

In the present project we develop our own financial tokenizer T-MuFin BERT Tokenizer in section 6.3. Our tokenizer goes behond the one word-one tokenization and creates multiword tokenization, adding the most frequent multi-words financial terms to the tokenizer's dictionary. As a result the tokenizer's fertility of a multilingual BERT tokenizer for the financial domain is reduced.

Just as an embedding can represent a word or a part of a word, it can also represent an entire paragraph, page, or even a full document. These embeddings numerically capture the context of the input text, with the specific position of the chosen vector within the *dl* model reflecting this context. These embeddings can be utilized as data sources for performing context-based searches. To compare high-dimensional vectors, various distance metrics are available, including Cosine Similarity, Inner Product, and Euclidean Distance:

- Cosine Similarity: Measures the cosine of the angle between two vectors, focusing on their direction rather than magnitude. Because of this, the vectors should be normalized. A cosine similarity of 1 indicates that the vectors are identical in direction, while -1 indicates they are opposite.
- Euclidean Distance: Measures the straight-line distance between two vectors in the vector space. It considers the overall dimensions, with smaller distances indicating greater similarity between the vectors.
- Inner Product: Calculates the sum of the products of the corresponding elements in the vectors. This metric reflects the degree to which the vectors point in the same direction, with a higher inner product indicating greater similarity.

2.9 Information Extraction

Given the vast amount of information present in free-form pages or semi-structured formats within financial documents, it is crucial to have tools that can extract this unstructured data and organize it into a structured format. Various authors have proposed different classifications of methods used for Information Extraction (IE) [Zaman et al., 2020, Adnan and Akbar, 2019a,c, Adnan et al., 2019, Adnan and Akbar, 2019b]. These methods include rule-based approaches, Maximum Entropy models, chronological labeling, template filling, sequential labeling, and learning-based techniques.

Aside from learning-based techniques, Zaman et al. [2020] highlights several limitations of other methods, such as domain specificity, which hinders their ability to generalize to new formats or datasets. Additionally, these methods often suffer from language specificity, typically being tailored to monolingual datasets and struggling with synonyms and ambiguities.

Other techniques and tools have been proposed for Information Extraction using triplets. One such tool is the Open Information Extraction (OpenIE) annotator Angeli et al. [2015], StanfordNLP [2020], developed by the Stanford Natural Language Processing Group, which uses a Logistic Regression classifier to extract information in the form of triplets. Another approach, DragonIE, proposed by Yu et al. [2022], leverages directed acyclic graphs to outperform OpenIE both within and outside the training domain. Additionally, Qu et al. [2022] introduced a method for extracting triplets from entities using a combination of BiLSTM networks and Graph Convolutional Networks (GCN), alongside a parallel BiLSTM network and Convolutional Neural Network (CNN), integrated with a softmax classifier. Examples of triplets provided in their study include *born_in(John, County Kerry)* and *educated_in(John, Saint Columbas College)*. The research indicates that triplets are typically limited to linking a certain number of pairs from a small input text (50 words/tokens).

Others authors such as Toprak Ahmet [2023] uses Named Entity Recognition (NER). This task involves identifying and labeling the most important entities within a text. For example, in the sentence *Apple Inc. announced its latest iPhone 15 on September 12, 2024, in Cupertino, California*, the tokenizer first breaks the text into individual tokens. Then, a model (such as BERT) recognizes specific entities and assigns a label to each token according to its entity type. In this example, the token-entity pairs are as follows: “Apple” is labeled as B-ORG, “Inc.” as I-ORG, “iPhone” as B-PRODUCT, “15” as I-PRODUCT, “September” as B-DATE, “12” as I-DATE, “2024” as I-DATE, “Cupertino” as B-

LOC, and “California” as I-LOC. Tokens that do not correspond to any entity are labeled as “O” (Outside). The prefixes “B” and “I” denote the beginning and continuation of an entity, respectively, within a multi-token entity. Having ORG as Organization and LOC as Location.

One example of using regular expressions and deep lerning models combined is Exsense [Guo et al., 2021]. It extracts sensitive information from unstructured data using a combination of context- and content-based mechanisms. It employs regular expressions (e.g., emails, addresses, securities) and a BERT-biLSTM-attention network to categorize sensitive information into personal, networking, secrets, and credential information. However, Exsense is limited to English and is highly sensitive to OCR errors.

On the other hand, Information Extraction using generative models like GPT can produce output in the desired format. However, a significant challenge with these models is the risk of generating hallucinationsfabricated or inaccurate information. For this work, we require a generative extraction model that reliably returns output in JSON format, minimizing the risk of such errors.

Training GPT-based Large Language Model (LLM) requires significant processing power and substantial GPU memory. For instance, the LLaMA3.1 8B model, which has 8 billion parameters and takes up 5.5 GB of disk space, needs about 22 GB of GPU RAM for fine-tuning due to the additional memory required for gradients and optimizer states. Parameter-Efficient Fine-Tuning (PEFT) with Quantized Low-Rank-Adapters (QLORA) helps reduce this memory burden. PEFT, by employing techniques such as Low-Rank Adapters (LoRA), compresses gradients and activations into smaller matrices during fine-tuning. QLORA further minimizes memory use by applying 16-bit, 8-bit, or even 4-bit quantization instead of the standard 32-bit precision, making it possible to fine-tune a 5.5 GB model on a GPU with just 16 GB of RAM.

In summary, many existing tools are constrained by factors such as language, structure, domain specificity, or triplet-based relations. Models that address these limitations are often either costly or too large to be efficiently deployed on smaller servers. section 5.5 describes the proposed fine-tuning methodology to create a GPT-based Information Extraction which allows to get sensitive information in JSON format.

2.10 Document Clustering

An important part of the available data consists of Annual Accounts, which include Financial Statements such as the Balance Sheet and Profit and Loss Statement, as well as Legal or Financial Annexes. While the Financial Statements follow a standard template, the Legal Annexes are presented as free-text pages and do not adhere to a minimum information or page requirement. Given that most accounting consultancy firms, audit firms, and accountants likely use similar templates for Annual Accounts for SMEs clients, with only specific modifications for individual companies, and considering that corporations often share a single template among their subsidiaries and affiliated companies, having this information readily available can be highly valuable. For this reason, it is important to have a service that processes Annual Accounts and clusters them based on their content.

Some existing approaches to document clustering include k-means, k-medoids, Latent Dirichlet Allocation (LDA) Blei et al. [2003], and Document-Term Matrix (DTM) [Xu et al.]. These methods typically require specifying the number of clusters in advance, which is not suitable for the proposed use case. On the other hand, DBSCAN Ester et al. [1996] is a widely used clustering algorithm that does not require predefining the number of clusters. However, DBSCAN relies on a finite and limited number of features for clustering, whereas the proposed use case involves an essentially infinite number of features derived from both content and format. This makes DBSCAN an impractical choice for this scenario.

In Xu et al., the authors proposed using Non-Negative Matrix Factorization (NMF) on the document-term matrix for clustering documents. This method involves selecting one of the predefined topics, which directs the clustering process towards topic-based clustering. In contrast, Hamza et al. employs a graph-matching approach for clustering administrative documents. This technique creates clusters based on the required number and does not rely on specific topics. The features utilized for clustering include the content and the relative position of text on each page of the document. The distance metric used in this approach is termed "graph-probing distance" (Equation 2.1), which involves incremental learning and compares the input document with all existing data within each cluster. Other commonly used distance metrics in clustering include Jaccard, Recall, Precision, Accuracy, Huberts statistics, and the Rand Index (RI) [Rand, 1971].

$$Pb = \sum |freq(N_{G1}) - freq(N_{G2})| \tag{2.1}$$

RI measures the correct assignment of an element into a cluster as shown in Equation 2.2, having a as the number of pairs of elements belonging to the same cluster, and b as the number of pairs of elements belonging to different clusters, according to the training dataset (T) and labeled dataset (L) with a total of n elements.

$$RI(T, L) = \frac{a + b}{n(n-1)/2} \tag{2.2}$$

2.11 Entity Resolution

Using multiple data sources helps uncover valuable insights from the combined information. This integrated data often reveals insights that are too difficult or time-consuming to find manually. However, a major challenge is matching entities from different sources to represent the same real-world entity, a process called Entity Resolution (ER).

ER is also referred to as record linkage, deduplication, the merge-and-purge problem, or data matching. These terms are often used interchangeably by researchers [Papadakis et al., 2023, Herath et al., 2021, Ebraheem et al., 2018, Binette and Steorts, 2022, Jehangir et al., 2023]. Notably, the term *record linkage* is described by Binette and Steorts [2022] as the process of assembling the pages of a book into a volume. While this concept will not be applied in the current research, it is considered a promising next step that could offer valuable insights in future studies.

Entity Resolution (ER) faces several challenges. One major issue is the lack of large labeled datasets needed to train complex models effectively. Additionally, since data often originates from diverse sources, it may suffer from low quality due to typographical errors or inaccuracies introduced during processes such as Optical Character Recognition (OCR).

The available data for comparison is typically vast, requiring the use of fast and lightweight models. Entity names often present ambiguities that can only be resolved by leveraging additional attributes. Furthermore, a single real-world entity may be represented by multiple names, such as a commercial name, brand name, or fiscal name. In the case of individuals, they may use nicknames, aliases, or abbreviations of their full registered names, adding complexity to the resolution process.

Binette and Steorts [2022] categorizes ER methods into two groups: deterministic and probabilistic. Deterministic methods include techniques such as exact matching and matching based on similarity functions (e.g., Levenshtein distance, Jaro-Winkler, Jaccard, cosine similarity, etc.). In contrast, probabilistic methods rely on machine learning approaches, including unsupervised, semi-supervised, and fully supervised models.

Benjelloun et al. [2009] introduced the Swoosh (G-Swoosh, R-Swoosh, and F-Swoosh). Despite some differences between these variations, they share a common approach: leveraging a dominant dataset. This dominant dataset consists of entities with a richer set of attributes compared to the dominated dataset. When a match is identified, new attributes from the dominated dataset are merged into the corresponding dominant entity, enriching its representation.

Herath et al. [2021] proposed converting names into multidimensional vectors, creating a matrix of distances between all the records. The main problem is the high demand of computational resources to build and analyze the matrix. The authors in Yao et al. [2022] proposed a Graph Hierarchy model (HierGAT) that transforms embeddings of entities, attributes, and tokens into a hierarchical relational structure. These graphs are then aggregated and processed through a dense classifier network. However, the primary drawback of this approach is its reliance on a large amount of data for effective training, which can limit its applicability in scenarios with limited labeled datasets.

Ebraheem et al. [2018] introduced DeepER, a model based on a bidirectional RNN with LSTM, which requires labeled data for training. The approach utilizes GloVe embeddings to convert words into vector representations, which are then processed by the RNN and LSTM. While using pre-trained GloVe embeddings reduces the effort compared to training a network from scratch, the authors noted that the model needs improvement to effectively handle data quality issues.

In contrast, Tang et al. [2022] proposed a GPT-3-based model for generic ER, leveraging zero-shot or few-shot learning to minimize the need for extensive training data. However, this approach exhibited lower performance compared to other deep learning-based methods. To address this, the authors also introduced an Encoder-Matcher architecture, which delivers results comparable to domain-specific models. Despite this, the details of the model's implementation were not clearly presented.

Most ER researchers agree that the process typically follows a pipeline or workflow to address the problem in similar stages. These stages often include blocking, filtering, and clustering. Papadakis et al. [2023] outlines a workflow that begins with *block building*, where records that are significantly different are discarded. This is followed by *block purging or cleaning* to remove unnecessary or low-quality blocks, *deduplication* to eliminate redundant records, *block filtering* to further reduce the number of compared records, and finally, *comparison cleaning*, where the final matches are identified.

In contrast, Binette and Steorts [2022] proposes a sequence starting with *attribute alignment*, where common attributes are identified. This is followed by *blocking*, where similar records are grouped, *entity resolution*, where similar records are matched, and finally *canonicalization*, where matched records are merged.

Similarly, a survey by Christophides et al. [2020] suggests a pipeline beginning with *blocking or indexing*, which aims to eliminate as many unnecessary comparisons as possible. Next is *block processing*, where dissimilar records within each block are discarded, duplicates are removed, and a *matching function* is applied to the remaining records. The final stage is *clustering*, where similar records are grouped into clusters.

In this research, the challenges of handling vast amounts of data and the need for fast and lightweight processing are addressed, alongside the issue of limited labeled data. The proposed phases are derived from previously mentioned workflows and adapted to meet the specific requirements of this study.

2.12 Graph Analysis

A graph is defined as a pair of sets $G = (V, E)$, where V represents the vertices (or nodes), and E represents the edges (or relationships/links) connecting these vertices. These nodes are interconnected through edges, which depict relationships or associations between them. While many graph frameworks and methods use their own syntax and language for building and querying nodes and relationships, graphs can be represented using various structures, such as lists (e.g., adjacency lists, edge lists) or matrices (e.g., adjacency matrices, incidence matrices, Laplacian matrices). These representations are particularly useful for computational graph algorithms. Choe and Garas [2021] applies graph-theoretical methods for bankruptcy prediction. Their approach involves leveraging graph and matrix-based techniques to construct solvency matrices. The authors generate graphs using financial performance indicators as the foundation and then compute a financial solvency matrix to evaluate the stability and solvency of businesses.

The authors in Ribeiro et al. [2019] explore patterns in graphs by conducting an isomorphic search to identify structural similarities. Their analysis considers factors such as industrial performance, management practices, financial flexibility, credibility, competitiveness, and operating risk. By identifying graph motifs, they distinguish clusters of healthy and unhealthy companies, highlighting the pivotal role of relational structures. These findings underscore that leveraging graph patterns can greatly improve the accuracy and performance of bankruptcy prediction models. As the term implies, graph isomorphism involves determining whether two graphs have the same structure, even if their labels differ. It focuses on matching the connections between nodes, rather than their specific names or identifiers.

As noted by Bi et al. [2022], traditional predictive methods, such as regression models or deep neural networks, treat each company individually. In contrast, tribe-style graph analysis involves analyzing a company in the context of its network of stakeholders. The authors define a *tribe* by connecting a company with its shareholders and then establishing inter-graph links using a news dataset. Their findings reveal that a *non-risky company* typically forms a star-shaped graph, where its shareholders are relatively disconnected from each other. In contrast, *risky companies* exhibit a much denser network of connections among their shareholders. For their graph analysis, the authors used five metrics: degree centrality, eigenvector centrality, clustering coefficient, average number of bridges, and central node degree. The results of these metrics consistently support their conclusions about the structural differences between risky and non-risky companies.

As discussed by Stefko et al. [2021], clustering methods offer an alternative to traditional statistical approaches and theoretical models for predicting business bankruptcy. The authors employ graphical methods, using Multidimensional Scaling (MDS) to analyze outliers and Principal Component Analysis (PCA) to examine key health indicators. The results from these methods are then fed into a Data Envelopment Analysis (DEA) model to derive critical indicators and assess the financial health of a business. MDS operates in a multidimensional space to uncover hidden dimensions that explain similarities or differences among companies, relying on a proximity matrix to evaluate distances. However, a significant limitation of MDS is that the new dimensions cannot be easily linked to specific companies for detailed interpretation. PCA, on the other hand, is a dimensionality reduction technique that retains only the dimensions with the most significant impact on predictions. Finally, DEA is a comparative method that evaluates each unit or producer relative to the best-performing units. Based on the key features identified by MDS and PCA, companies are grouped into clusters that reflect their financial health.

2.13 Summary

This chapter primarily describes the current state of the art concerning the various architectures used in this thesis, as well as previous architectures proposed to address similar problems. The first subsection focuses on the different approaches to risk prediction in the current state of the art, many of which are based on numerical data or short texts such as tweets and news articles. Since the proposed approach involves using multiple datasets for a multidimensional risk analysis, it heavily relies on NLP. This section outlines the key concepts and significant research in NLP, with a particular emphasis on DNN models, including the Attention Mechanism and Transformers, which are among the most important models proposed.

The architectures derived from the Transformer, specifically the BERT and GPT models, serve as the foundation for the best-performing models in classification and generative tasks, respectively. Additionally, the main concepts of tokenization and embeddings, which are fundamental to NLP models, are detailed. Finally, the chapter includes an overview of the concepts and current state of the art in Information Extraction and Document Clustering, which are integral to the later stages of the present research data pipeline and the core concepts of Entity Resolution and Graph Analysis.

Chapter 3

Methodology and Proposed Architecture

Contents

3.1 Introduction 40

3.2 Research Design 40

3.3 Dataset 40

3.3.1 European Commission: List of NACE codes 41

3.3.2 Luxembourg Business Registers (LBR) 41

3.3.3 Institute of Business Auditors (IRE) 44

3.3.4 Chambre des notaires du Grand-Duche de Luxembourg . 45

3.4 General Methodology 45

3.4.1 Data Gathering and Refining 46

3.4.2 Information Enrichment 47

3.4.3 Data Analysis and Insights Generator 47

3.4.4 Presentation and Knowledge Exploitation 47

3.5 Component-Specific Methodologies 48

3.5.1 Basic Software Engineering 48

3.5.2 Data Exploratory 49

3.5.3 Supervised and unsupervised Learning 50

3.5.4 Agile 53

3.6 Technologies and Proposed Architecture 55

3.6.1 Application Architecture View 55

3.6.2 Deployment View 58

3.7 Summary 61

3.1 Introduction

This chapter presents the technical details of the proposed platform. The primary components of the solution utilize Machine Learning and Deep Learning models. Consequently, a quantitative research design is required, along with high-quality data and IT infrastructure capable of handling large volumes of data.

This chapter explains the selection of the research design type and provides a brief description of the dataset, detailing its features and size. The methodology is then presented as a pipeline of steps divided into three main stages. Finally, the technology and IT infrastructure used to support these components are detailed.

3.2 Research Design

Given the nature of Machine Learning and Deep Learning models, this study adopts a quantitative research design. This approach systematically utilizes data, deriving results through mathematical and statistical methods.

To address the research questions and achieve the research goals, data was collected from multiple sources. By employing various Machine Learning models, we aim to enhance the accuracy of the risk prediction model.

Machine Learning and Deep Learning models systematically use statistical methods to identify patterns and make predictions. These models rely heavily on the quality and diversity of the data. Therefore, ensuring high-quality input data is essential for generating accurate inferences and predictions.

3.3 Dataset

In the current research, four datasets were utilized. These datasets are considered reliable as they originate from Luxembourgish government websites or private institutions endorsed by the government. Details about how the data was collected is detailed in section 4.2.

3.3.1 European Commission: List of NACE codes

Countries within the European Union have agreed to adhere to various standards and recommendations. One such standard is the hierarchical classification system used for organizing companies for statistical and management purposes, known as Nomenclature of Economic Activities code (NACE) codes. These codes are assigned to companies based on their primary activities, which fall into one of the specified classes.

The NACE code is structured into four hierarchical levels: Section (S), Division (D), Group (G), Class (C).

Table 3.1 shows detailed statistics on the composition of NACE codes. This table presents data published by the European Commission.

Table 3.1: Number of divisions(D), groups(G) and classes (C) per section(S)

Code	**Section (S)**	**D**	**G**	**C**
A	Agriculture, forestry and fishing	11	25	26
B	Mining and quarrying	1	2	2
C	Manufacturing	82	156	159
D	Electricity, gas, steam and air conditioning supply	3	7	7
E	Water supply; sewerage; waste management and(...)	6	8	9
F	Construction	9	21	29
G	Wholesale and retail trade; repair of motor vehicles(...)	21	91	107
H	Transporting and storage	13	20	20
I	Accommodation and food service activities	7	8	10
J	Information and communication	13	26	26
K	Financial and insurance activities	10	17	33
L	Real estate activities	3	4	4
M	Professional, scientific and technical activities	15	19	25
N	Administrative and support service activities	19	33	35
O	Public administration and defence; compulsory S.S.	2	3	3
P	Education	4	8	8
Q	Human health and social work activities	8	11	13
R	Arts, entertainment and recreation	5	14	14
S	Other services activities	6	17	18
U	Activities of extraterritorial organisations and bodies	1	1	1

3.3.2 Luxembourg Business Registers (LBR)

The LBR oversees the government platform[1], which offers public access to information about Luxembourgish companies. In our context, when we mention the LBR dataset, we are referring to the data obtained

[1] www.lbr.lu

from this source. This dataset includes both company information and associated documents.

The main information we obtain from the LBR website is the company name, commercial and trade name if it exists, the registered address, registration date, type of legal form, company NACE code, and business status (Deleted or Active).

In this section, we describe the characteristics of the LBR dataset and present statistics about the entire dataset used for this research. Each chapter that involves model training will include a brief description of the specific characteristics of the LBR dataset and any additional manual labels created for those particular purposes.

As explained in section 1.1, in 2023 there were approximately 40,000 SMEs registered in Luxembourg. However, for this project, a reduced dataset consisting of 10,000 SMEs is used. Figure 3.1 shows the distribution of SMEs per year in the dataset. It can be observed that most of the registered companies appear to be from after the year 2000. This is because when the LBR platform was released (2002/2003), only the active companies at that time were registered. Consequently, most of the companies that had already closed are not considered in this dataset or even in the entire LBR data.

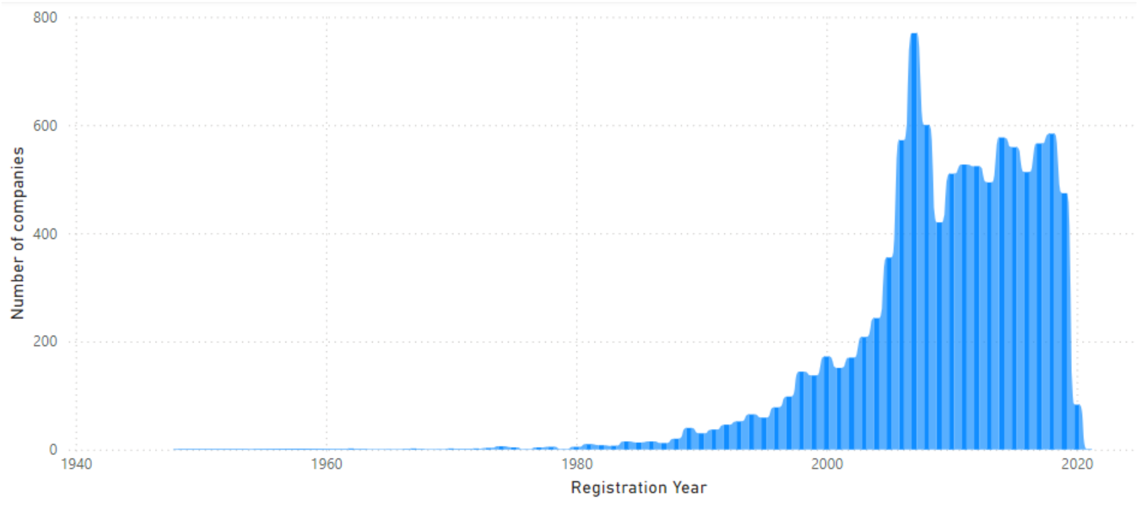


Figure 3.1: LBR: Number of registered SMEs per year. The small number of registered companies from 1940 is mainly because the LBR only keeps information from active companies when it started in 2002/2003, that is why there is a significant increment after that.

The previous section explains the NACE standard, and Table 3.2 shows the distribution of the dataset according to the assigned NACE codes. It is notable that most of the SMEs in Luxembourg are in the Financial and Insurance Activities sector (≈64.1%).

Table 3.2: LBR: Number of SMEs per NACE category

Code	NACE Section	#SMEs	%
A	Agriculture, forestry and fishing	16	0.2
C	Manufacturing	40	0.4
D	Electricity, gas, steam and air conditioning supply	3	0.0
E	Water supply; sewerage; waste management and(...)	5	0.0
F	Construction	270	2.7
G	Wholesale and retail trade; repair of motor vehicles(...)	911	9.1
H	Transporting and storage	85	0.9
I	Accommodation and food service activities	291	2.9
J	Information and communication	321	3.2
K	Financial and insurance activities	6,408	64.1
L	Real estate activities	294	2.9
M	Professional, scientific and technical activities	779	7.8
N	Administrative and Support service activities	304	3.0
P	Education	77	0.8
Q	Human Health and social work activities	33	0.3
R	Other services activities	46	0.5
S	Activities of extraterritorial organizations and bodies	117	1.2
	Total	10,000	100

Table 3.3 shows the number of documents available for download for the 10,000 SME's. The table shows that most of the documents are in French (≈88%), and the rest in German(≈6.7%) and English(≈5.11%).

Table 3.3: SME's documents statistics.

	Total
Number of document's types	67
Number of business activities	473
Number of SME's	10,000
Number of reports	99'117
* Number of reports in French	87'374 (88.15%)
* Number of reports in German	6'643 (6.70%)
* Number of report in English	5'100 (5.15%)
Total of pages	674'730

As shown in Table 3.3, there are 67 different document types. This classification is done by LBR and can be added (but not modified or deleted) when deemed necessary. In Table 3.4, some dataset statistics by document type are presented. Most of the registered or filed documents are Annual Accounts and their corresponding modifications. This is because these documents are required for all companies each year, whereas other documents are necessary only when specific business events occur, such as a change of manager, address, deletion, etc.

Table 3.4: Dataset statistics by document type.

Financial Report	Number of documents					Number of pages		Pages/doc.
	French	German	English	Total	Perc.	Total	Perc.	(Avg)
Annual Accounts (AA)	62'257	5'800	4'910	72'967	73.6%	522'345	77.4%	7.16
Modifications AA	8'096	345	72	8'513	8.6%	59'733	8.9%	7.02
Court orders (Bankruptcy)	3'890	0	0	3'890	3.9%	8'082	1.2%	2.08
Registration	2'607	204	0	2'811	2.8%	44'116	6.5%	15.69
Deletion	2'656	42	2	2'700	2.7%	6'242	0.9%	2.31
Articles of Association	1'298	149	7	1'454	1.5%	8'608	1.3%	5.92
Court order - Disolution	1'210	0	0	1'210	1.2%	1'972	0.3%	1.63
Resignation	608	23	0	631	0.6%	1'279	0.2%	2.03
Others	4'752	80	109	4'941	5.0%	22'353	3.3%	4.52
Total	87'374	6'643	5'100	99'117		674'730		

From the average pages per document shown in the previous table, it can be seen that the longest documents are the Registration documents, which contain around 15 pages per document. In contrast, the Annual Accounts and Articles of Association have around 7 and 6 pages respectively, while the rest tend to be shorter documents.

3.3.3 Institute of Business Auditors (IRE)

The Institut des Reviseurs d'Enterprises (IRE) is a professional organization representing auditors in Luxembourg. From this source, we obtained a comprehensive list of auditors and audit firms operating in the country.

This dataset is composed of 70 audit firms which are classified as *Audit Firm: Cabinet de révision* or *Certified Audit Firm: Cabinet de révision agréé*, and 724 auditors that could belong to any of these audit firms or being independents.

The list of auditors can change periodically, specially the relation between the auditor and the audit firm. The Table 3.5 presents a distribution of audit firms and auditors, categorized by the type of audit firm.

Table 3.5: Distribution of Audit Firms and Auditors by Firm Type

Audit Firm Type	**# Audit Firms**	**# Auditors**
Audit Firm	14	35
Certified Audit Firm	56	479
Independent	-	210
Total	70	724

The Table 3.6 presents the top-ranked audit firms in Luxembourg based on the number of registered auditors they employ.

Table 3.6: Top Audit Firms in Luxembourg by Number of Registered Auditors

Audit Firm Type	Audit Firm	# Auditors
Certified Audit Firm	PWC	137
Certified Audit Firm	Ernst & Young	76
Certified Audit Firm	KPMG Audit	73
Certified Audit Firm	Deloitte Audit	68
Certified Audit Firm	Mazars Luxembourg	18
Audit Firm	Cabinet virtuel	21
Audit Firm	Callens, Pirenne, Theunissen & Co	3

3.3.4 Chambre des notaires du Grand-Duche de Luxembourg

The Chambre des notaires du Grand-Duche de Luxembourg is the government authority responsible for overseeing approved notaries in the country. We obtained the list of notaries and their locations in Luxembourg from their website.

36 notaries are distributed among the country divided into 24 localities. 21 localities has assigned only one notary and the rest are shown in Table 3.7.

Table 3.7: Top localities in Luxembourg by Number of Certified Notaries

Locality	# Notaries
Luxembourg	10
Esch-sur-Alzette	3
Mersch	2
Others	21
Total	36

3.4 General Methodology

This section outlines the methodology employed in developing and evaluating a platform that leverages machine learning and deep learning models for predictive and risk analysis. Each component plays a crucial role in enhancing the overall data pipeline process. As depicted in Figure 3.2, the process consists of four sequential phases, representing the key stages in any machine learning project.

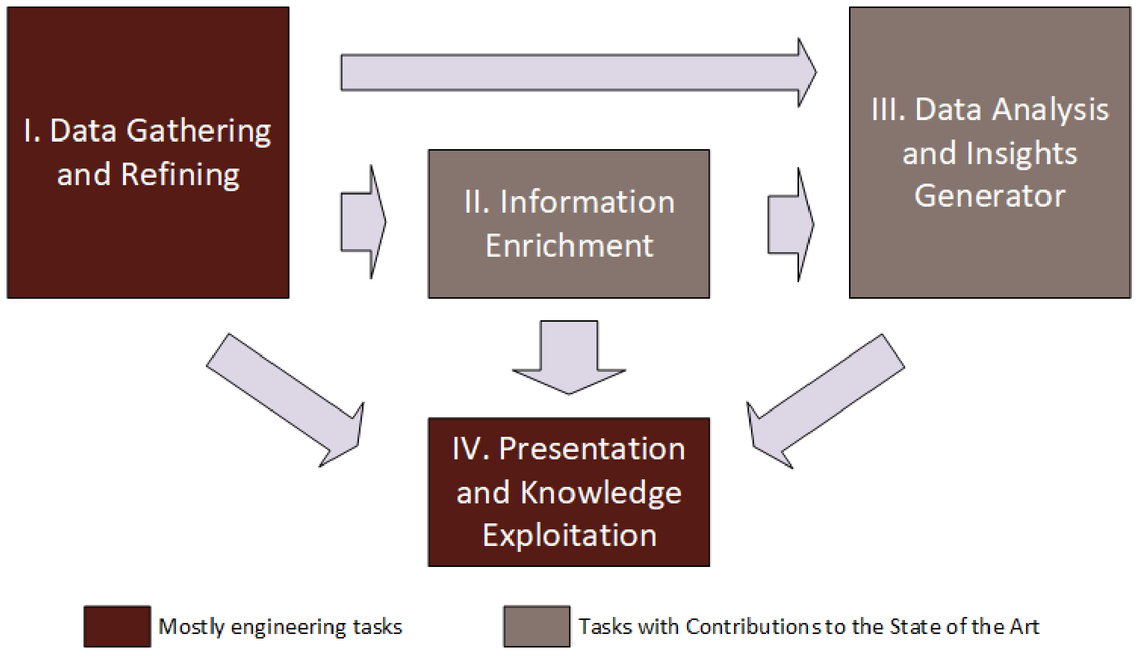

I. Data Gathering and Refining
II. Information Enrichment
III. Data Analysis and Insights Generator
IV. Presentation and Knowledge Exploitation
Mostly engineering tasks
Tasks with Contributions to the State of the Art

Figure 3.2: Data pipeline process. The proposed pipeline consists of four distinct stages, each of which is detailed in the chapters of this thesis. The first and final chapters do not contribute to the state of the art, unlike the second and third stages, which do.

The first and last phases primarily involve engineering tasks, while the second and third phases focus on research, making clear contributions to the state of the art. As illustrated in Figure 3.2, the data collected and refined during the initial phases, along with the enriched data and insights generated from the subsequent phases, are utilized in the final phase.

3.4.1 Data Gathering and Refining

A detailed data flow is illustrated in Figure 4.1 in chapter 4. During this phase, diverse datasets are collected and cleaned. The primary and more complex data source is the information from the LBR. Processing this data involves a series of tasks that should be parallelized while maintaining a certain sequence.

The LBR data primarily comprises company metadata and a series of PDF documents. As a result, OCR and data cleaning procedures are employed to extract both textual and numerical information from these documents.

For resource management, a basic server is needed to handle the OCR extraction tasks, along with a database and file server to manage the intermediate processing steps.

The main components in this phase are crawlers, scrappers, OCR and massive insertion tools. For these components, the main methodologies relies on a basic software engineering methodology which is detailed in subsection 3.5.1.

3.4.2 Information Enrichment

A comprehensive data flow for this phase is illustrated in Figure 5.1 in chapter 5. During this stage, new data is generated from the previously collected datasets. This involves the application of various machine learning and deep learning models designed for tasks such as classification, information extraction, and creating diverse data representations.

The primary components of this phase include both machine and deep learning models. These models follow an exploratory data analysis methodology combined with supervised learning techniques, as detailed in subsection 3.5.2 and subsection 3.5.3.

3.4.3 Data Analysis and Insights Generator

The detailed data flow for this phase is shown in Figure 6.1 in chapter 6. This phase focuses on generating new knowledge and predictions, such as identifying clusters and predicting risk levels. Techniques utilized include Deep Learning, Machine Learning, and Graph Neural Networks.

Similar to the previous phase, this phase also employs a structured methodology. It begins with exploratory data analysis (subsection 3.5.2) followed by the application of both supervised and unsupervised learning methods (subsection 3.5.3).

3.4.4 Presentation and Knowledge Exploitation

This section can be considered a software development project by itself because it requires the use of a complex methodology to satisfy the user requests. The main request is still the main goal of this thesis which is the identification of analysis and assessment of various dimensions of risk for SMEs in Luxembourg. The complexity arises when we are defining how to present this results to the final user, that is why for this step of the current work, we are using an own methodology based on Agile Methodologies. This methodology is defined in detail in subsection 3.5.4 respectively.

3.5 Component-Specific Methodologies

3.5.1 Basic Software Engineering

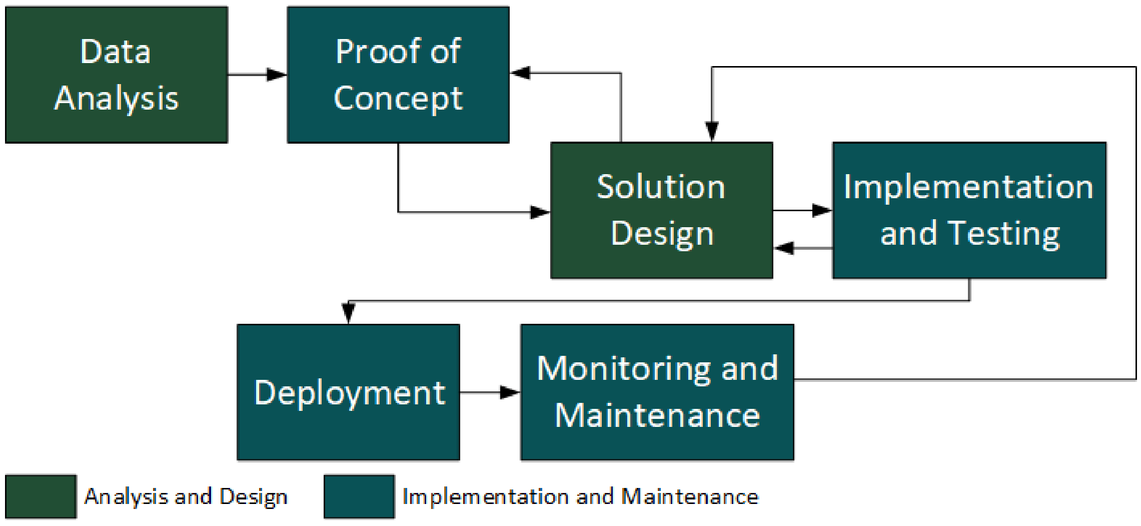

Data Analysis
Proof of Concept
Solution Design
Implementation and Testing
Deployment
Monitoring and Maintenance
Analysis and Design
Implementation and Maintenance

Figure 3.3: Basic Software Engineering Methodology. The methodology follows a cyclic process that involves analyzing the available information to create a proof of concept using various tools and architectural ideas. Once candidate architectures are defined, the solution is designed, implemented, and subsequently deployed and monitored. If new improvements are identified, the process returns to the solution design phase for further refinement.

This methodology provides the foundation for implementing tools with existing libraries. It addresses specific issues, such as data collection from particular sources. The steps involved in this methodology are illustrated in Figure 3.3 and are explained in the following lines:

- *Data Analysis*: This activity evaluates the data according to the component's goal. The available data and the desired output are analyzed to fulfill the requirements.

- *Proof of concept:* Based on the available data, the current tools are analyzed, and a proof of concept is conducted for each tool.

- *Solution Design:* After analyzing and comparing these tools, a solution is designed, which may involve a single step or multiple steps, using one tool or a combination of several. The proposed design(s) include an evaluation method to ensure objectivity for the resulting solution. This activity also includes storage analysis, such as determining whether to save data directly to the database or use intermediate storage files.

- *Implementation and Testing:* Develop the main components and evaluate each proposed design if multiple designs are considered. New designs can be generated, and the corresponding changes implemented. The development includes log files to analyze the performance of the execution.

- *Deployment:* Deploy the solution for large-scale execution. These tools are not independent services but need to be run in an independent container. Consequently, the deployment container should be created or updated. Configure the corresponding ports and IP authorizations to complete the deployment.

- *Monitoring and Maintenance:* Monitor the system and handle any unforeseen issues. If use cases that were not considered during design and development arise, they can be included as updates to the current version or considered for a new version.

3.5.2 Data Exploratory

This foundational step is crucial for both supervised and unsupervised learning projects. It encompasses key activities performed before the data is used in modeling and analysis. The main goals are to understand the data, identify gaps, and prepare it for subsequent analysis. Figure 3.4 illustrates the activities involved in this methodology. Due to potential data availability issues, the project may conclude at this stage without proceeding to the supervised or unsupervised phases. The key activities in this step include:

- *Requirement Analysis and Data Analysis:* According to the desired goals, the available information is analyzed. A gap analysis is con-

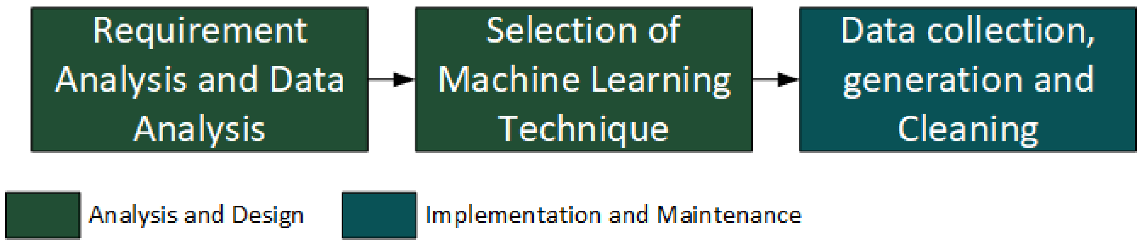

Requirement Analysis and Data Analysis
Selection of Machine Learning Technique
Data collection, generation and Cleaning
Analysis and Design
Implementation and Maintenance

Figure 3.4: Data Exploratory Methodology: This methodology forms the foundation for subsequent methodologies that focus primarily on evaluation, analysis, and design. The process involves analyzing the requirements and data to evaluate various Machine Learning options based on the available data and specific needs. Finally, data collection and generation are carried out according to the selected Machine Learning technique or architecture.

ducted to determine which data and data sources should be used to achieve the desired goals.

- *Selection of Machine Learning Technique:* Determine whether to use a supervised or unsupervised model based on the availability of data and the specific goals of the model. If a supervised model is chosen, the corresponding data labels should be available; otherwise, the data should be labeled manually or programmatically in the next activity.
- *Data collection, generation and Cleaning:* If any data needs to be collected, a basic software engineering project is executed. The collected data should be analyzed and cleaned if necessary, and filtered if required. If data labeling is needed, it can be done manually or through a set of rules. Alternatively, a portion of the data can be labeled using these methods, and then the labels can be applied to other data.

These activities help prepare the data properly and ensure it meets the project's needs, setting up a strong base for later analysis and modeling.

3.5.3 Supervised and unsupervised Learning

As shown in Figure 3.5, for this methodology, during the previous data exploratory steps, we could determine which are the main objectives of the models, analyze that the most appropriate model is a supervised

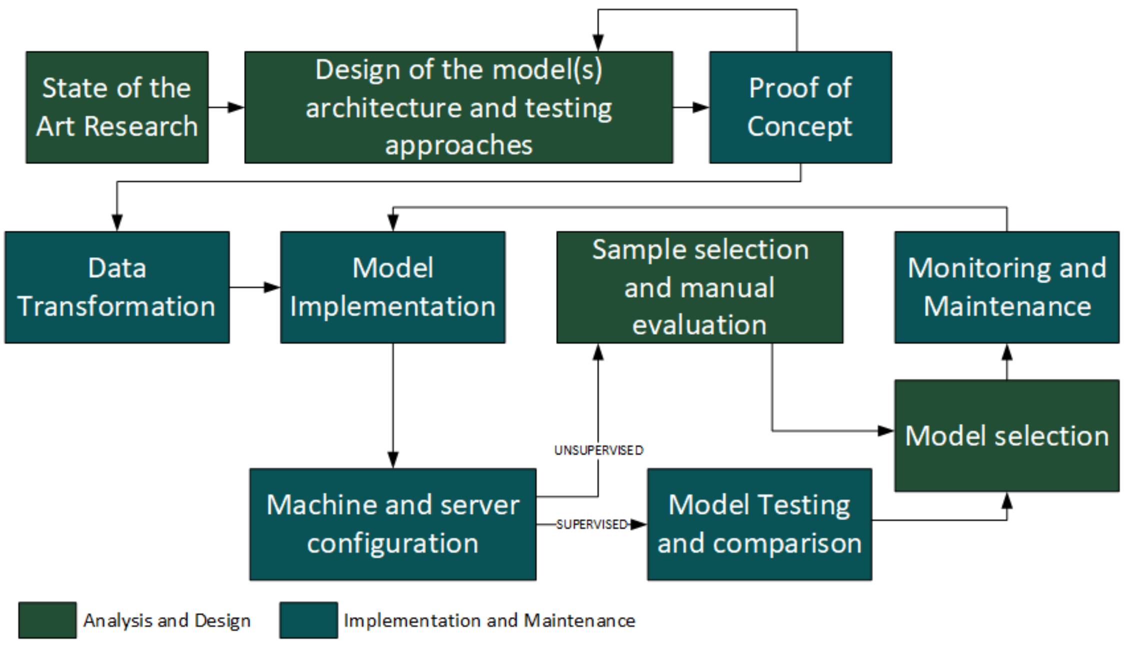

State of the Art Research
Design of the model(s) architecture and testing approaches
Proof of Concept
Data Transformation
Model Implementation
Sample selection and manual evaluation
Monitoring and Maintenance
Model selection
UNSUPERVISED
Machine and server configuration
SUPERVISED
Model Testing and comparison
Analysis and Design
Implementation and Maintenance

Figure 3.5: Supervised and Unsupervised Learning Methodology. The process involves researching the state of the art in relation to the problem and requirements, followed by evaluating alternative solutions and determining whether a selected or newly proposed solution meets the needs. Data transformation and model implementation are then carried out. Prior to the final execution of test cases, the necessary machines and servers must be configured. For supervised learning approaches, labeled data is required, which can be generated manually, automatically, or semi-automatically; otherwise, unsupervised methods may necessitate manual evaluation. After comparing models and configurations, the most suitable model is selected and deployed for monitoring and maintenance. If errors are identified or improvements are needed, the process cycles back to the model implementation phase.

learning approach and to label the required data. In consequence, the following steps includes:

- *State of the Art Research:* This crucial initial step in the process can be undertaken even before the data exploratory analysis. It is important because it involves reviewing the latest technologies and methodologies. By examining how other authors utilize these technologies to achieve their project goals, valuable insights and best practices can be identified and leveraged for our own project.

- *Design of the model(s) architecture and testing approaches:* It is important to analyze the available technological options and propose architectural designs that can address the research problem. Additionally, developing a robust mechanism to evaluate these architectures, whether singular or multiple, is essential for determining the most effective solution.

- *Proof of Concept:* In order to discard some options based on data, infrastructure, or other restrictions, it is important to create proofs of concept for each component. Just using a small dataset and brief tests.

- *Data Transformation:* This step involves processing and transforming the data to make it suitable for analysis. It includes normalizing or scaling numerical values, encoding categorical variables, and engineering new features as needed. If there are multiple architectures, data transformation and model implementation can be performed together to streamline the process.

- *Model Implementation:* This step involves downloading the base models, implementing the designed solutions, and debugging the process from data loading to prediction and testing. It is also important to use a small dataset to accelerate the initial testing process and include mechanisms to save and load the models. Is important to generate information, warning and error logs that will allow us to improve the implementation for unseen scenarios.

- *Machine and server configuration:* This step is necessary when using a supercomputer for training due to the availability of GPUs or to increase speed compared to local training. If not required, for instance, if training can be done on Google Colab, this step should be performed during model training to prepare for deployment. This involves configuring the container, testing the deployment, opening the necessary ports, and performing any other activities required in a production environment.

- *Model Testing and comparison:* This step is done only for supervised learning methods. Testing should be conducted for each designed architecture, with metrics saved for comparison and evaluation. This comparison will enable us to select the best-performing model for deployment.

- *Sample selection and manual evaluation* This step is applicable only for unsupervised learning methods. A representative and diverse set of samples should be selected for evaluation, ensuring that it adequately covers the range of data. However, since manual evaluation is time-consuming, the set should be sufficiently small to allow for thorough review without being overly unmanageable.

- *Model selection:* Based on the results from previous steps, this step select the model that best aligns with the project goals.
- *Deployment* Once the best-performed model has been selected, this is deployed in the corresponding server into a docker container.
- *Monitoring and Maintenance:* During this phase, it is important to monitor the execution logs for warnings or errors that could indicate areas for model improvement or reveal scenarios that were not previously considered.

3.5.4 Agile

- *Elicitation and Customer Analysis*: The first step in this agile methodology involves meetings with key stakeholders and users to gather their needs and main concerns. If there are conflicting requirements, a customer or client prioritization should be performed to select the most critical requirements.
- *Prototyping*: Once the scope and main needs are defined, prototyping and sketching are conducted. This allows for refinement of the requirements through subsequent meetings with the most important users.
- *Define product backlog*: After defining the business requirements and clarifying the sketches for the initial version, the requirements are prioritized and grouped into small, incremental sprints. Each sprint includes various development layers such as front end, back end, security, and infrastructure.
- *Feature Implementation and unit testing*: The development and implementation of each component are carried out in sprints. Each sprint is tested as a unit. Once a sprint is successfully tested, the development and testing of a new sprint begin, continuing the cycle until an MVP is completed. Server configuration and security should be addressed before the next step and, if possible, tested during this phase.

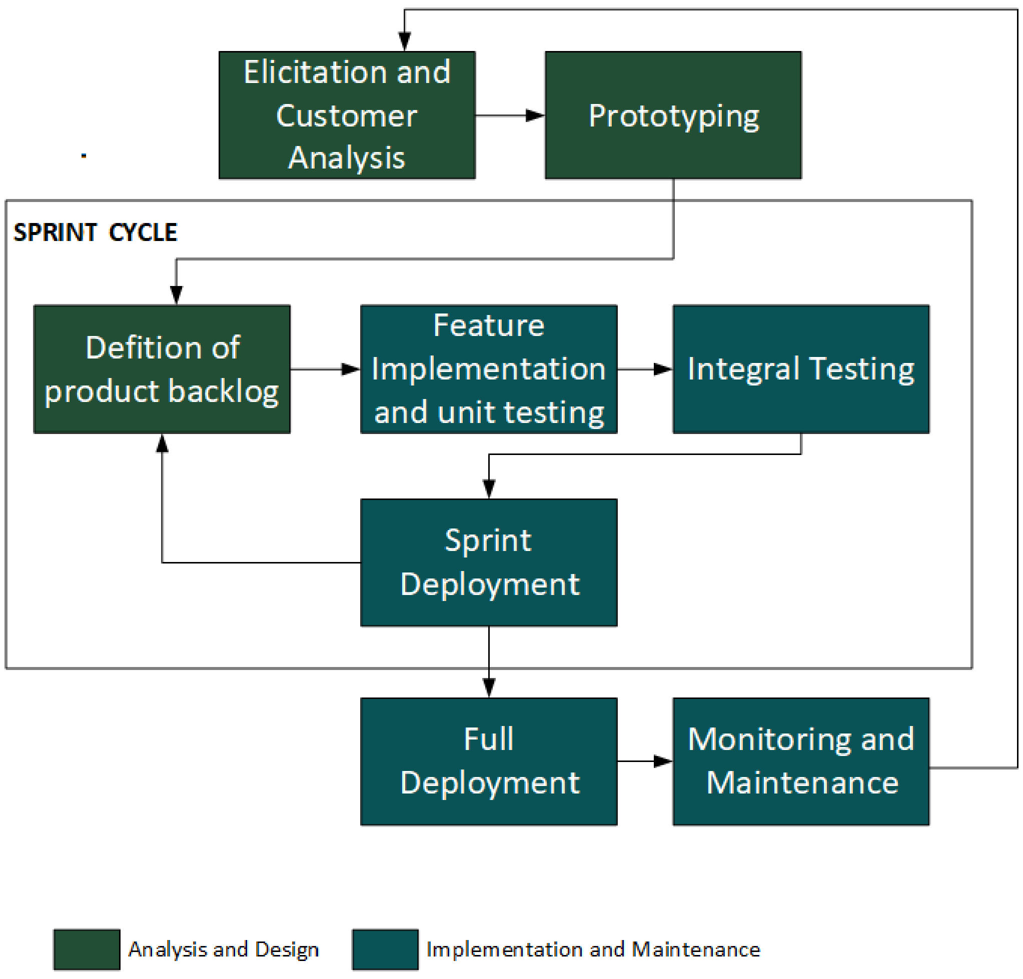

Elicitation and Customer Analysis
Prototyping
SPRINT CYCLE
Defition of product backlog
Feature Implementation and unit testing
Integral Testing
Sprint Deployment
Full Deployment
Monitoring and Maintenance
Analysis and Design
Implementation and Maintenance

Figure 3.6: Agile Methodology. This methodology is used for implementing the user interface and adheres to the Sprint cycle. It involves incremental updates with a product backlog that outlines the components or functionalities to be developed. The process includes implementing each component, conducting comprehensive testing, and deploying the updates. Several sprints are repeated until a Minimum Viable Product (MVP) is achieved and full deployment is completed. Following deployment, the system is monitored, and based on feedback and observations, a new cycle begins.

- *Integral Testing*: When the Minimum Viable Product (MVP) is ready, integral testing of all sprints is performed. Testing users are created, and credentials are distributed among them. Observations and errors are addressed and retested, while new features suggested by users are added to the product backlog for future analysis.

- *Sprint Deployment*: Machine learning and deep learning components are deployed on the appropriate servers according to their GPU needs. Each component updates the corresponding credentials, and deployment is automated with a workflow. The cycle returns to redefining the product backlog and continues until all components of a certain version are completed.

- *Full Deployment*: Once the components have passed the tests, they are deployed to a production environment. Production credentials are updated and distributed among the final users.
- *Monitoring and Maintenance*: Maintenance involves gathering client observations and reports, addressing urgent and priority fixes, and analyzing new requirements. Another round of elicitation is performed as needed.

3.6 Technologies and Proposed Architecture

3.6.1 Application Architecture View

To enhance understanding, Figure 3.7 illustrates the proposed architecture from an application architecture perspective. This diagram depicts the primary components and their interactions within the main applications. To simplify the diagram, components are grouped into categories based on their main objectives. For instance, the "Collecting Services" category includes applications for web scraping, data crawling, and REST-API-based collectors. In this case, the diagram represents this group with general categories labeled "Web Scraping/Crawling" and "API Services" as representative applications.

The diagram categorizes the proposed architecture into eight distinct application groups, described as follows:

- **Repositories:** There are five main types of repositories:
  1) File Server: This server is dedicated to storing and managing files and documents in various formats, such as Excel sheets, JSON, TXT, PDFs, HTML, and more. It is optimized for handling and providing access to a large volume of files and documents.
  2) RDBMS Server: In this case, the server is a PostgreSQL server. It stores data in relational tables and also supports tables for vectors (Pgvector) and JSON documents in an integrated manner.
  3) Cache Server: This is an in-memory key-value database designed for high-performance, low-latency data access. Keys are hashes of

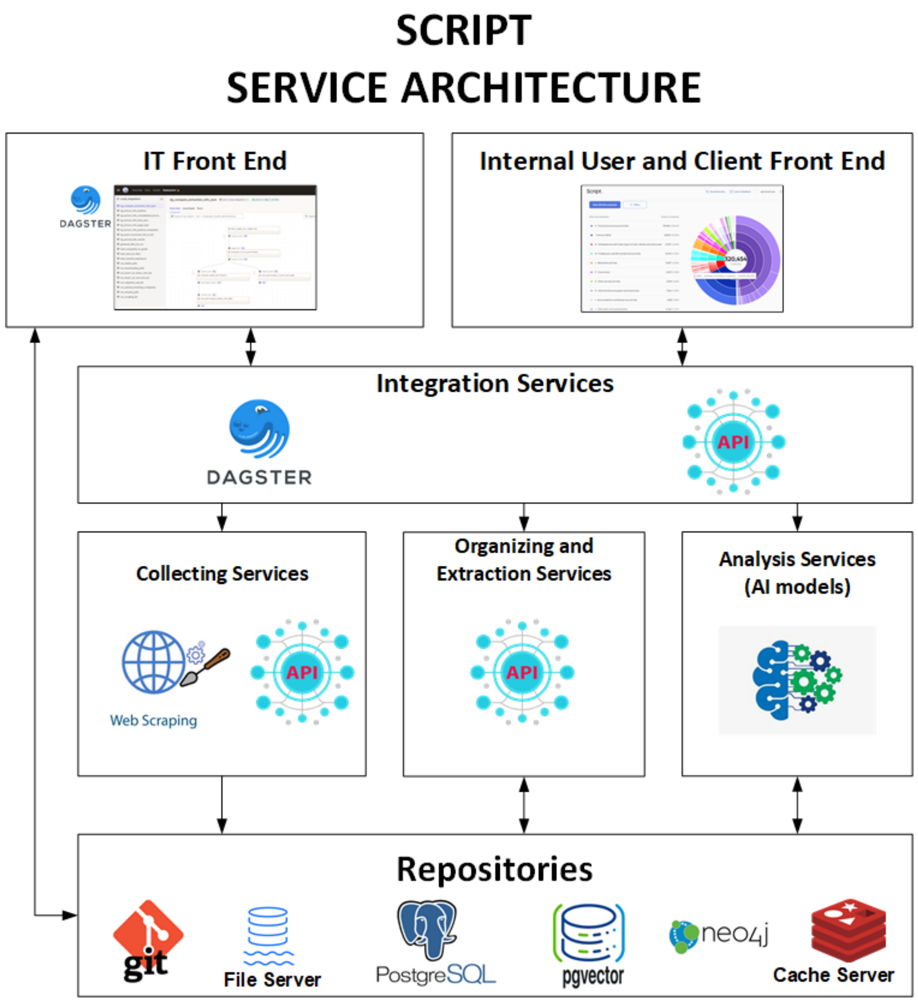


Figure 3.7: Overview of the applications architecture: The proposed architecture illustrates the various repositories utilized in the current project, including Git, a file server, PostgreSQL (As DBMS, Document server and Vector DB), Redis Cache server and Neo4j. The next layer consists of collection services accessed through APIs, which are linked to these repositories. Following this, data extraction and organization services are implemented, and AI models are integrated, all of which connect with one or more repositories and are published through APIs. The subsequent layer is a workflow orchestration layer that integrates these components. It also features an API to facilitate consumption by a final presentation layer. The ultimate layer is the user-friendly dashboard, which includes both the internal user and client front end, as well as the proposed IT front end.

user SQL queries, with results stored as values. As a result, if the same query is requested again, the response time is significantly reduced.

4) Graph Database: Implemented using Neo4j, this database stores data in the form of nodes, edges, and properties, making it well-

suited for managing and querying complex relationships.
5) Code Repository: Implemented using GitHub, this central repository manages code and DevOps workflows, facilitating the deployment of servers into testing and production environments.

- **Collecting Services:** This category of services group all the services used for collecting data from different data sources. Mainly we have three king of data collection:
  1) Manual Data Collection: This approach is typically used when scraping is not feasible, such as when the data is static, small in volume, or available as downloadable files from the website. In such cases, the data can be easily saved into formats like Excel or plain text files.
  2) Data Scraping: Some sources permit and facilitate scraping or crawling, even if the data volume is large and frequently updated. These services typically perform both full scrapes and incremental updates.
  3) API Services: Some sources, often on a subscription basis, provide access to data through secure REST APIs. All these services are detailed in chapter 4. These components were implemented using python 3.

- **Organizing and Extraction Services:** These services assist in post-processing raw data by handling intermediate files or accessing the RDBMS directly. They retrieve files from the file server, clean the data, and insert it into the appropriate data storage. Most of the services belong or to the Data Gathering phase chapter 4 or the Information Enrichment phase chapter 5. These components were implemented using python 3 and PyTorch. The API services to expose diverse Machine Learning services are published internally using python and FastAPI[2].

- **Analysis Services** Each component within this application group consists of either a Deep Learning model or a Machine Learning model. Most of these services use Deep Learning models, which are typically trained through supervised learning. However, some components, such as those oriented towards clustering, utilize Machine Learning models trained with unsupervised learning techniques. Details of all the services in this application group are provided in

[2]https://fastapi.tiangolo.com/

chapter 6.

- **Integration Services:** These services are in charge of the service orchestration to manage the complete data pipeline through the application layers and data availability to the front end layers.
  1) Orchestrator: This component is implemented in Dagster[3]. Dagster is an open-source tool designed to programmatically create data flows for integrating various system components. It offers features to build both on-demand and cron-based workflows. Additionally, Dagster provides tools for monitoring, orchestration, and leveraging parallelism to optimize the execution of data pipelines.
  2) API services: To publish data and online services to end users (both internal and external clients), REST services were developed using C# .NET and Entity Framework. These services are secured at the endpoint level, requiring a valid authentication token for access.

- **IT Front End:** The IT front end is the web interface that allows the IT manager to manage the data workflows. This graphical tool is part of Dagster.

- **Internal User and Client Front End**: Finally, both the client front end and internal user front end were initially developed as proof-of-concept applications using Python and Flask[4]. 1) Internal User Front End: This interface allows IT administrators to manage system access, and monitor any suspicious behavior.
  2) Client Front End: This interface serves as the connection between end users or clients and the system, enabling interaction with the system and providing feedback to facilitate improvements.

3.6.2 Deployment View

The deployment view shows the proposed architecture from a different perspective as shown in Figure 3.8.

The deployment view outlines two main server groups: On-Demand Services and Online Services, which are detailed below. It is important to note that in a production environment with high user traffic, considerations such as redundancy, replication, and load balancing are crucial.

[3]https://dagster.io/
[4]https://flask.palletsprojects.com/en/3.0.x/

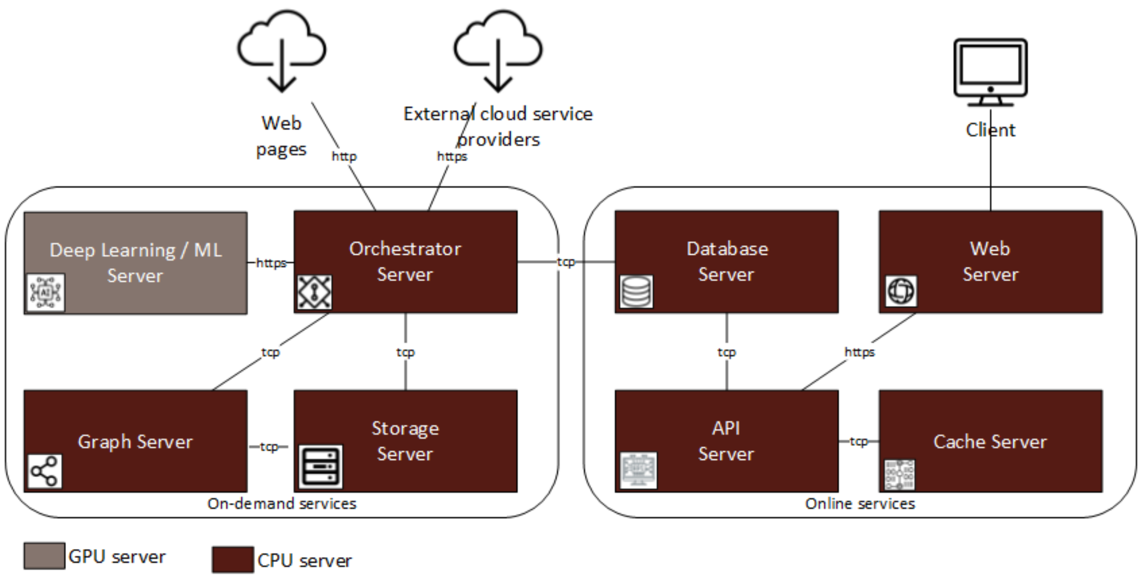


Figure 3.8: Deployment View. This view provides an alternative perspective on the implementation of the solution, emphasizing the hardware requirements for running the various components. It distinguishes between two types of servers based on computational needs: GPU-based and CPU-based servers. The servers are categorized into two groups: those on the left operate in offline and batch modes, while those on the right function in online mode.

Architectural decisions should be based on factors such as the number of concurrent users, service availability requirements, component risks, and the minimum level of fault tolerance needed.

- **On-demand services:** These services collect, clean, enrich, and insert data into the Relational Database Management System (RDBMS) or the graph database. They are triggered either by a scheduled cron job or manually by an IT responsible. On-demand services enable preprocessing and precalculation of predictions, ensuring that data and predictions are readily available when requested by clients.
  1) File Server: This server node consists of two types of file storage.
  Temporal Hot Storage: This is used to store files obtained from data collection during the cleaning process until the data is ingested. Once the data ingestion is complete, the files are moved to the second storage type.
  Permanent Cold Storage: This storage contains files that are no longer actively used or are retained for long-term archival purposes.

This movement of files is required to reduce the server costs.
2) Graph Server: This server node hosts the Neo4j graph database. Currently, it is not part of the online services since all necessary information is pre-calculated. However, it could be implemented as an online service in the future if real-time predictions or responses to direct client queries are required.
3) Deep Learning/Machine Learning Server: This server is typically represented by multiple nodes rather than a single node, with each node supporting various AI components. For example, components such as the Information Extractor may require multiple instances running on a multi-GPU node, while other components, such as the Page Type Predictor, can be implemented on a single GPU.
4) Orchestrator Server: This unique node facilitates the connection of components from data collection, data enrichment, and AI/ML models with databases, including File Server, RDBMS, and Graph storage systems. Dagster is deployed on this node to manage and orchestrate these interactions.

- **Online services:** These services provide end users or clients with interfaces for interaction, ensuring the fastest possible performance. They are commonly found in many web application infrastructures.
1) Database Server: This node contains the RDBMS and it is accessed in an online mode only through the API server.
2) Cache Server: This node contains the Cache database and also it is accessed by the API server.
3) API Server: This node is the online orchestator and is the only interface by any external application (such as direct consumption of third party systems or the web application which is located in the DMZ (Demilitarized Zone)
4) Web Server: This node serves as the primary point of interaction with external users through a user-friendly interface and represents the highest risk in any web infrastructure. It is situated in a separate network to mitigate potential threats. To enhance security, this node is granted limited permissions and access to internal resources. Consequently, any access to internal systems is mediated exclusively through the API server node.

3.7 Summary

This chapter introduces the adoption of a quantitative research design as the mechanism for addressing the research questions. The datasets are explained and analyzed; these datasets will be used in subsequent chapters, contributing additional features and entities to the initially collected dataset. The current research is based on data collected from the LBR, primarily consisting of company information and documents filed and presented by SMEs. The chapter also describes the general methodology and its four phases. Each phase generates multiple components that form part of a larger integrated platform. Additionally, the specific methodology applied to each component is briefly explained according to its nature (software piece or ML/DL model). Finally, the chapter presents the technologies used and the proposed architecture through an application architecture view and a deployment view of the aforementioned large integrated platform.

Chapter 4

Data Gathering and Refining

Contents

4.1 Introduction **62**
4.2 Data Collection **63**
4.2.1 European Commission: List of NACE codes 63
4.2.2 Luxembourg Business Registers (LBR) 65
4.2.3 Institute of Business Auditors (IRE) 66
4.2.4 Chambre des notaires du Grand-Duche de Luxembourg . 66
4.3 PDF downloader **67**
4.4 OCR, extraction and cleaning **68**
4.5 Data ingestion **70**
4.6 Summary **71**

4.1 Introduction

This chapter details the initial stage of the data pipeline process, from collecting raw data to inserting it into the database. Depending on the data source, the process may involve different levels of data cleaning. Direct information from government websites is considered the ground truth, while information found in documents are considered as additional sources complementing this data during the fusion phase.

Most tools developed for this stage do not contribute new knowledge to the state of the art. These engineering steps rely on existing tools, which are essential and necessary for subsequent stages.

The Figure 4.1 shows the main steps of this phase which is going to be discussed in detail in this chapter.

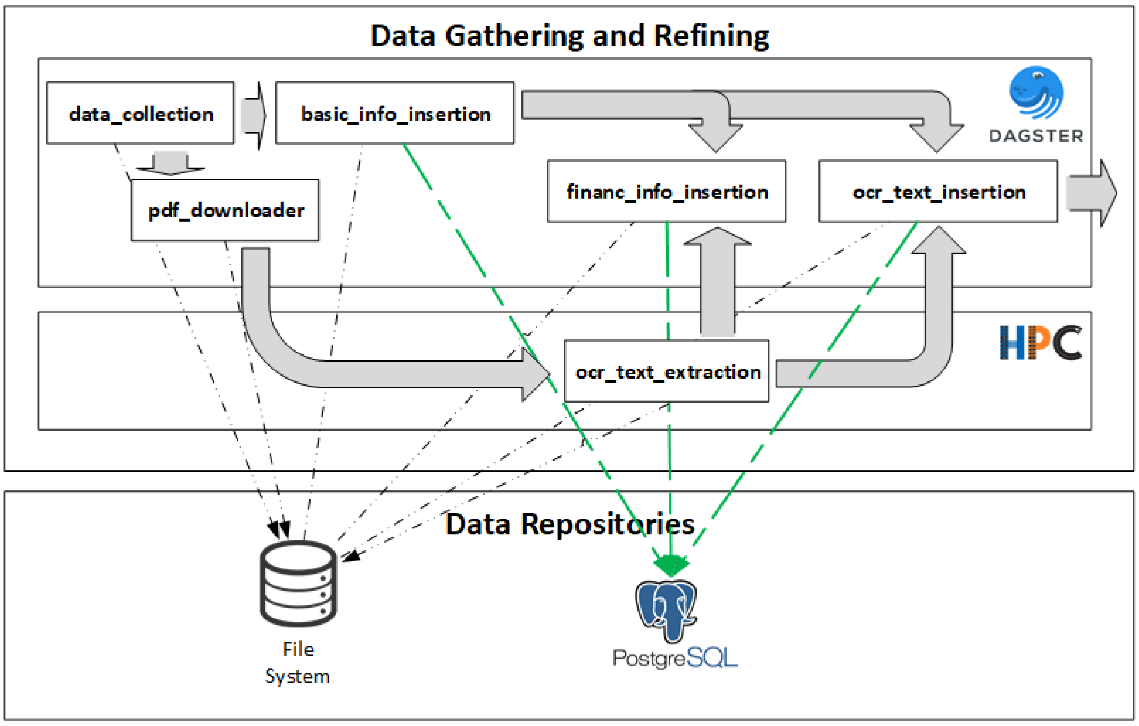


Figure 4.1: Steps for Gathering and Refining stage. This is the initial stage where data is collected from various sources. The primary focus is on gathering LBR data, which includes essential details as well as information extracted from PDF documents. Key components in this stage involve ensuring the accuracy and completeness of data from these diverse sources.

4.2 Data Collection

After understanding the problem and identifying feasible approaches, it is necessary to analyze the available datasets and their collection mechanisms. This section describes the datasets used in this work classified in two groups: Trusted sources and Moderate Reliability sources.

- Trusted Sources: European Comission, Luxembourg Business Registers, Institute of Business Auditors, Chambre des notaires du Grand-Duche de Luxembourg.
- Moderate Reliability sources: Google Maps.

4.2.1 European Commission: List of NACE codes

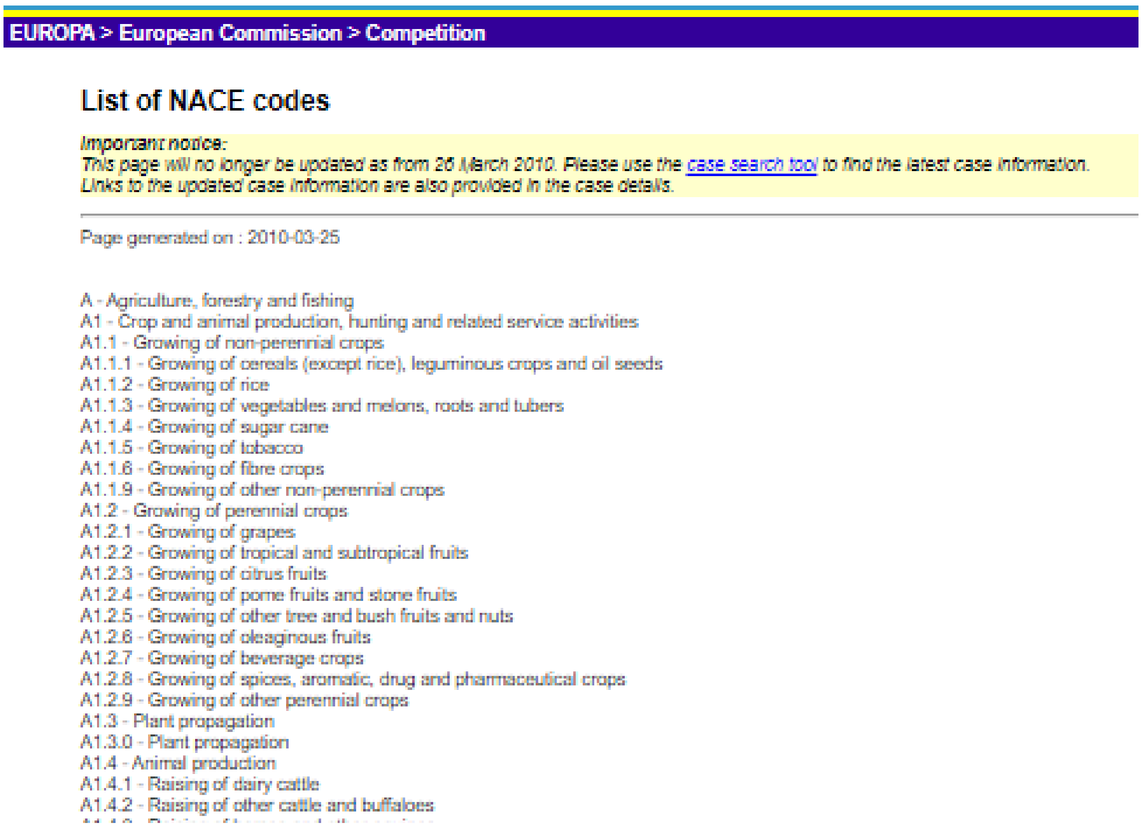
The European commission published on March 26th, 2010 the list of NACE codes[1]. These codes are a classification system to categorize

[1]https://ec.europa.eu/competition/mergers/cases/index/nace_all.html

economic activities and to standardize them in Europe. It consist in a hierarchical set of letters and numbers which explain the hierarchy. Is composed of four levels: Section, division, group and class, for example M70.2.2 which is refereed to Business and other management consultancy activities.

- Section: 1 letter. e.g. M: Professional, Scientific, and Technical Activities
- Division: 2 digits. e.g. 70: Activities of Head Offices; Management Consultancy Activities.
- Group: 3 digits. e.g. 70.2: Management Consultancy Activities
- Class: 4 digits. e.g. 70.2.2: Business and other management consultancy activities.

As shown in Figure 4.2, the list is presented in a single web page, which can be easily copied and then inserted directly to the database.



EUROPA > European Commission > Competition

List of NACE codes

Important notice:
This page will no longer be updated as from 26 March 2010. Please use the case search tool to find the latest case information.
Links to the updated case information are also provided in the case details.

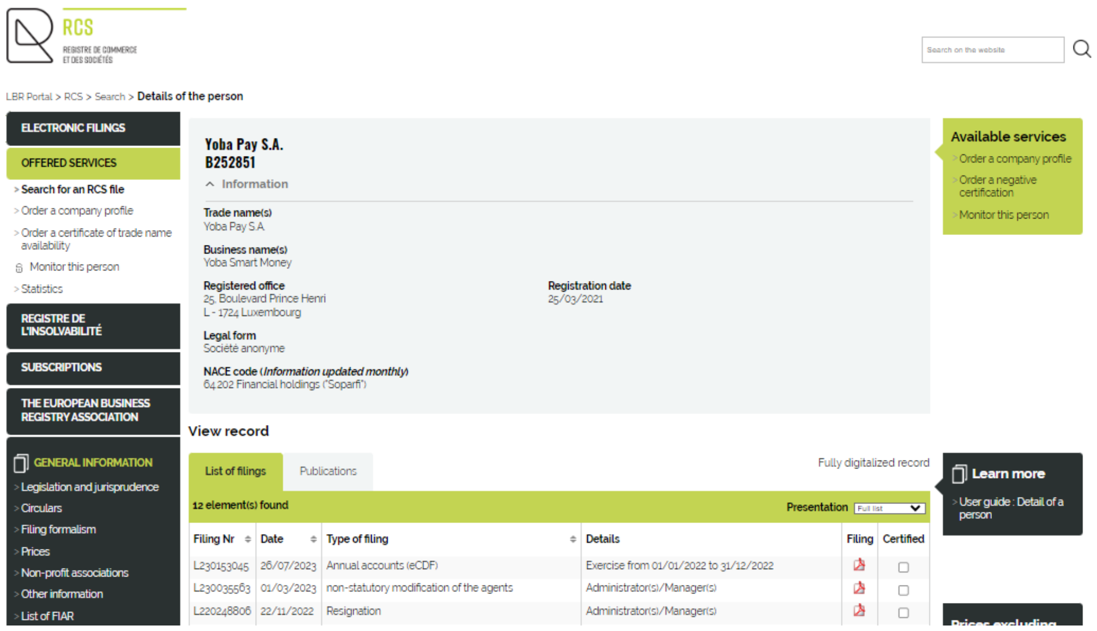
Page generated on : 2010-03-25

A - Agriculture, forestry and fishing
A1 - Crop and animal production, hunting and related service activities
A1.1 - Growing of non-perennial crops
A1.1.1 - Growing of cereals (except rice), leguminous crops and oil seeds
A1.1.2 - Growing of rice
A1.1.3 - Growing of vegetables and melons, roots and tubers
A1.1.4 - Growing of sugar cane
A1.1.5 - Growing of tobacco
A1.1.6 - Growing of fibre crops
A1.1.9 - Growing of other non-perennial crops
A1.2 - Growing of perennial crops
A1.2.1 - Growing of grapes
A1.2.2 - Growing of tropical and subtropical fruits
A1.2.3 - Growing of citrus fruits
A1.2.4 - Growing of pome fruits and stone fruits
A1.2.5 - Growing of other tree and bush fruits and nuts
A1.2.6 - Growing of oleaginous fruits
A1.2.7 - Growing of beverage crops
A1.2.8 - Growing of spices, aromatic, drug and pharmaceutical crops
A1.2.9 - Growing of other perennial crops
A1.3 - Plant propagation
A1.3.0 - Plant propagation
A1.4 - Animal production
A1.4.1 - Raising of dairy cattle
A1.4.2 - Raising of other cattle and buffaloes

Figure 4.2: List of NACE codes from the European Commission

4.2.2 Luxembourg Business Registers (LBR)

The Luxembourg Business Registers [LBR] is an economic consortium that includes the State, the Chamber of Commerce, and the Chamber of Trades of Luxembourg. LBR oversees several key registers, including the RCS, the Electronic Register of Companies and Associations (Electronic Register of Companies and Associations (RESA)), the Register of Beneficial Owners (RBE), and the Insolvency Register (REGINSOL).


RCS
REGISTRE DE COMMERCE ET DES SOCIÉTÉS
Search on the website
LBR Portal > RCS > Search > Details of the person
ELECTRONIC FILINGS
OFFERED SERVICES
> Search for an RCS file
> Order a company profile
> Order a certificate of trade name availability
Monitor this person
> Statistics
REGISTRE DE L'INSOLVABILITÉ
SUBSCRIPTIONS
THE EUROPEAN BUSINESS REGISTRY ASSOCIATION
GENERAL INFORMATION
> Legislation and jurisprudence
> Circulars
> Filing formalism
> Prices
> Non-profit associations
> Other information
> List of FIAR
Available services
> Order a company profile
> Order a negative certification
> Monitor this person
Learn more
> User guide : Detail of a person
Information
View record
List of filings
Publications
Fully digitalized record
Presentation
Full list

Yoba Pay S.A.
B252851

Trade name(s)
Yoba Pay S.A

Business name(s)
Yoba Smart Money

Registered office
25, Boulevard Prince Henri
L - 1724 Luxembourg

Registration date
25/03/2021

Legal form
Société anonyme

NACE code (*Information updated monthly*)
64.202 Financial holdings ("Soparfi")

12 element(s) found

Filing Nr	Date	Type of filing	Details	Filing	Certified
L230153045	26/07/2023	Annual accounts (eCDF)	Exercise from 01/01/2022 to 31/12/2022		☐
L230035563	01/03/2023	non-statutory modification of the agents	Administrator(s)/Manager(s)		☐
L220248806	22/11/2022	Resignation	Administrator(s)/Manager(s)		☐

Figure 4.3: Screenshot of the LBR website with a company profile information

The publicly available registers for download are the RCS, REGINSOL, and RESA. From the RCS and REGINSOL, the available information include the registered name, trading name, registration address, legal form, industry NACE code[2], company status (e.g., deleted, in process of liquidation), and various documents filed for the company, including court orders and judicial decisions. The RESA register publishes a daily journal listing the filings presented on that day. The RBE, which was open to the public from September 1, 2019, until its suspension on November 22, 2022, is no longer publicly accessible.

The most important types of documents are summarized in Table 4.1.

Our business partner provided us with the information obtained from the LBR and a list of document's urls. The characteristics of this dataset was described in the previous chapter.

[2]https://ec.europa.eu/competition/mergers/cases/index/nace_all.html

Table 4.1: List of most frequent filed documents in LBR

Annual Accounts	Modification Documents	Articles of Association
Registration	Deletion	Court Orders
Consolidated accounts	Merger/Demerger	Autorized signature lists
Modification of Agents	Investment Funds	Liquidation
Branch Registration	Associate List	Legal Notice

4.2.3 Institute of Business Auditors (IRE)

As an additional database, we get the list of auditors and auditing companies which is managed by the IRE[3].

The IRE is a professional body representing auditors in Luxembourg. This institute provides training and continuous training, regulating and supervising auditing practices and serving as a bridge between auditors and regulatory entities, They are required to maintain a database of auditors and audit firms with the following information:

- Names and contact information of the auditor and audit firms.
- Registration details and professional qualifications.
- Records of training and education.
- Disciplinary actions or sanctions.

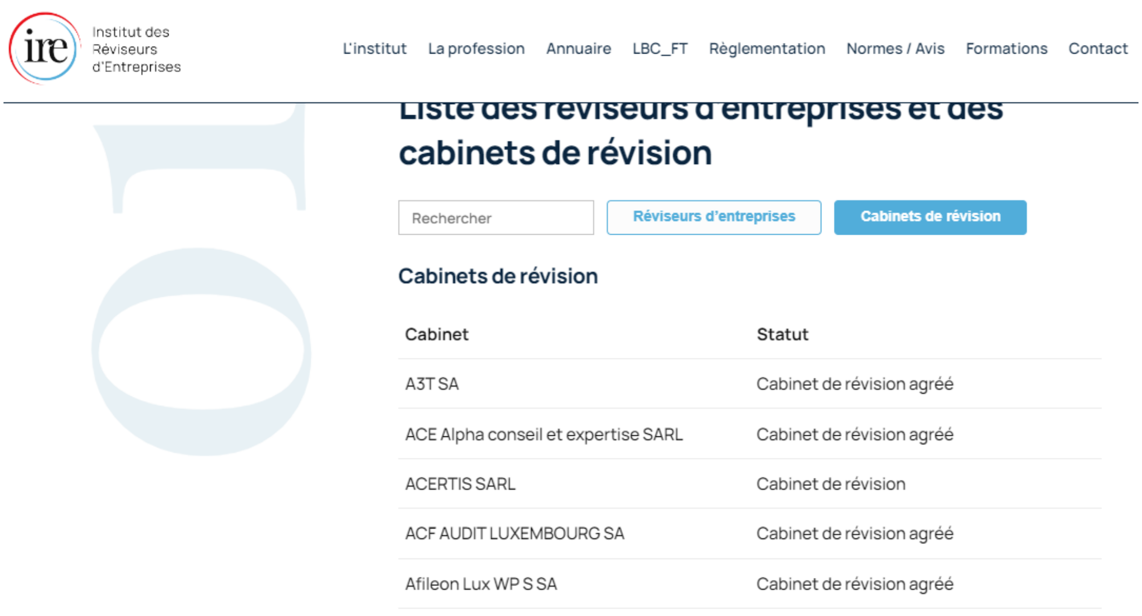
As shown in Figure 4.4, the information publicly available from IRE is the name of the auditor, audit firm if exists and the current status. This information was manually copied into an excel file and then inserted directly into the database.

4.2.4 Chambre des notaires du Grand-Duche de Luxembourg

The Chamber of Notaries of Luxembourg [4] contains the information of the 36 notaries in Luxembourg with their full name, corresponding address, telephone number and e-mail as shown in Figure 4.5. The notaries are distributed among the territory proportionally to the population.

This information also was directly inserted into a excel file and then inserted directly into the database.

[3]https://ire.lu/fr/liste-reviseurs-entreprises-cabinets-revision/
[4]https://www.notariat.lu/trouver-notaire

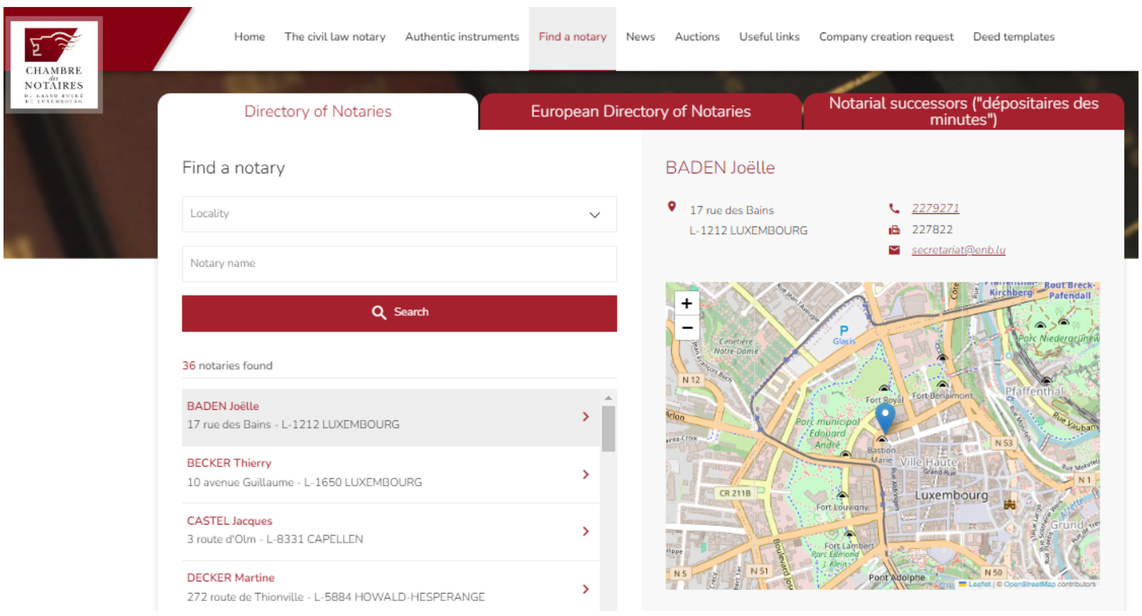


Liste des réviseurs d'entreprises et des cabinets de révision

Cabinets de révision

Cabinet	Statut
A3T SA	Cabinet de révision agréé
ACE Alpha conseil et expertise SARL	Cabinet de révision agréé
ACERTIS SARL	Cabinet de révision
ACF AUDIT LUXEMBOURG SA	Cabinet de révision agréé
Afileon Lux WP S SA	Cabinet de révision agréé

Figure 4.4: Screenshot of IRE website with the public information available



36 notaries found

BADEN Joëlle
17 rue des Bains - L-1212 LUXEMBOURG

BECKER Thierry
10 avenue Guillaume - L-1650 LUXEMBOURG

CASTEL Jacques
3 route d'Olm - L-8331 CAPELLEN

DECKER Martine
272 route de Thionville - L-5884 HOWALD-HESPERANGE

BADEN Joëlle
17 rue des Bains
L-1212 LUXEMBOURG
2279271
227822
secretariat@enb.lu

Figure 4.5: Screenshot of Chambre des notaires website with the public information available

4.3 PDF downloader

With the list of LBR document's urls, we proceeded to download them for processing. These documents are saved in the file system.

4.4 OCR, extraction and cleaning

For the OCR tool, diverse Python libraries have been utilized. The tool includes a 'Generator' class that accepts a list of PDF locations or URLs if download is required. The following process is shown in Figure 4.6. It then distributes the processing of each file among the specified or maximum number of threads. Each thread downloads the PDF file to a temporary directory if it is an URL. Using PyMuPDF[5], an attempt to read the PDF is made. If the PDF is encrypted for reading (the most basic form without a password), decryption is performed. The workload of each document is divided into sequential chunks to avoid memory issues.

The tool is designed to handle large volumes of PDFs efficiently. By dividing the workload into chunks, it minimizes memory usage and maximizes processing speed. This approach allows for the seamless integration of OCR capabilities, ensuring that even encrypted files are processed correctly. With multi-threading, the tool significantly reduces the time required for large-scale document processing, making it ideal for applications needing high-speed OCR.

Using pdf2image[6], each page of the PDF is converted into an image. With pytesseract and a set of pre-configured languages, the tool reads the page image to determine the page orientation and rotates the image if necessary. Once the image is properly oriented, the tool uses dictionaries of stop-words for the pre-configured languages to determine the most frequent language, setting the page language accordingly with the corresponding confidence level. The angle of rotation also determines if the page is in landscape or portrait orientation. If no text is extracted, the page is categorized as "Blank", and no further actions are taken for that page. Using the identified language, pytesseract is used again, but only with the corresponding language, resulting in better OCR accuracy for non-English documents.

Pytesseract allows us to extract text along with its x and y coordinates, enabling the recreation of lines and columns. Once each line is identified, the tool uses PyMuPDF for PDF-readable documents to obtain the corresponding text in HTML format. Many documents, especially those from the financial domain, encrypt parts of the text that

[5] `https://pymupdf.readthedocs.io/en/latest/`
[6] `https://pypi.org/project/pdf2image/`

cannot be read by tools like PyMuPDF and can only be extracted using OCR image conversion. The tool calculates the percentage of encrypted text relative to the total readable text to determine if the page can be categorized as “Encrypted”, providing the corresponding confidence level. This indicator can be used for risk analysis.

Using both the HTML and OCR information, a line-by-line comparison is performed. The OCR text replaces the HTML text where encryption has been detected; otherwise, the HTML text is used if available. Additionally, an extra step is carried out using regular expressions and patterns to extract tables, which are then formatted in JSON, particularly for documents like financial statements.

As a result of the OCR process, the following files are generated:

- Extracted Raw Text: This file contains the raw OCR data extracted from the documents.
- Corrected Text: This file includes the text corrected using OCR and HTML data.
- Metadata: This file stores page-level information, such as language, rotation, encoding status, file name, page number and confidence levels.
- Tables: All discovered tables are saved in this separate file.

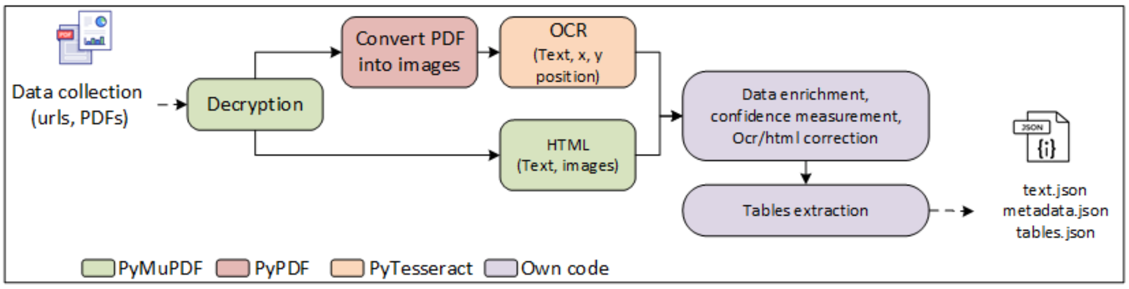


Figure 4.6: Workflow of the OCR Extractor Tool. The tool processes either URLs or PDF files, decrypting them if necessary, as they are often restricted from being read by certain tools. Once decrypted, two parallel actions are performed: first, the PDF text and metadata are directly extracted; second, the PDF pages are converted into images, and an OCR library is applied. The data from both processes is then analyzed to generate new metadata and extract text. Additionally, tables within the PDF are identified and extracted into separate files for further use.

4.5 Data ingestion

The data ingestion process consists of three main tasks: inserting basic information, incorporating text data extracted from the OCR process, and integrating financial information sourced from tables also obtained through OCR.

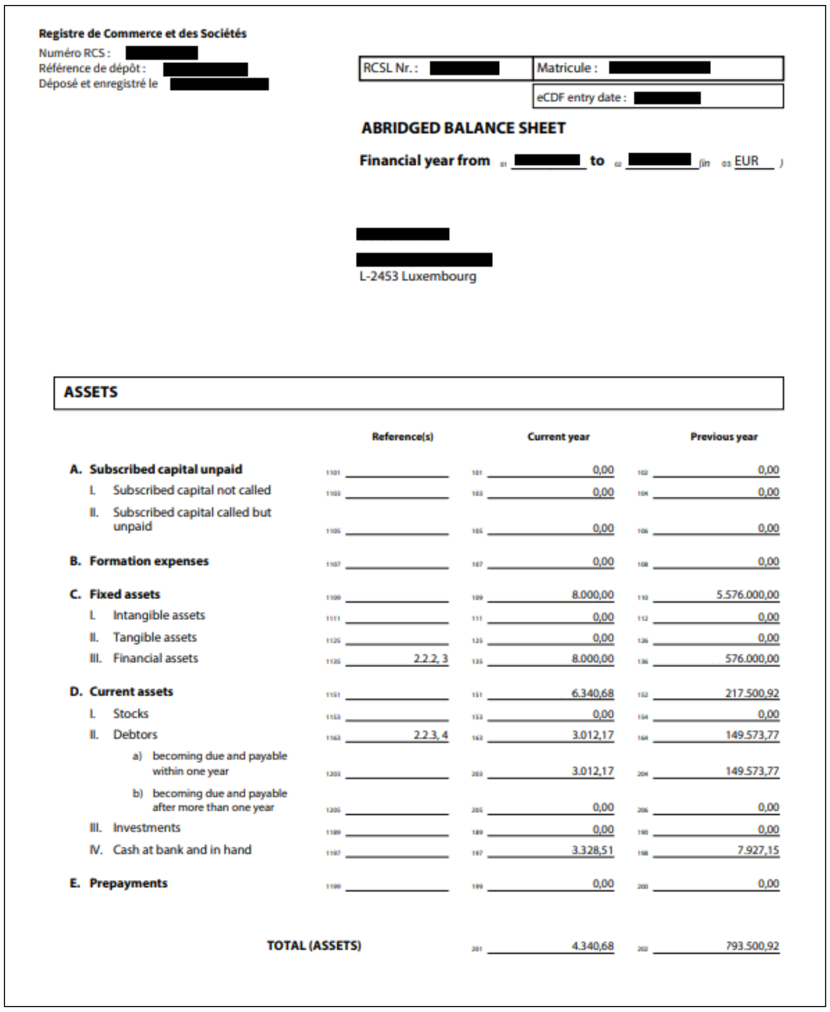
The basic information includes details from Figure 4.3, such as the company name, business code, registered office address, registration date, legal form, NACE code and description, and status.

The text extracted from the OCR process is incorporated as plain text, while metadata is stored as a JSON field. This metadata contains page-level information, including language, content type (Blank, Scanned, Readable, Encoded), encryption status, rotation angle, and orientation (Portrait or Landscape), among other details.

To establish a uniform accounting framework, the International Financial Reporting Standarts (IFRS) was introduced in 2001 to enhance transparency and comparability in financial reporting across different countries. As part of an EU initiative, Luxembourg adopted these standards in 2005 and established the European Centralized Data Facility (ECDF) platform, which has been mandatory since 2011. This platform facilitates the ingestion of data into our database in a more consistent and standardized manner. The current report details the Balance Sheet for each Annual Account and optionally includes the Profit and Loss Statement and financial annexes.

To link financial concepts with quantities, the format includes numbers next to the quantities that correspond to specific financial concepts. Consequently, we filter out tables that do not adhere to this pattern and only insert those that comply.

As illustrated in Figure 4.7, each line in the table comprises four columns: the financial concept in text, a four-digit reference number, the current value with its corresponding financial code, and the previous value with its corresponding financial code. For some companies, a lack of reported data for a specific financial concept may be represented as either zero or a blank space.


Registre de Commerce et des Sociétés
Numéro RCS : [redacted]
Référence de dépôt : [redacted]
Déposé et enregistré le [redacted]

RCSL Nr. : [redacted]	Matricule : [redacted]
	eCDF entry date : [redacted]

ABRIDGED BALANCE SHEET

Financial year from 01 [redacted] **to** 02 [redacted] (in 03 EUR)

[redacted]
[redacted]
L-2453 Luxembourg

ASSETS

		Reference(s)		Current year		Previous year
A. Subscribed capital unpaid	1101		101	0,00	102	0,00
I. Subscribed capital not called	1103		103	0,00	104	0,00
II. Subscribed capital called but unpaid	1105		105	0,00	106	0,00
B. Formation expenses	1107		107	0,00	108	0,00
C. Fixed assets	1109		109	8.000,00	110	5.576.000,00
I. Intangible assets	1111		111	0,00	112	0,00
II. Tangible assets	1125		125	0,00	126	0,00
III. Financial assets	1135	2.2.2, 3	135	8.000,00	136	576.000,00
D. Current assets	1151		151	6.340,68	152	217.500,92
I. Stocks	1153		153	0,00	154	0,00
II. Debtors	1163	2.2.3, 4	163	3.012,17	164	149.573,77
a) becoming due and payable within one year	1203		203	3.012,17	204	149.573,77
b) becoming due and payable after more than one year	1205		205	0,00	206	0,00
III. Investments	1189		189	0,00	190	0,00
IV. Cash at bank and in hand	1197		197	3.328,51	198	7.927,15
E. Prepayments	1199		199	0,00	200	0,00
TOTAL (ASSETS)			201	4.340,68	202	793.500,92

Figure 4.7: Example of Balance Sheet from LBR

4.6 Summary

In this chapter, we explain the four steps involved in Data Gathering and Refining. Depending on the data source, some steps may be skipped. For stable data sources like NACE codes and databases of Notaries, the data can be easily copied into a file and directly inserted into the database. Similarly, the database of Business Auditors can be handled in the same way but can also be periodically scraped to identify new or removed members before direct insertion into the database. For more complex sources, such as the LBR, several intermediate steps are necessary before data ingestion. These steps include downloading documents through provided links, extracting the text into intermediate files, and then inserting the data into the database.

Chapter 5

Information Enrichment

Contents

5.1 Introduction 72

5.2 Page-level Labeler 74

5.3 Financial Tokenizer 74

5.3.1 Generation of new terms 75

5.3.2 Embeddings training 76

5.4 Page type classifier 76

5.4.1 Long-text BERT Architecture 77

5.4.2 Page type classifier using Long-text BERT 83

5.5 Information Extraction 84

5.5.1 Dataset 85

5.5.2 Fine-tuning process 88

5.5.3 Model's Evaluation and model selection 91

5.6 Address Geolocation 95

5.7 Entity Resolution: Embedding Generation and Entity Insertion 96

5.7.1 Record Embeddings 96

5.7.2 Embeddings strategies 99

5.7.3 Dataset of Companies 100

5.7.4 Dataset of People 102

5.7.5 Testing and Results 103

5.8 Summary 106

5.1 Introduction

In this section, the primary DL and ML components used to enrich the existing data with new, valuable information are presented. First, a

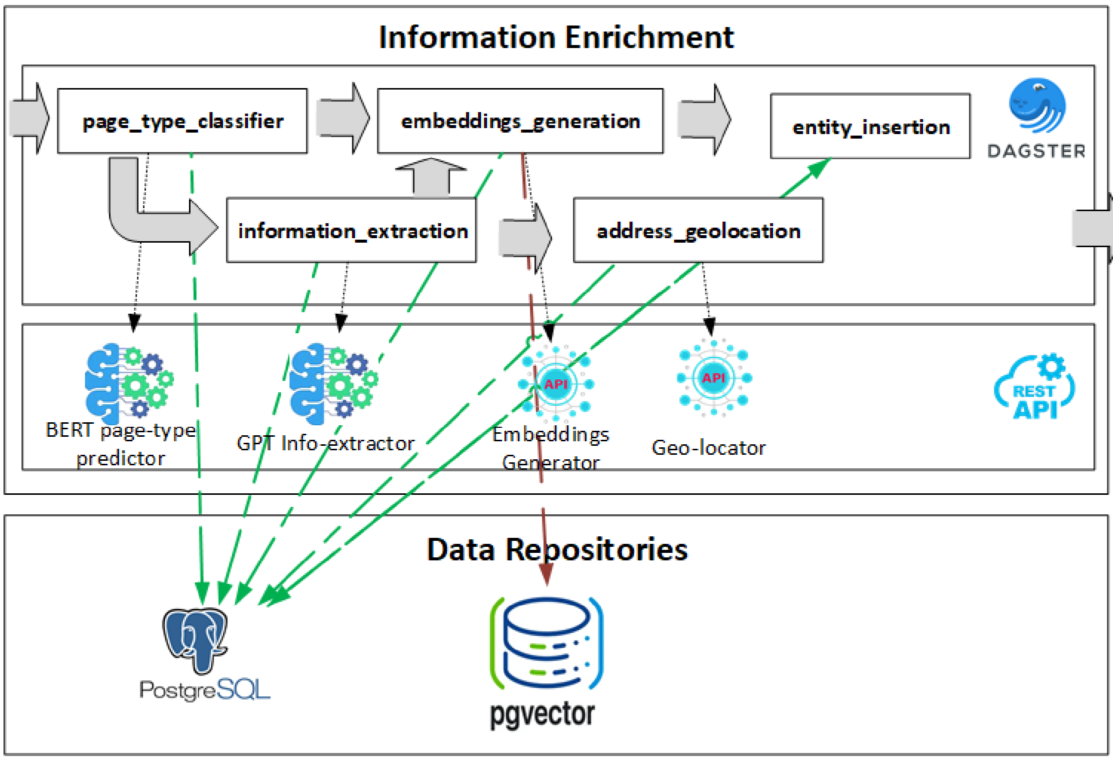
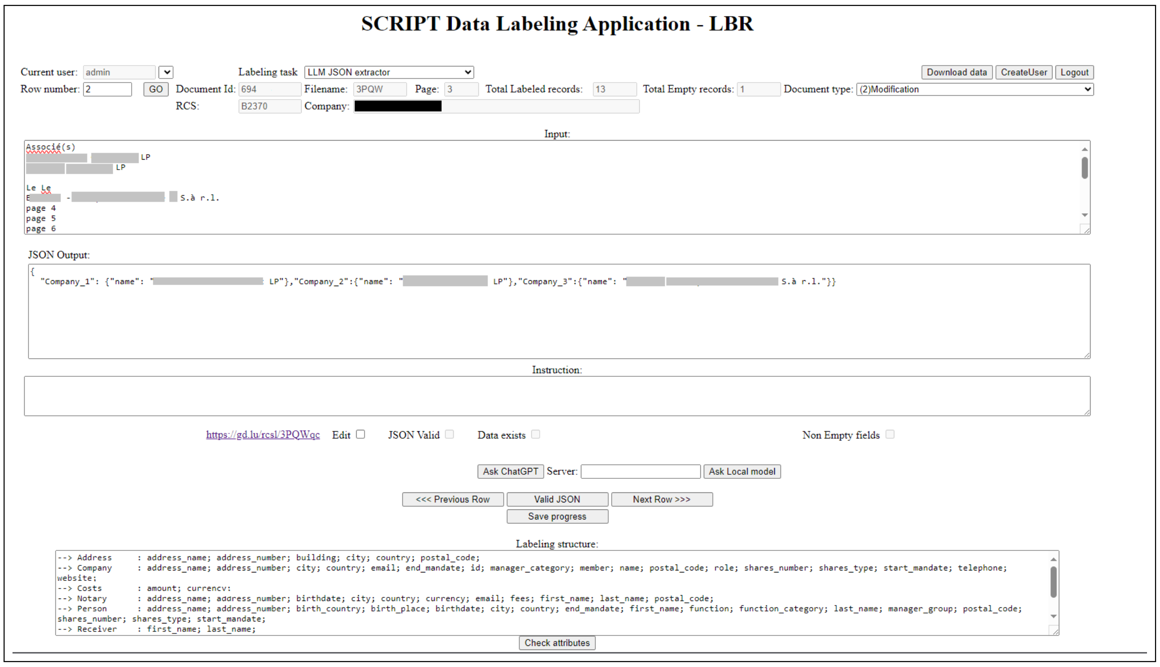


Figure 5.1: Steps for Information Enrichment stage. In this stage, the collected data is further enhanced using several processes. New data is generated through categorization models, including a page-type classifier, structured data extraction from a GPT-based information extractor, and the generation of embeddings for a vector database. Additionally, by utilizing an existing geolocation tool and the addresses in the database, latitude and longitude coordinates are obtained.

custom Financial BERT Tokenizer (T-MuFin) is developed to enhance the BERT-based models with additional data from the current BERT model. Subsequently, two architectures are proposed to handle longer text sequences, which involve the parallel operation of BERT instances to perform categorization tasks. Three models are employed at this stage. The first is a BERT-based model that analyzes the text of each document to classify the type of information, ensuring it is properly processed for the next step. The second and third models are GPT-based, fine-tuned with the dataset. After extracting the structured information, an external geo-locator API is utilized, and the enriched data is then inserted into the relational database management system (RDBMS) and graph databases. This process is illustrated in Figure 5.1.

5.2 Page-level Labeler

For page-level labeling, a web application was developed to iterate over text documents from the LBR (Figure 5.2). This tool allows users to view the extracted text in plain format, review metadata about the document and company, and select the document type. The JSON output textbox displays the labeled text, structured in JSON format in the example provided. Below this is the Instruction textbox, where users can input a prompt for a local LLM model or for API calls to ChatGPT 3.5 or Claude 3. The documents clickable link enables users to review the original document to ensure accurate manual labeling. Additional important features of the user interface include navigation buttons, JSON validation (if needed), and a labeling structure that updates as the labeling process progresses.


SCRIPT Data Labeling Application - LBR

Current user: admin
Labeling task: LLM JSON extractor
Row number: 2
Document Id: 694
Filename: 3PQW
Page: 3
Total Labeled records: 13
Total Empty records: 1
Document type: (2)Modification
RCS: B2370
Company: [redacted]

Input:

```
Associé(s)
[redacted] LP
[redacted] LP

Le Le
E[redacted] - [redacted] S.à r.l.
page 4
page 5
page 6
```

JSON Output:

```
{
  "Company_1": {"name": "[redacted] LP"},"Company_2":{"name": "[redacted] LP"},"Company_3":{"name": "[redacted] S.à r.l."}}
```

Instruction:

https://gd.lu/rcsl/3PQWqc Edit ☐ JSON Valid ☐ Data exists ☐ Non Empty fields ☐

Server:

Labeling structure:

```
--> Address     : address_name; address_number; building; city; country; postal_code;
--> Company     : address_name; address_number; city; country; email; end_mandate; id; manager_category; member; name; postal_code; role; shares_number; shares_type; start_mandate; telephone;
website;
--> Costs       : amount; currency;
--> Notary      : address_name; address_number; birthdate; city; country; currency; email; fees; first_name; last_name; postal_code;
--> Person      : address_name; address_number; birth_country; birth_place; birthdate; city; country; end_mandate; first_name; function; function_category; last_name; manager_group; postal_code;
shares_number; shares_type; start_mandate;
--> Receiver    : first_name; last_name;
```

Figure 5.2: Information Enrichment: Screenshot of the Web Application: Page-level Labeler

5.3 Financial Tokenizer

For the proposed T-MuFin (Term-based Multilingual Financial) Tokenizer, the initial step to increase the information capacity within a BERT model involves reducing the fertility of the BERT-base tokenizer. This process is carried out by adding new terms to the existing dictionary and subsequently training the associated weights using a Masked Language Modeling (MLM) task. The result is a newly proposed model, comprising the updated dictionary and the corresponding weights. The base tokenizer used for this process is the multilingual uncased BERT[1].

5.3.1 Generation of new terms

A method is proposed for generating new terms, beginning with the selection of candidate terms. To focus on a domain-specific use case, multi-word terms are prioritized for inclusion in the dictionary. For instance, the phrase *balance sheet* would be added as a single entry, while ensuring that the individual terms *balance* and *sheet* are also included if absent. To generate these terms, n-grams are employed with the value of n empirically set to 5, which is found to cover most financial terminology.

In the next step, the terms are cleaned using regular expressions, where enumerators and noise characters are removed. Dates and numbers are replaced with the special tokens “DATE” and “NUMBER”. Apostrophes are substituted with blank spaces. Once cleaned, the terms are sorted by frequency, and the top τ terms are selected as the “base list”.

Other terms are also added to the “base list”. Manually labeling *Balance Sheets* and *Profit and Loss Statement* of each language and getting the tree dependency of the financial terms such as “Assets” -> “Fixed Assets” -> “Intangible assets”. The resulting new list is the “extended base list”. Additionally, a parallel list of the hierarchical dependency is kept.

Once the “extended base list” is obtained, a step of term decomposition is performed. Each multi-word term is associated with its component terms at a lower level. For example, the term “shared premium account” is linked to “shared premium” and “premium account”, while “premium account” is further linked to “premium” and “account”. Each decomposed term is added to the “extended base list”. The hierarchical dependencies between terms, including their parent-child relationships and sibling terms, are recorded in the list of hierarchical dependencies.

[1]https://huggingface.co/google-bert/bert-base-multilingual-uncased

5.3.2 Embeddings training

Once all new terms have been defined in the final "extended base list", a set of training records is created using a MLM task. From the text of the Annexes of the Annual Accounts, each sentence is extracted. If the sentence contains a newly defined term, the κ preceding and following words, referred to as the context size, are also retrieved. Each term, along with its context, is added to the training records. Additionally, pairs of terms from the hierarchical dependency list, in both directions, are incorporated into the training records list.

For the MLM training, 20% of the tokens are replaced with the term "MASK". The dataset is then divided into training (70%) and testing (30%) sets. The 2nd to the 12th BERT encoders are frozen, and various hyperparameters are tested as detailed in the study by Blanco Lambruschini et al. [2023a]. Subsequently, the Masked Language Modeling training and evaluation are performed.

The best performance of the fine-tuning of the model was reached with a context size $\kappa = 5$, freezing 10 layers, a learning rate of 1e-5, and a dropout of 10%.

Table 5.1 presents the results of testing the BERT model with the MLM task, comparing it to the version using the new dictionary. It is evident that the F1 Score improves to 98%, the fertility is reduced from 1.25 to 0.89, and the percentage of truncated samples decreases from 0.79% to 0.2%.

Table 5.1: Performance of BERT Baseline

Model	**F1 Score Training**	**F1 Score Testing**	**Fertility**	**% Truncated samples**
Base	95.1%	94.9%	1.259	0.79
Best fine-tuned	98.2%	98.8%	0.891	0.20

5.4 Page type classifier

This Deep Learning model analyzes the text from a single page to determine its category. For this categorization task, a BERT-based model is considered the most effective, as explained in Section 2.6. One primary limitation of BERT-based models is the restricted number of tokens they can process. To address this issue, a novel architecture for handling longer texts in categorizations is presented in Blanco Lambruschini and Brorsson [2024b] and detailed in Section 5.4.1. The specific application of the best model for this research is then discussed in Section 5.4.2.

5.4.1 Long-text BERT Architecture

Table 5.2 was taken partially from the paper. Considering that the base model is a multilingual BERT model, the average number of tokens per word varies per each language. As we can see, English words has the lowest tokenization ratio, that means that almost one token is generated per word. On the other hand, in average, German generates 1.5 tokens per word, that means that a BERT model can be fed with less than 512 words. The table shows that even if the average number of tokens per page is less than 512, which means that it can be easily be fed into a single BERT instance, there are documents that can need more, specially in french documents, which requires at least three BERT segments.

Table 5.2: Long-BERT: Tokenization: Distribution per language. Taken from: Blanco Lambruschini and Brorsson [2024b]

Language	*avg* **t/w**	*max* **w/p**	*max* **t/p**	*avg* **t/p**
French	1.21	1,002	1,252	330
German	1.53	498	740	325
English	1.05	901	950	417

To handle longer texts, the paper introduces an architecture composed of three phases, as shown in Figure 5.3. The first phase involves pre-processing, where the input text is divided into smaller segments, each of which can be processed by an individual BERT instance. In the second phase, each segment is analyzed in parallel using BERT. Finally, the third phase integrates the results of these parallel analyses into a unified output. For the final phase, three architectural approaches have been designed and tested. Each phase is detailed below.

5.4.1.1 Pre-processing

This phase involves three key steps, as illustrated in Figure 5.4. First, the input text undergoes cleaning using regular expressions to remove unnecessary elements such as multiple blank spaces, dates, special characters, enumerators, and emails. Next, the cleaned text is tokenized using a BERT multilingual tokenizer. As previously mentioned, the

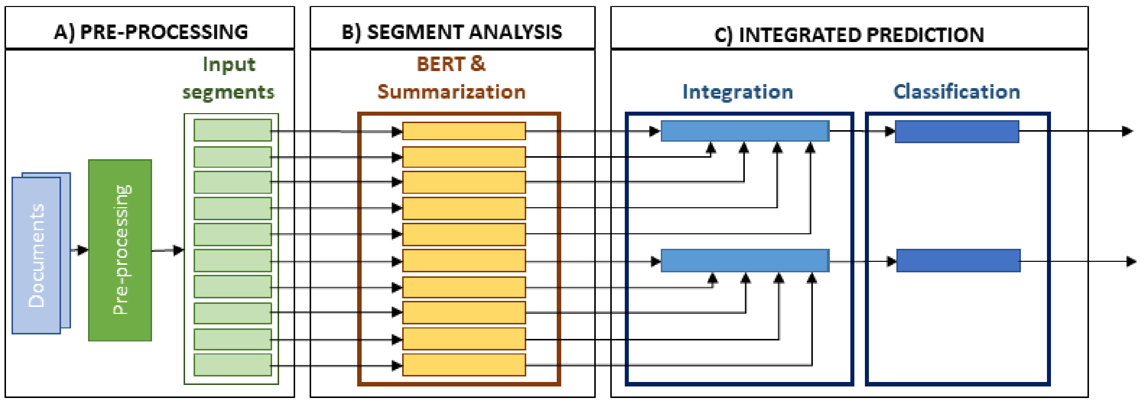


Figure 5.3: General Architecture: The proposed architecture has three phases. (1) Pre-processing is in charge of data cleaning, tokenization and divide the input text into segments able to fit into parallel BERT instances. (2) Segment analysis takes the parallel segments and fed them into the BERT instances, their corresponding outputs are then fed into the next phase. (3) Integrated prediction takes the output of the previous layer and integrates the parallel segments back into one instance for a single integrated prediction. Image taken from: Blanco Lambruschini and Brorsson [2024b]

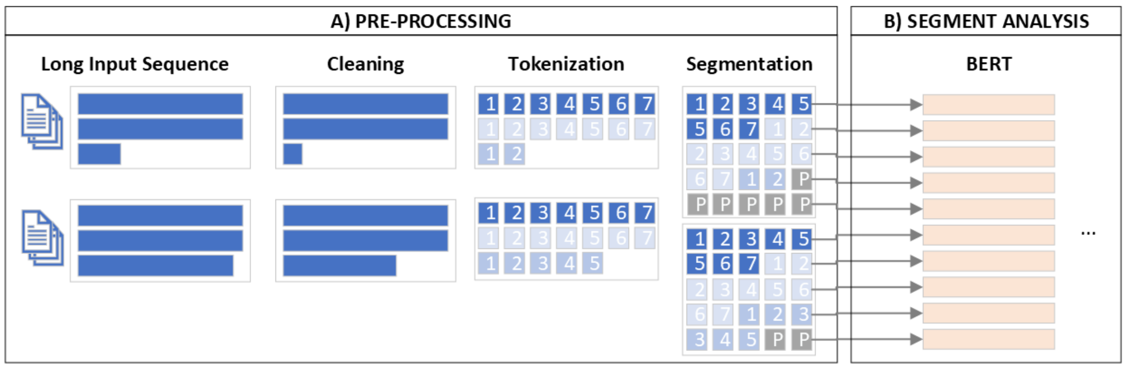


Figure 5.4: Pre-processing: In this phase, the entire text is first cleaned using regular expressions. Next, the cleaned text is tokenized using the BERT tokenizer. Finally, the text is divided into segments of up to 510 tokens, utilizing a sliding window to preserve context between segments. Image taken from: Blanco Lambruschini and Brorsson [2024b]

number of tokens generated per word varies depending on the tokenizer, its vocabulary, and the languages it has been trained on. For this particular tokenizer, which supports 110 languages, the vocabulary size is 110,000 tokens.

After tokenization, the text is divided into segments of 510 tokens or fewer. To minimize loss of context between segments, a sliding window approach is employed, where a portion of the last δ tokens from one segment is included at the beginning of the next segment.

Equation 5.1 provides the formula for calculating the number of segments per document (κ), based on the total number of tokens generated per document (τ) and the maximum number of tokens that can be input into BERT ($BERT_{input} = 510$).

$$\approx \kappa_{doc_i} = \lceil (\tau_{doc_i} - \delta)/(BERT_{input} - \delta) \rceil \tag{5.1}$$

Once the maximum number of segments is calculated, or the number that covers most of the text, the value of κ is set. If the number of tokens in a segment is less than the required 510 tokens, the special token *PAD* is used to fill the gap. Conversely, if there are more tokens than can fit into the κ segments, the excess text is trimmed from the beginning or end.

The final output of this phase consists of κ parallel segments, each containing 512 tokens. These input segments will be fed into the BERT model in the next phase.

5.4.1.2 Segment Analysis

This phase consists of two main steps. The first step involves using BERT to predict the output for each segment. The second step is an optional summarization, where the number of dimensions produced by BERT is reduced. This process is illustrated in Figure 5.5.

Each segment is processed independently by a single BERT instance in parallel. The output of BERT is a matrix with dimensions 512x768, where each 768-dimensional vector corresponds to an output token. The first vector, representing the *[CLS]* token, summarizes the entire input segment. The paper proposes two approaches: in the first, BERT's output is directly connected to the next phase; in the second, BERT's output serves as the input for a summarization network. This network reduces dimensionality through two consecutive dense layers, with the second layer being smaller than the first.

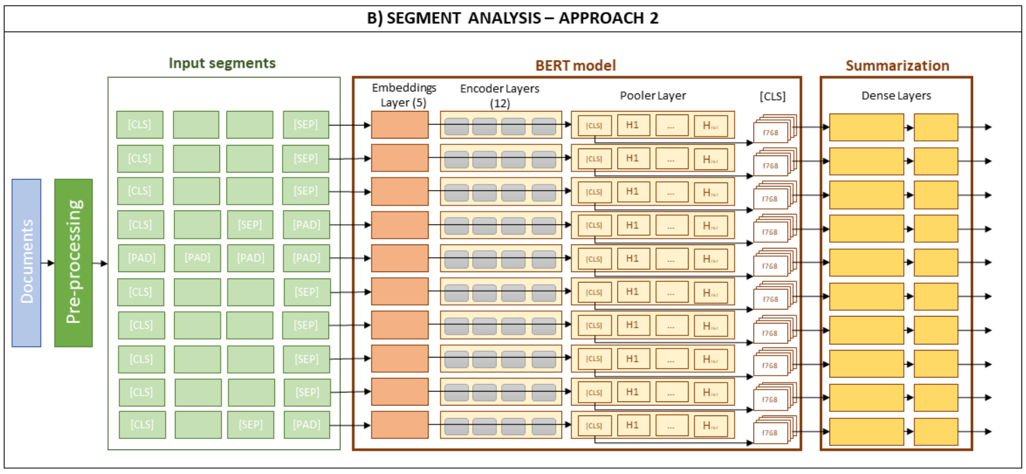


Figure 5.5: Segment Analysis: In this phase, the parallel segments are processed by multiple BERT instances. Each segment can be processed either without a summarization layer (approach 1) or with one (approach 2). Only the second approach is shown. Image taken from: Blanco Lambruschini and Brorsson [2024b]

5.4.1.3 Integrated prediction

In this final phase, the results from the segment analysis, which are still divided and processed in parallel, are integrated into a single sequential network. Three approaches are suggested for this integration. As illustrated in Figure 5.6, the first approach involves combining the parallel segments using an LSTM network, followed by connecting this output to a final dense classification layer. This method results in a drastic dimensionality reduction, transitioning from the high-dimensional LSTM output to a more compact classification layer.

In the second approach, an intermediate summarization layer is added between the output of the LSTM layer and the final classification layer. This intermediate layer helps mitigate the impact of the drastic dimensionality reduction. This approach is illustrated in Figure 5.7.

The final approach introduces a significant change by replacing the LSTM network with a concatenation layer, as shown in Figure 5.8. This allows the parallel outputs to be combined and fed into a sequential dense layer. The resulting neurons are then connected to an intermediate summarization layer, followed by a final classification layer, similar to the previous approach.

In general, the proposed architecture divides the text into segments that are processed in parallel by BERT instances and then integrated back for providing a single integrated prediction. The paper tests six

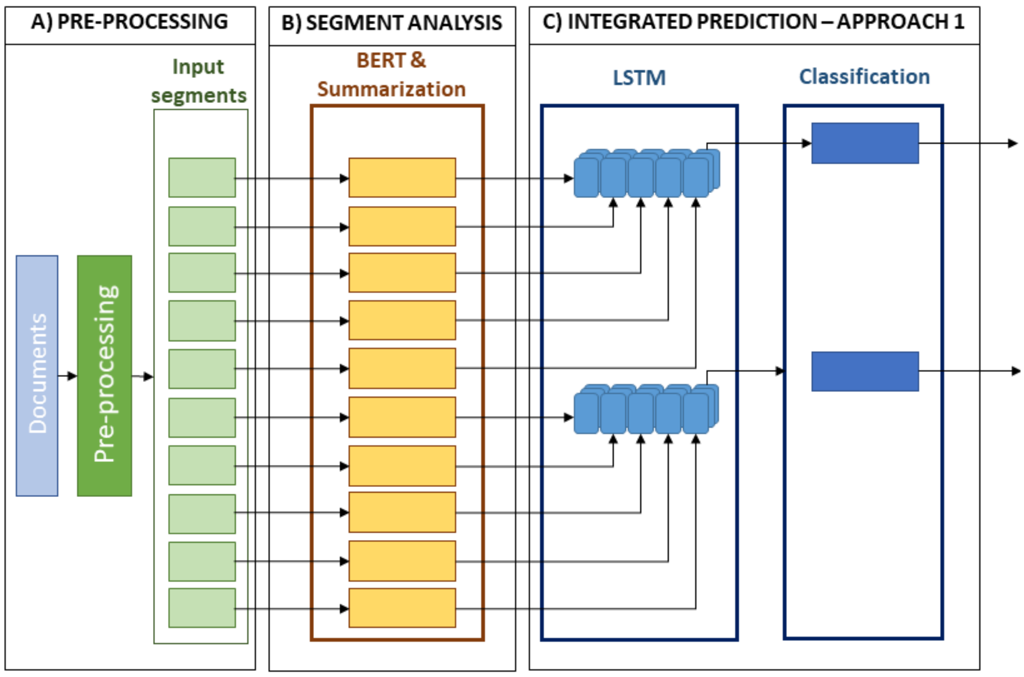


Figure 5.6: Integrated Prediction: First Approach. In this approach, each segment from the previous phase is fed into an LSTM layer, which processes the entire input as a single instance. A classification layer is then added to generate the final prediction. Image taken from: Blanco Lambruschini and Brorsson [2024b]

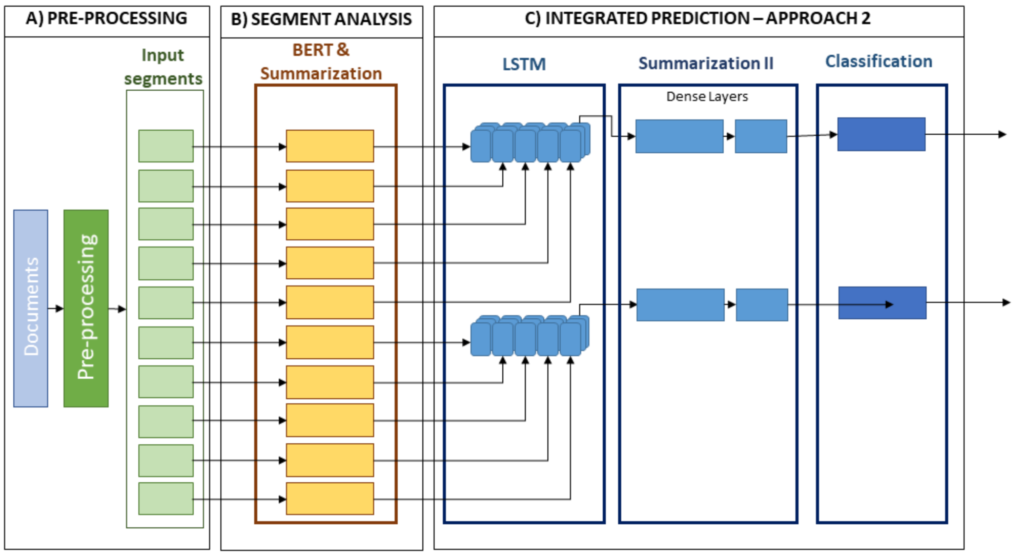


Figure 5.7: Integrated Prediction: Second Approach. In this approach, each segment from the previous phase is fed into an LSTM layer, which processes the entire input as a single instance. An intermediate summarization layer then takes the lengthy LSTM outputs and processes them through two dense layers with reduced dimensionality. Finally, a classification layer is added to generate the final prediction. Image taken from: Blanco Lambruschini and Brorsson [2024b]

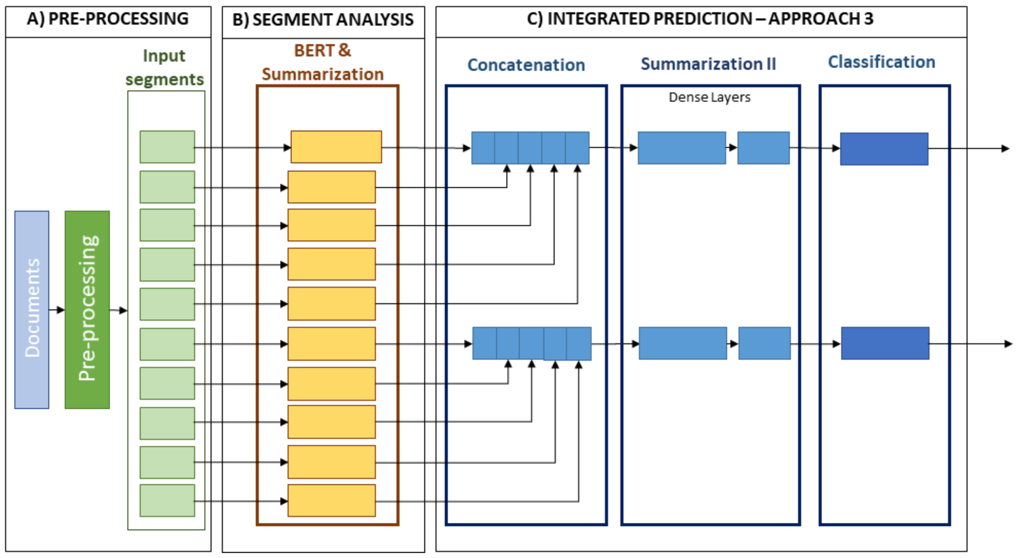


Figure 5.8: Integrated Prediction: Third approach. In this approach, each segment from the previous phase is fed into a dense layer, where the segments are concatenated, allowing the entire input to be processed as a single instance. An intermediate summarization layer then takes the concatenated output and processes it through two dense layers with reduced dimensionality. Finally, a classification layer is added to generate the final prediction.
Taken from: Blanco Lambruschini and Brorsson [2024b]

different test configurations based on various combinations of the proposed architecture and approaches as shown in Figure 5.9.

From the Figure 5.9, each model requires to define the size of the dense layer. These values are defined in Table 5.3.

Table 5.3: Value of parameters for the models.

Models	Parameter	Value	Analysis
2,4	λ_1	512	$2^{\lfloor \log_2 768 \rfloor}$
2,4	λ_2	256	Half of the previous dense layer
3,4,5,6	λ_3	512	$2^{\lceil \log_2(2*max(\phi)) \rceil}$, having $max(\phi) = 200$
3,4	λ_4	256	Half of the previous dense layer
5	λ_3	variable	$2^{(\lfloor \log_2(768*\kappa) \rfloor - 1)}$
6	λ_3	variable	$2^{(\lfloor \log_2(256*\kappa) \rfloor - 1)}$
5,6	λ_4	variable	Half of the previous dense layer
5	λ_5	variable	$768*\kappa/2$, half of the concatenated sequence
6	λ_5	variable	$256*\kappa/2$, half of the concatenated sequence

These architectures were evaluated on two tasks using two distinct datasets. Based on the results of these tests with the proposed configurations, the recommendation is to use the m4 model for shorter text sequences. This architecture consists of a BERT model for processing

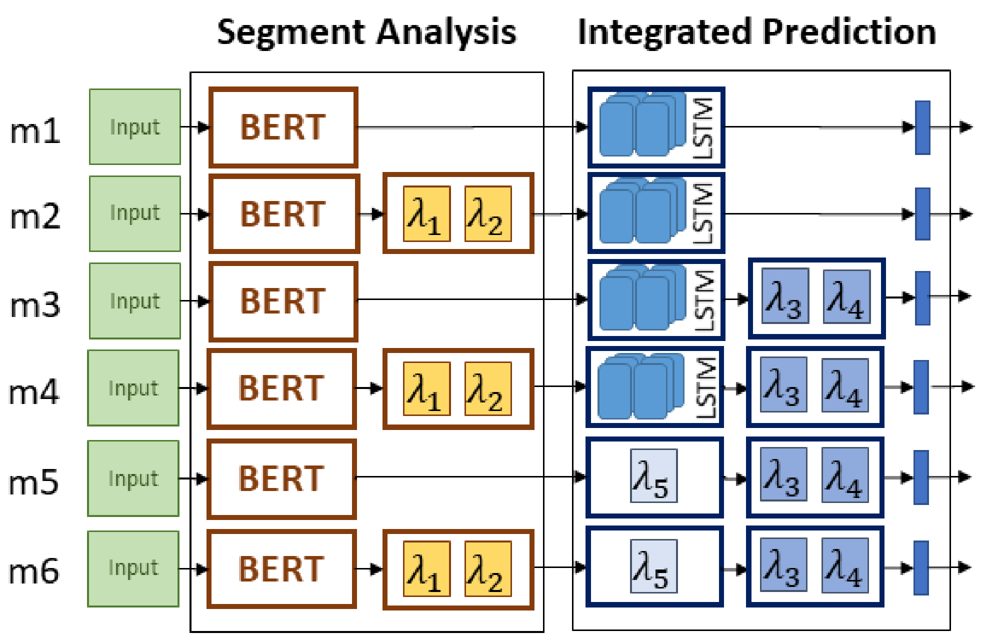


Figure 5.9: Test Configurations. The proposed six configurations are based on the combination of the previous approaches for each phase. The m1 model is the current base mode of the state of the art and the remaining are the proposed combinations. The lambda (λ) variables
Taken from: Blanco Lambruschini and Brorsson [2024b]

parallel segments, followed by two summarization layers, which are then integrated into an LSTM network. This is followed by two additional summarization layers, concluding with a final classification layer. On the other hand, if longer text sequences are required, the most suitable architecture is the m6 model, which replaces the LSTM layer in the m4 model with a concatenation layer.

5.4.2 Page type classifier using Long-text BERT

Due to the high variability in the formats of the documents within the dataset, it is necessary to predict the type of content based on the text of each page. Since this task involves categorization of moderately sized text that exceeds the maximum capacity of a BERT model, the m4 model from the proposed architectures was selected for this prediction task. As shown in Figure 5.10 this is the best performed model. This categorization enables the appropriate models to be applied based on the specific information found on each page.

For this reason, each page of the *Modifications* documents was eval-

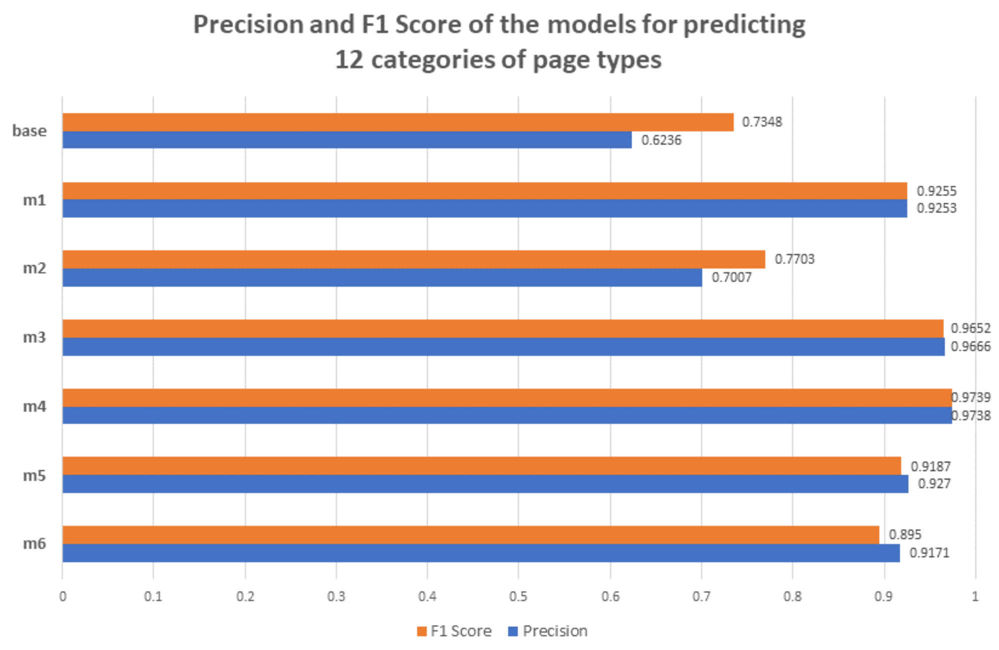


Figure 5.10: Results for not long sequences of texts. The best-performing models are m4 and m3, though m5 and m6 also shows strong results and is not far behind. Among the test configurations, m4 stands out as the most complex model. Image taken from: Blanco Lambruschini and Brorsson [2024b]

uated and categorized into one of 12 categories, including *annex*, *person information*, *summary general*, *shareholders*, *people summary*, *business info*, *fusion*, *continuation*, *liquidator*, *blank*, *branch information*, *foreign company*.

The prediction was carried out using $\kappa = 3$, with the last 100 tokens ($\delta = 100$) copied to the next sequence, and truncation applied at the end if necessary.

5.5 Information Extraction

The information extraction component is a DL model based on a generative model, specifically using GPT. The base model for this component is LLaMA 3.1 - 8B[2], which has been fine-tuned to process various company report pages. This model generates a JavaScript Object Notation (JSON) object containing sensitive information extracted from the provided content. The sensitive information includes details about individuals, companies, and their relationships.

[2]https://ai.meta.com/blog/meta-llama-3-1/

5.5.1 Dataset

For fine-tuning, the model utilized 1,000 manually labeled pages from documents sourced from the LBR. In addition to language, the document types were also considered to show the basic distribution of languages across different document types, as presented in Table 5.4. The dataset primarily consists of French (≈ 68%), English (≈ 32%), and German (≈ 14%) pages. Most of the pages pertain to modification-related documents and registration forms.

Table 5.4: Distribution of fine-tuning pages per type of document and language.

Document Type	**Fr**	**Ge**	**En**	**Total**
Modification	373	10	73	456
Registration	243	11	59	313
Non-statutory modification of the agents	0	119	0	119
Articles of association	66	2	44	112
Total	682	142	176	1,000

After running each page through the fine-tuned model described in section 5.4, the following 12 categories were identified: (1) annex, (2) person information, (3) summary general, (4) shareholders, (5) people summary, (6) business information, (7) fusion, (8) continuation, (9) liquidator, (10) blank, (11) branch information, and (12) foreign company. For statistical analysis, these categories were grouped into five broader groups, as shown in Table 5.5.

Table 5.5: Grouped Labeled Page Type Categories.

Group	**Categories**
Free Layout (FL)	(1)annex, (6)business information
Semi-structured (SS)	(2)person information, (4)shareholders, (7)fusion
	(9)liquidator, (12)foreign company
Incomplete Information (II)	(8)continuation
Branch Information (BI)	(11)branch information
Others (OT)	(3)summary general, (5)people summary, (10)blank

The Free-Layout (FL) group refers to text without a fixed format, such as annexes. The Semi-Structured (SS) group consists of structured pages that do not follow a uniform template and may include some free-layout text. The Branch Information (BI) group contains both structured and free-form information, typically describing the company or its branches. The Incomplete Information (II) group includes content from the SS or BI categories that was too lengthy to fit on the previous page and could not be easily categorized within the existing context. Finally, the Others (OT) group encompasses page summaries and blank pages.

Based on these categories, the statistics of the labeled data are presented in Table 5.6. The primary group in the dataset is the “Free Layout” group, followed by “Semi-Structured Information”.

Table 5.6: Distribution of labeled pages per type of content.

Content type	**# Pages**	**Avg. # words**
Business information (BI)	43 (4.3%)	129
Semi-structured information (SS)	216 (21.6%)	165
Incomplete information (II)	35 (3.5%)	134
Free-layout page (FL)	536 (53.6%)	310
Others (OT)	170 (17.0%)	205
Total	1,000 (100%)	259

Below is an example of the manually labeled JSON is shown in Listing 5.1 using the tool presented in section 5.2. From the example, we can see that there is one company where the document belongs, and another associated company, where we can see the name, id, address information and the relation between this company and the previous one which is *contrôle des comptes* or audit firm. Finally we can see the notary first name, last name and city.

```json
 {
   "Company_1":{
    "id":"B184831",
    "name":"British SYG Limited"
    },
   "Company_2":{
        "id":"B77840",
        "name":"Cayman Management S.à r.l.",
        "address":"Route d'Arlon",
        "number":"8",
        "postal_code":"1140",
        "city":"Luxembourg",
        "role":"contrôle des comptes",
        "start_mandate":"01/10/2015"
    },
    "Notary": {
        "last_name": "CASTELLONI",
        "first_name": "Javier",
        "city":"Bertrange"
    }
}
```

Listing 5.1: Example of Manually labeled JSON for Information Extraction

The labeling process does not modify the text, even in the presence of OCR errors, and preserves the original case. This approach helps to minimize hallucinations or false corrections that a GPT-based model might introduce.

After the labeled dataset is created, the main extracted entities and their corresponding attributes are evaluated. Examples of the most frequent attributes and entities are shown in Table 5.7.

Table 5.7: Most frequent entities and attributes from the labeled dataset.

Entity	**Most frequent Attributes**
Company	name, id, address_name, address_number, postal_code, city, role, country, email,start_mandate, end_mandate; shares_number, etc
Person	first_name, last_name, birthdate, birth_country, function, etc. birth_place,address_name, address_number, postal_code, city, country, start_mandate, end_mandate, function; shares_number, etc
Notary	first_name, last_name, email, city, country, etc.
Address	address_name, address_number, postal_code, city, country.
Transaction	amount, type, currency.
Others	Costs, Receiver, Signatures, etc.

The generation of synthetic data involved obtaining a list of values for each frequent attribute of each common entity using ChatGPT[3]. For example, this included requesting lists of 1,000 people's first names or last names. The main challenge was that, with multiple iterations, the names often repeated, resulting in limited growth of the list of unique values. To deal with this problem, we regionalize most of the queries.

For example, for *Person's*: *first name* the initial prompt was:

> "Act as a Synthetic dataset generator and generate a random list of 1,000 Person's first name"

and the regionalized prompt was:

> "Act as a Synthetic dataset generator and generate a random list of 200 Person's first names from *East Asia*."

The regions created were based on similarity of names. The following 24 regions were defined: (1) North America, (2) Latin America, (3)

[3]https://chat.openai.com

Caribbean and Central America, (4) Australia and Oceania, (5) Polynesia, Micronesia and Melanesia, (6) Nordic countries, (7) Central Europe, (8) UK and North Europe, (9) Germany and Central Europe, (10) France and Benelux, (11) Spain and Portugal, (12) Eastern Europe, (13) Italy and Mediterranean countries, (14) Arabic countries, (15) India and south Asia, (16) Israel and middle East, (17) Russia and Central Asia, (18) China, Japan, Korean and Eastern Asia, (19) Thailand, Cambodia, Vietnam and South Eastern Asia, (20) Northern Africa, (21) Southern Africa, (22) Middle Africa, (23) Western Africa, and (24) Egypt and Northern Africa.

After we got the different list of values for most of the attributes of the following entities: Person, Company, Address, Countries and Cities. Other attributes like dates and numbers were randomly assigned. Other attributes with lower diversity were manually created such as currencies, roles, functions.

After generating lists of values for the majority of attributes associated with the entities Person, Company, Address, Country, and City, additional attributes such as dates and numbers were randomly assigned. Attributes with low cardinality, such as currencies, roles, and functions, were manually created.

Replacing sensitive or personal information with synthetic data helps mitigate the risk of data leakage in the models.

5.5.2 Fine-tuning process

Figure 5.11 shows the fine-tuning process for the Information Extraction using a local LLM model. The proposed fine-tuning process includes as the first step the manual labeling the desired JSON output based of sensitive information based on the text of the page. As explained in Table 5.4, there were four kind of documents manually labeled (Modification, Registration, Non-statutory modification of the agents, Articles of association). Based on the manual labelled records and the synthetic data, three kind of datasets were created. The first and the third dataset contains only manually labeled and only synthetic records respectively, the second one is a mixture of both datasets.

The creation of datasets with synthetic data includes randomly selecting manually labeled data and replacing the real data with synthetic data. This is referred to as a semi-synthetic dataset because it contains real text from the labeled data, while sensitive information is replaced with synthetic data.

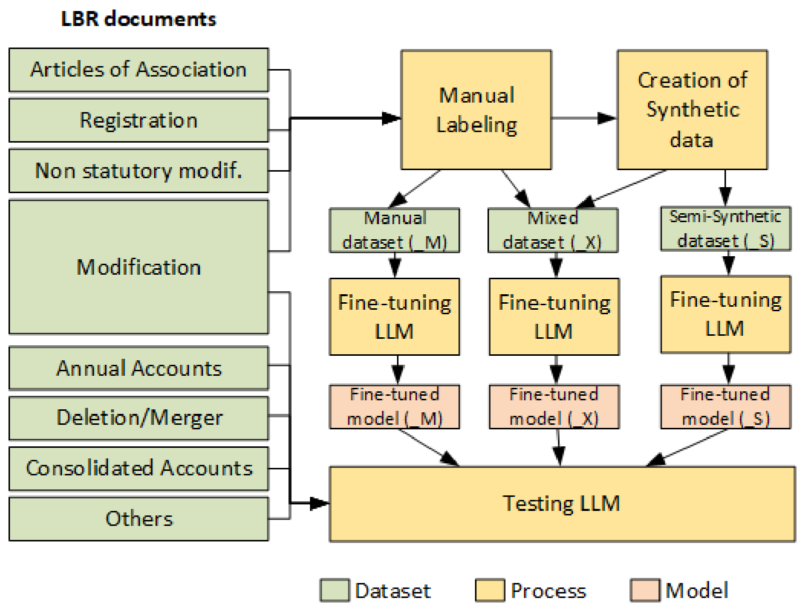


Figure 5.11: The fine-tuning and testing process begins with the manual labeling of various document formats. Next, synthetic data is generated using a large language model (LLM) and is used to replace sensitive information, creating a semi-synthetic dataset. Finally, the resulting model is fine-tuned and tested. Image taken from: Blanco Lambruschini and Brorsson [2024a]

The total number of records in the Manual Labeled dataset (_M) is 1,000. For fine-tuning purposes, three dataset sizes were created: 500, 750, and 1,000 records. Additionally, for the other two datasets (_S and _X), five sizes were used: 500, 750, 1,000, 2,000, 3,000, and 4,000 records (Table 5.8). Given that the maximum number of manually labeled records is 1,000, datasets with fewer records have a 50% ratio of real to synthetic data. For datasets with 3,000 and 4,000 records, they include 1,000 real records and the remaining as synthetic data.

Each dataset was fine tuned with LLaMA 3.1-8B and using quantization QLORA of 4 bits and PEFT. Having nf4 quantization type, 64 as LoRA dimension, 2e-4 as learning rate, 1e-3 as weight decay and optimizer AdamW.

LLaMA 3.1 8-b was selected because of its high performance without fine-tuning with respect to Mistral 7B [4]. The comparison between

[4]https://mistral.ai/news/announcing-mistral-7b/

Table 5.8: Dataset configurations for fine-tuning.

code	**Manual (_M)**	**Synthetic (_S)**	**Mixed (_X)**
_500	500	500	500
_750	750	750	750
_1000	1,000	1,000	1,000
_2000	-	2,000	2,000
_3000	-	3,000	3,000
_4000	-	4,000	4,000

these models and other free models are specified in the research paper Blanco Lambruschini and Brorsson [2024a].

To evaluate the performance of each model, five metrics were defined:

1. % Validity of JSONs (v): To ensure the model generates pure JSON responses, the percentage of outputs that are valid JSON without any extraneous information or remarks is measured. A higher percentage reflects improved adherence to the desired format. If required, valid JSON can be extracted by trimming any content before the first "{" and after the last "}"; however, this adjustment does not affect the metric, even if such a correction is feasible. The symbol $\uparrow$ indicates that higher values are more desirable.

2. % Existing-Data (ϵ): When the model's response is either valid JSON or can be corrected by removing extraneous text surrounding the JSON structure, an automatic verification is conducted to ensure that the attribute values for each entity match those found in the input text. A higher percentage signifies a lower occurrence of hallucination. The symbol $\uparrow$ indicates that higher values are preferable.

3. % Emptiness(ϕ): As before, for extractable JSON responses, the number of empty attribute values in the model's output is measured. Fewer empty values indicate greater model flexibility and adaptability across various scenarios. The symbol $\downarrow$ indicates that lower values are preferable.

4. % Repetitions(r): Additionally, for extractable JSON responses, the number of repeated attribute names and/or values in the model's output is measured. A lower percentage of repetitions reflects a

higher quality response. The symbol ↓ indicates that lower values are preferable.

5. % 2-Level Structure(s): This metric targets JSON structures that are simple and straightforward. Each JSON response should consist of a main object with its corresponding attributes, avoiding any complex nested objects within the primary structure. The required format is exemplified in the prompt.

The metrics ϕ and r, which evaluate the quality of the output, can typically be improved with a simple cleaning process. Although v can also be refined, this is not always feasible, as even a minor extraneous charactersuch as a comma, curly bracket, colon, or semicoloncan hinder information extraction. In contrast, ϵ and s focus more on the quality of the returned data, aiming to minimize hallucinations and assess the model's ability to adhere to the desired structure. Since ϕ and r are common issues even in top-performing models like GPT-4[5], Claude Haiku [6], and other subscription-based LLM services, the expectation is that our model will improve the remaining metrics while maintaining low values for these common issues.

Based on these metrics, it was concluded that the best-performing models were primarily those based on LLaMA3.1. The top model was found to be the one using only half of the manually labeled data, rather than the full dataset. However, if the privacy of the training data is of utmost importance, a Mixed Model (X_500) with the same number of records could be a suitable alternative.

5.5.3 Model's Evaluation and model selection

Figure 5.11 illustrates that a different dataset from LBR is used for testing the LLM models. This approach enables the assessment of the fine-tuning's impact across various document types. The testing dataset, detailed in Table 5.9, consists of 854 pages from 60 documents. The documents are evenly distributed among languages, with 20 documents for each language.

In addition to testing the LBR, the best-performing model is evaluated using financial documents randomly selected from the internet,

[5]https://openai.com/index/hello-gpt-4o/
[6]https://www.anthropic.com/claude

Table 5.9: Distribution of testing pages per type of document.

Document Type	**# Pages**
Consolidated Accounts (CA)	401
Modification (MD)	252
Annual Accounts (AA)	142
Deletion / Merger (DM)	20
Others (OT)	39
Total	854

covering various press releases and business records from companies in Spain and Sweden and in their respective languages as shown in Table 5.10.

Table 5.10: Distribution of testing pages per language and document types.

Language	**Press Releases**	**Business Records**	**# Pages**
Spanish	3	10	13
Swedish	9	5	14
Total	12	15	27

To test each model, FastAPI [7] is used to deploy the LLM models. Below is the generic prompt used for extracting the sensitive information.

<|begin_of_text|><|start_header_id|>system<|end_header_id|>Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.<|eot_id|><|start_header_id|>user<|end_header_id|>You are a helpful assistant designed to output only a short JSON. Do not add information which is not in the provided text. Please don't share false information. DO NOT HALLUCINATE.### Input: $$INSTRUCTION_TEXT$$ <|eot_id|> <|start_header_id|>assistant<|end_header_id|><|eot_id|>{

Each call, according to the page type, is generating a custom prompt that is sent to the API to be replaced in $$INSTRUCTION_TEXT$$. The custom prompt replaces the Entity detail such as Person and its attributes with the corresponding page's entity.

Extract sensitive information from multiple entities in a single JSON object (one level), if there is no information available do not include the field at all. Extract information such as: Person (first_name, last_name, birthdate, birth_country, birth_place, city, country, address_name, address_number). From the following text: $$PAGE_TEXT$$.

All test cases are presented in detail in Blanco Lambruschini and

[7] https://fastapi.tiangolo.com/

Brorsson [2024a]. As extracted from the paper, Table 5.11 shows the results for each test case model using LLaMA 3.1 as the base model. The results demonstrate that, in terms of output quality (v, ϕ, r), our fine-tuned models surpass the leading commercial models. Regarding the quality of the generated information (ϵ), both the manually labeled model (_M) and the mixed model (_X) achieve performance comparable to the best commercial model. Additionally, our fine-tuned models outperform others in the required structural quality (s).

Table 5.11: Performance of baseline models and fine-tuned models

Model	$v\uparrow$	$\epsilon\uparrow$	$\phi\downarrow$	$r\downarrow$	$s\uparrow$
GPT-3.5	**89.3**	**79.5**	14.9	4.3	**71.7**
Claude3	48.9	46.4	**0.0**	**3.6**	40.4
LLaMA:8B	0.0	30.9	61.6	1.1	11.9
LLaMA3.1_M500	100	**78.1**	0.0	6.7	100
LLaMA3.1_M750	100	64.4	0.0	**2.2**	100
LLaMA3.1_M1000	100	72.9	0.0	3.7	100
LLaMA3.1_S500	100	46.6	0.0	1.1	100
LLaMA3.1_S750	100	34.3	0.0	1.4	100
LLaMA3.1_S1000	100	**66.3**	0.0	1.7	100
LLaMA3.1_S2000	100	44.0	0.0	1.3	100
LLaMA3.1_S3000	100	53.4	0.0	1.3	100
LLaMA3.1_S4000	100	27.6	0.0	**0.9**	100
LLaMA3.1_X500	100	**71.8**	0.0	2.6	100
LLaMA3.1_X750	100	67.0	0.0	2.4	100
LLaMA3.1_X1000	100	51.8	0.0	2.0	100
LLaMA3.1_X2000	100	52.5	0.0	**1.8**	100
LLaMA3.1_X3000	100	32.4	0.0	1.7	100
LLaMA3.1_X4000	100	45.5	0.0	1.8	100

Table 5.12 presents the performance of the baseline and fine-tuned models from a different perspective, categorized by content type. This analysis underscores the model's versatility in processing various content types. Consistent with the previous results, the LLaMA3.1-based models outperform even the best-performing commercial model.

Table 5.13 displays the performance of the models on previously unseen document types. The models perform better on Manual datasets, though the difference in performance compared to Semi-Synthetic and Mixed datasets is not significant. It is important to note that these documents primarily consist of numerical tables, where the models often adjust formatting by adding semicolons as thousand separators or inserting zeros in the decimal places.

Table 5.12: Baseline model's performance results by type of content: ϵ & s

Model	BI (%)		FL (%)		II (%)		SS (%)	
	$\epsilon\uparrow$	$s\uparrow$	$\epsilon\uparrow$	$s\uparrow$	$\epsilon\uparrow$	$s\uparrow$	$\epsilon\uparrow$	$s\uparrow$
GPT-3.5	**100**	**100**	**79**	**70**	**50**	**50**	**95**	**97**
Claude3	79	78	44	38	50	50	89	89
LLaMA3.1:8B	60	40	30	10	**50**	**50**	56	**39**
LLaMA3.1_M500	**100**	**100**	**77**	**100**	50	**100**	**97**	**100**
LLaMA3.1_M750	100	100	62	100	50	100	94	100
LLaMA3.1_M1000	100	100	71	100	**100**	100	95	100
LLaMA3.1_S500	0	100	46	100	0	100	64	100
LLaMA3.1_S750	100	100	31	100	50	100	88	100
LLaMA3.1_S1000	**100**	**100**	**64**	**100**	**100**	**100**	89	**100**
LLaMA3.1_S2000	50	100	42	100	50	100	86	100
LLaMA3.1_S3000	83	100	51	100	50	100	**97**	100
LLaMA3.1_S4000	83	100	24	100	50	100	89	100
LLaMA3.1_X500	**100**	**100**	**70**	**100**	50	**100**	**97**	**100**
LLaMA3.1_X750	100	100	65	100	100	100	93	100
LLaMA3.1_X1000	83	100	49	100	100	100	96	100
LLaMA3.1_X2000	100	100	50	100	100	100	89	100
LLaMA3.1_X3000	83	100	30	100	100	100	70	100
LLaMA3.1_X4000	83	100	43	100	50	100	86	100

Table 5.13: Baseline models Performance by document type: ϵ & s

Model	AA (%)		CA (%)		DM (%)	
	$\epsilon\uparrow$	$s\uparrow$	$\epsilon\uparrow$	$s\uparrow$	$\epsilon\uparrow$	$s\uparrow$
GPT-3.5	**88**	**86**	**78**	**73**	**64**	**64**
Claude3	67	60	42	38	18	18
LLaMA3:8B	**55**	**50**	**35**	**25**	45	20
Mistral_M500	66	67	58	58	63	64
Mistral_M750	76	76	62	62	62	64
Mistral_M1000	**82**	**83**	**66**	**67**	**63**	**64**
Mistral_S500	59	60	45	46	55	55
Mistral_S750	**65**	**67**	45	48	**58**	**64**
Mistral_S1000	56	56	**51**	**53**	51	55
Mistral_S2000	16	16	8	9	22	22
Mistral_S3000	31	31	20	20	26	27
Mistral_S4000	16	17	18	20	9	9
Mistral_X500	**64**	**64**	**62**	**64**	27	27
Mistral_X750	57	58	54	57	53	55
Mistral_X1000	59	60	44	47	**54**	**55**
Mistral_X2000	8	9	26	25	0	0
Mistral_X3000	17	17	17	18	9	9
Mistral_X4000	21	21	12	12	18	18

Based on the previous results, the *Llama3.1_S1000* model was selected for this project due to its ability to minimize the risk of sensitive data leakage while maintaining performance comparable to leading commercial models. As shown in Table 5.14, the model's performance across different languages further confirms its versatility.

Table 5.14: Performance of the best performed semi-synthetic-based model with other languages and formats

Model	$v\uparrow$	$\epsilon\uparrow$	$\phi\downarrow$	$r\downarrow$	$s\uparrow$
LLaMA3.1_S1000	100	81	0	4	100

5.6 Address Geolocation

The geocoding is a process where a text address is converted into geographic coordinates such as latitude/longitude pairs.

There are existing libraries and API services that allow to convert textual addresses into this coordinates. Some of the most popular python libraries are Nominatim, Positionstack and OpenCage[8].

In the dataset, the company information includes the registration address, which, in many cases for SMEs, coincides with the commercial address. This text address is used to obtain the geographic coordinates. This research used the Google Geocoding API [9], which demonstrated with some random tests, being the most accurate with respect to Google Maps.

We have only geocoded addresses with at least the following information: number, address name and city. In total we got 17,350 geocoded locations from the metadata from the LBR dataset and addresses extracted from the documents. Is important to mention that many addresses could refer to the exact same location because of the apartment number or slightly variations to the street name or city, which is solved in the graph using a distance comparison.

[8]https://www.kaggle.com/code/ahmedshahriarsakib
[9]https://developers.google.com/maps/documentation/geocoding/overview

5.7 Entity Resolution: Embedding Generation and Entity Insertion

As explained before, one of the most difficult task after gathering data from different datasources, is to integrate them into the same entities if those have enough similar information to consider them as the same entity. As a input of this stage, we have the extracted information as JSON objects in two levels, at entity at attribute level. For example:

```
{
   'Company': {
        'name': 'Yoscnir S.à r.l.',
        'id':'XG6UO'
   },
   'Person':{
       'first_name':'SUMIN',
       'last_name':'Karlst',
       'city':'Hämeenlinna',
       'country':'Finland'
   },
   'Notary': {
       'last_name': 'CASTELLO',
       'first_name': 'Marie',
       'city':'Belvaux'
    }
}
```

For the purpose of this research, only two main entity types are considered: Companies (e.g., Shareholders, Auditors, Liquidators) and People (e.g., Shareholders, Notaries, Auditors, Liquidators, Managers). Unlike previous studies, the company data is sourced from two distinct origins: (A) a baseline provided by the LBR and information extracted from additional sources. In contrast, the data for individuals is primarily derived from these other sources.

In this research, a proposition for ER is made to utilize a ground truth baseline of company and individual names. A unique identifier (ID) is created in the database, and information from other data sources is matched to determine the corresponding ID if it exists.

5.7.1 Record Embeddings

The foundation of the proposition is based on comparing the embeddings of records. This will be achieved by leveraging pre-existing embeddings, such as OpenAI's Text-embedding-3-small[10], Llama3.2-3B embeddings[11], accessed through Ollama[12], the best performed fine-tuned Llama3.1, and a proposed character-based tokenization method.

The size of the embedding vectors varies depending on the model used, as shown in the Table 5.15. The proposed embeddings is going to be called character-entity based.

Table 5.15: Embedding vector size according to the model

Embedding model	**Vector size**
OpenAI	1,536
Llama3.2	3,072
Llama3.1 Fine-tuned	4,096
character-entity based	392

The size of the proposed embedding vector is determined by the characteristics of the unstructured data. A tool has been developed to generate a class that transforms an entity into its corresponding vector representation.

5.7.1.1 Embeddings Generator Tool

The tool first processes the records at two hierarchical levels: the entity level and the attribute level. It then calculates the frequency of each entity and its attributes. A parameter ϵ is defined as the minimum frequency threshold required for an entity or attribute to be included in the model.

A JSON configuration file is automatically generated by the tool, listing entities and their attributes sorted by frequency. The configuration file also identifies key attributes, designated based on the presence of terms such as *"id"* or *"code"* in their names. If no such attributes are found, the most frequent attribute is selected as the key. Only the attributes included in this configuration file are used for embedding generation.

To accommodate specific business requirements, the configuration file can be modified as needed. Additionally, a supplementary JSON configuration file is provided, containing a list of entities along with empty

[10]https://openai.com/index/new-embedding-models-and-api-updates/
[11]https://www.llama.com/
[12]https://ollama.com/blog/embedding-models

vectors. This secondary file allows for the inclusion of new attributes initially excluded due to the frequency threshold. These additional attributes are not incorporated into the embedding vector but are used as filters to extract meaningful information, which can then be added to the baseline data.

The embedding vector is constructed as a concatenated series of individual vectors, each of size 28. Each position within these vectors corresponds to a specific character: the 26 letters of the English alphabet (az) and three special tokens ($: representing special characters, #: representing numerical characters, and &: representing spaces). Each individual vector represents either an attribute name or an attribute value, as illustrated in Figure 5.12.

Characters outside the predefined set, including accented or special forms (e.g., ç, æ, å, á, é, í, ó), are replaced with their corresponding standard equivalents (c, ae, a, a, e, i, o) to ensure consistency.

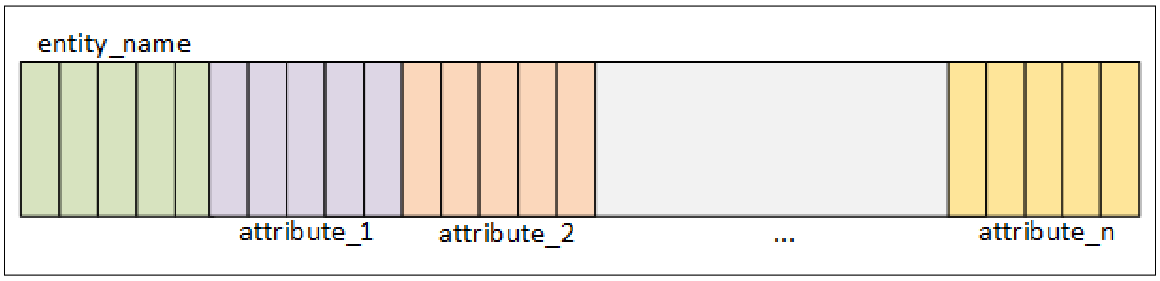


Figure 5.12: ER Embeddings

Examples of entity and attribute names include:

- **Company**: name, ID, postal code, city, address, etc.
- **Person**: first name, last name, birthdate, birth place, birth country, etc.
- **Costs**: amount, currency, type, etc.

The size of the embedding vector is calculated by multiplying the maximum number of attributes by the size of the individual vector, then adding one as shown in Equation 5.2

$$\text{Embedding vector size} = \text{vector\_size} \times (\text{max\_attributes} + 1) \quad (5.2)$$

For this dataset, the maximum number of attributes is 13. Consequently, the size of the embedding vector is 392.

To generate the embeddings for each entity, the following steps are performed (Figure 5.13):

- **A. Entity Cleaning**: The text of the entity name or attribute values is converted to lowercase. Special vowels and consonants are replaced using a dictionary of equivalences. Remaining special characters are replaced with the corresponding token ($), numbers with (#), and blank spaces with (&).
- **B. Weight Iteration**: For each entity name or attribute value, the occurrences of each character are counted to determine its weight. As a base weighting strategy, each occurrence is assigned a weight of 1.
- **C. Concatenation**: For each entity, individual vectors are concatenated in the order specified by the configuration file to form the final embedding vector.

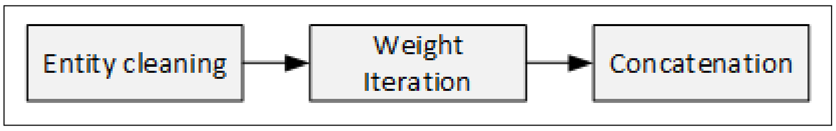


Figure 5.13: ER Embeddings Generation Process

5.7.2 Embeddings strategies

To evaluate different options for the proposed embeddings, five weight strategies will be tested in the *Weight Iteration* step:

- **Base**: No weight strategy is applied; all occurrences are assigned a weight of 1.
- **Sliding Window**: The first letter and every nth letter thereafter are assigned a weight of 2. Empirically, n was set to 3.
- **First Upper**: The first letter of each word is assigned a weight of 2.
- **First and Last**: Both the first and last letters of each word are assigned a weight of 2.

- **Weighted Vocabulary**: Using the dataset, a frequency counter was developed to rank the frequency of each character. The five most frequent letters were assigned a weight of one, with subsequent groups of five letters being assigned a weight incremented by one. Then the corresponding weight per character is assigned while creating the embedding.

5.7.3 Dataset of Companies

The proposed ER method relies on a ground truth against which unidentified records will be compared. In this case, the ground truth is the list of registered companies in Luxembourg obtained from the LBR. For this reason, each company record is going to generate its own embedding and saved into the vector database.

The ER process for companies is simpler than that for individuals, primarily because the company name can be used as the main attribute. However, the primary challenge lies in ensuring the cleanliness of company names across different data sources. In this case, the company name in the extracted information often includes the company type, leading to variations such as *Total Engine* being recorded as *Total Engine S.à r.l.*, *Total Engine Sàrl*, *Total Engine Sà rl.*, or other similar formats. Additionally, typos and OCR errors in the names are not addressed during this stage of processing.

For this purpose, a BERT-based model was developed to process company names by producing two outputs. The first output removes the company type from the name, while the second output identifies the company type ID, regardless of the language. This model was fine-tuned using manually labeled data and provides confidence scores for both outputs, Achieving an F1 score of approximately 90%, this model has been deployed as an API service. It facilitates batch data cleaning during the preprocessing phase, prior to embedding generation, specifically for the proposed models.

Figure 5.14 illustrates the two steps involved in the ER process. The first step focus on preparing the data for comparison within a real-world business context and generating the corresponding embeddings. The second step consists of three sub-steps, which align in certain aspects with previously proposed solutions.

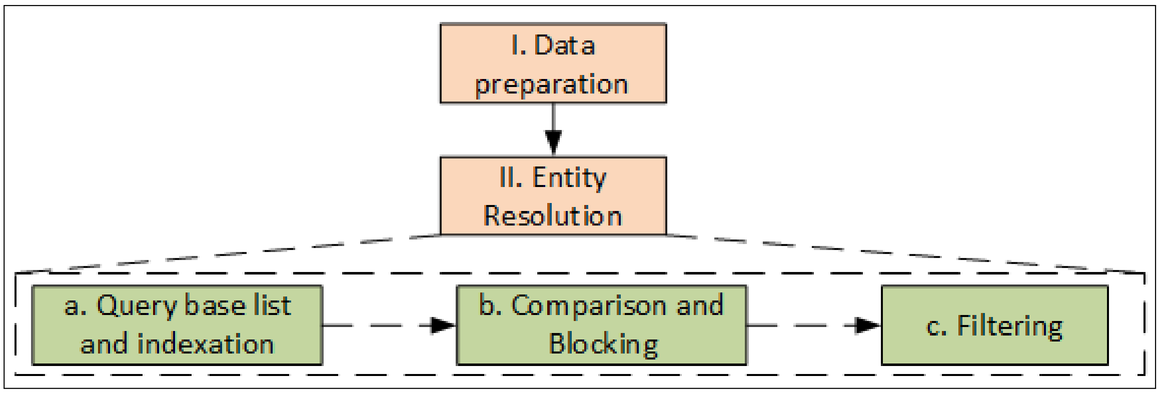


Figure 5.14: ER process for companies

- **I. Data preparation**: Information from various data sources is listed and integrated, generating a key based on the attributes specified in the JSON configuration file. When a new key is generated, the record is added to a vector. If the key already exists but has at least one different or additional attribute, the record is still added to the vector. The records grouped by key are consolidated at the document level, represented by an internal ID derived from metadata. Finally, an embedding vector is generated for each record.

- **II. Entity Resolution**: The proposed ER method consists of three steps, taking as input a list of records and their corresponding embeddings.

  – *a. Query Base List and Indexation*: The ground truth, which already includes precomputed embeddings, is queried, and the records are indexed according to a similarity measure. Three measures were tested: *cosine similarity*, *euclidean distance*, and *inner product*. Among these, *inner product* yielded the poorest results, while *cosine similarity* slightly outperformed *euclidean distance*. Therefore, *cosine similarity* was selected for subsequent testing. Faiss for Python was employed for this task, where the embedding vectors were normalized and indexed using `IndexFlatIP`.

  – *b. Comparison and Blocking*: Each record and its corresponding embedding were processed. For GPT-based models, two approaches were tested: (1) using the embeddings generated from the record in JSON string format, and (2) creating a natural language description of the JSON data. For example, a JSON object like {"Company":{"name":"Total Engine

Sàrl", "city":"Luxembourg"}} was converted into a textual representation such as: "The company name is Total Engine Sàrl and the city is Luxembourg". For the proposed embeddings, the API service was used to remove the company type from the name, perform text cleaning, and generate the embedding. The comparison is done using Cosine Similarity. Blocking was performed by selecting the top n most similar records.

  – c. *Filtering*: Levenshtein distance was applied to the records in the selected block, and a Levenshtein confidence score was calculated as $1 - \frac{\text{Levenshtein distance}}{\text{maximum number of characters in both strings}}$. The record with the highest Levenshtein confidence score was selected. If the confidence score was below 0.8, it was determined that no matching record existed in the database, and no selection was made.

5.7.4 Dataset of People

As previously explained, a ground truth is required to obtain the IDs. Since no publicly available database of individuals exists, a custom dataset will be created by defining the minimum attributes necessary to serve as a ground truth. Consequently, all information from other sources will be compiled, and only records containing more than the minimum required attributes will be included in the preliminary list. As a final step in this process, the list of individuals will be sorted by last name and the number of attributes. In cases where duplicate names are identified, the record with the most attributes will be retained.

The generation of the embeddings is the same explained for companies, The ER step change a bit because of the complexity of the entity.

As shown in Figure 5.15, the process for entity resolution with people records is similar to that for companies. However, it includes a loop between the blocking and filtering steps. The differences are detailed as follows:

As shown in Figure 5.15, the process for entity resolution with people records is similar to that for companies. However, it includes a loop between the blocking and filtering steps. The differences are detailed as follows:

- b. *Comparison and Blocking*: In the initial iteration, a mask is

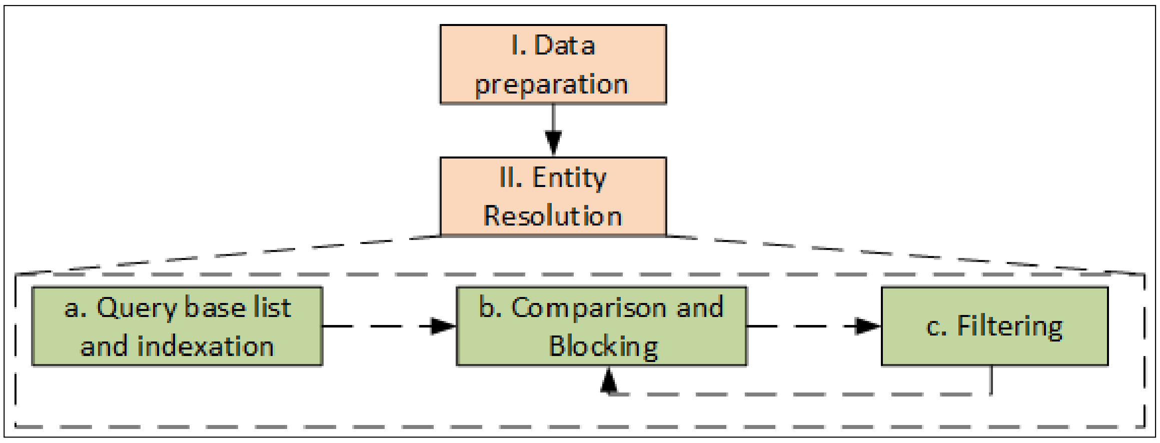


Figure 5.15: ER process for people

applied such that only the last name vector is compared; all remaining attribute vectors are set to zero in both the ground truth and the record being compared. A top n most similar embeddings are selected, with n empirically set to 50. For subsequent iterations, the block size remains unchanged, but the mask is applied to focus on the next attribute vector in the priority order defined in the configuration JSON file.

- *c. Filtering*: If the first name is represented by an initial only, records within the block are filtered to include only those matching the first letter. For other cases, filtering is conducted based on Levenshtein confidence. However, instead of selecting a record immediately, all records in the block are ranked by their confidence scores. This ranked list is carried forward to the next iteration, where the next attribute in the priority order is used for comparison. The process continues until no more attributes remain in the record for comparison. The final rank is calculated as the simple average of the ranks across all attributes. The record ranked highest (top 1) is returned as the result.

5.7.5 Testing and Results

To test the different approaches a set of extracted information from documents was executed in the eleven configurations as shown in Table 5.16.

After processing the set of documents, 350 records containing company information were randomly selected and manually verified to determine if the selected IDs were correct. Similarly, 200 records containing

Table 5.16: Test case Scenarios for Entity Resolution

Model Name	Description
base	Base: All character weights are 1.
sliding_window	Sliding Window: The first character and the nth letter of each word have a weight of 2, while the rest have a weight of 1.
first_upper	First Upper: Only the first character of each word has a weight of 2, while the rest have a weight of 1.
first_last_upper	First and Last: Only the first and last characters of each word have a weight of 2, while the rest have a weight of 1.
weighted_vocab	Weighted Vocabulary: Character weights depend on their infrequency in the dataset.
llama3	Llama 3.1-7B: Fine-tuned with JSON input.
text_llama3	Llama 3.1-7B: Fine-tuned with descriptive input.
ollama_llama3.2	Llama 3.2-3B: Fine-tuned with JSON input.
text_ollama_llama3.2	Llama 3.2-3B: Fine-tuned with descriptive input.
openai_3-small	OpenAI 3: Fine-tuned with JSON input.
openai_text_3-small	OpenAI 3: Fine-tuned with descriptive input.

personal information were manually reviewed to confirm the accuracy of the selected person IDs.

Table 5.17 presents the accuracy of the various executed models (both GPT-based and the proposed models) for the list of companies. It is evident that the proposed models outperform the others by a significant margin, followed by the fine-tuned Llama 3.1 model.

Table 5.17: Results ER for Companies

Model Name	Accuracy	Model Name	Accuracy
first_last_upper	0.806	llama3	0.210
base	0.792	ollama_llama3.2	0.006
weighted_vocab	0.789	openai_3-small	0.003
first_upper	0.781	openai_text_3-small	0.003
sliding_window	0.772	text_llama3	0.003
		text_ollama_llama3.2	0.003

Table 5.18 presents the accuracy of the various executed models (both GPT-based and the proposed models) for the list of people. As observed previously, the proposed models outperform the others by a significant margin, followed by the fine-tuned Llama 3.1 model. Unlike the results for the previous dataset, the base approach performs the worst for the people dataset, while the weighted vocabulary approach achieves better

results. In both cases, the first and last upper approach demonstrates the best performance.

Table 5.18: Results ER for People

Model Name	Accuracy	Model Name	Accuracy
first_last_upper	0.781	llama3	0.109
weighted_vocab	0.763	ollama_llama3.2	0.036
first_upper	0.727	openai_3-small	0.036
sliding_window	0.709	openai_text_3-small	0.036
base	0.672	text_llama3	0.018
		text_ollama_llama3.2	0.036

As shown in Figure 5.16, it is evident that the proposed methods for Entity Resolution outperform the GPT-based model, whose embeddings do not perform well with very short text sequences. The fine-tuned Llama 3.1 model achieves better results, likely because the model has already been trained to understand the JSON structure, entities, and their attributes. Notably, the *first and last upper* method demonstrates the best performance in both cases, achieving an accuracy of approximately 80% in both scenarios.

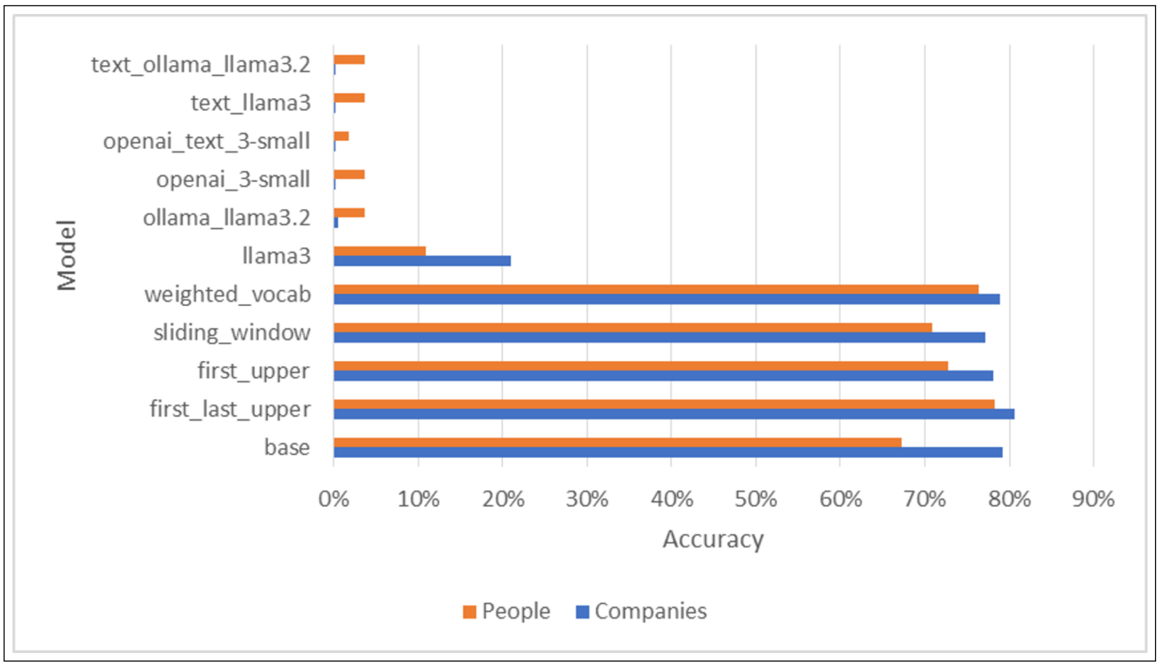


Figure 5.16: Results ER for companies and people datasets

5.8 Summary

In this chapter, existing information, along with Machine Learning and other tools, is used to generate additional, more structured data for easier manipulation. A new multiword-based Financial BERT Tokenizer (T-MuFin) is first proposed to manage more information within BERT instances. Following this, a BERT-based architecture is introduced, designed to process long text sequences for categorization tasks. For documents containing fewer than five pages, the architecture involves segmenting the document and processing each segment through parallel BERT instances. Two dense layers are added, and the parallel networks are combined using an LSTM, followed by two additional dense layers for summarization. For longer text sequences, it is recommended to replace the LSTM with a concatenation layer.

With this proposed model, a page type classifier model has been fine-tuned to predict the type of page based on its content for non-financial documents with an accuracy of approximately 97%. Additionally, a GPT-based model has been fine-tuned using one of the lightest and most efficient open-source models, LLaMA 3.1-8b, for information extraction. This model allows for the generation of a JSON output from an unstructured and free-layout form that can be easily processed later. An accuracy of around 78% for non-hallucination and 100% for JSON output structure has been achieved. Using an existing tool, all addresses in the database have been geocoded into latitude and longitude coordinates. The proposed Entity Resolution model achieves better performance than GPT-based models and ensures to have the best accuracy ratio, around 80%. Finally, a model for embedding entities has been proposed to integrate them into a single instance.

Chapter 6

Data Analysis and Insights Generator

Contents

6.1 Introduction . . . 107

6.2 Authors fingerprint autoclustering . . . 108

6.2.1 Autoclustering Algorithm . . . 109

6.3 Text-based bankruptcy prediction . . . 116

6.4 Companies' Cluster Analysis . . . 119

6.4.1 Data ingestion . . . 119

6.4.2 Node Graph Centrality . . . 126

6.5 Graph Risk Analysis: Multidimensional Risk Proposed Model . . . 127

6.5.1 Proposed dimensions . . . 128

6.5.2 Proposed sub-dimensions . . . 133

6.5.3 Evaluation of the Integrated proposed multidimensional Risk Model . . . 134

6.5.4 Multidimensional Risk Report . . . 138

6.6 Summary . . . 140

6.1 Introduction

This chapter introduces the application of Machine Learning and Deep Learning models to extract valuable insights from existing data. First, an autoclustering algorithm is employed to identify the "hidden accountant" by analyzing textual similarities within the annexes of the Annual Accounts. Additionally, a long-text BERT model is applied to predict the risk of bankruptcy based solely on textual information from these annexes. Finally, leveraging all the generated insights and a Graph Network, new dimensions are defined for the multidimensional risk model, leading to a unified risk assessment.

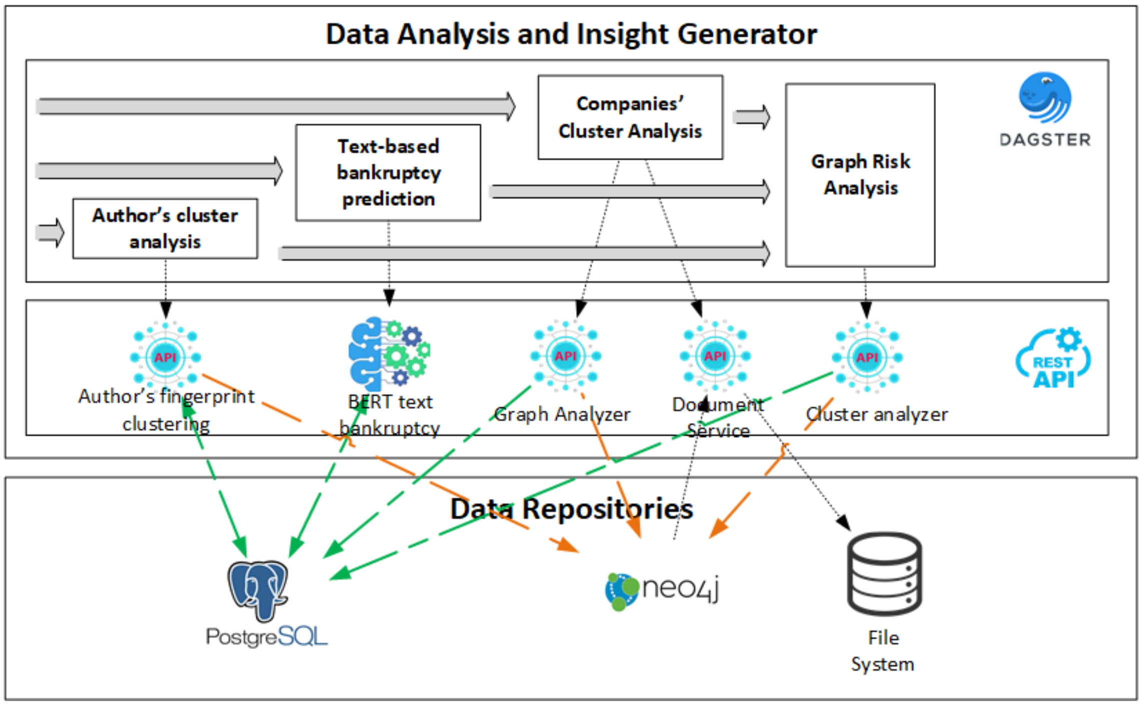

Data Analysis and Insight Generator
Companies' Cluster Analysis
Text-based bankruptcy prediction
Author's cluster analysis
Graph Risk Analysis
DAGSTER
Author's fingerprint clustering
BERT text bankruptcy
Graph Analyzer
Document Service
Cluster analyzer
REST API
Data Repositories
PostgreSQL
neo4j
File System

Figure 6.1: Steps for Data Analysis and Insight Generator stage. In this stage, predictive and clustering models generate key components for the risk analysis. A machine learning model clusters documents to identify the author's fingerprint, a graph-based clustering algorithm analyzes and groups companies based on distinct features. Finally two Machine Learning models predict bankruptcy risk and overall risk across various dimensions.

6.2 Authors fingerprint autoclustering

Financial reports often contain valuable insights that may not be easily discernible when analyzed individually, but can reveal significant information when examined in aggregate. Due to the nature of filing annual reports and other financial documents, patterns related to the authorship of these documents can be traced. While this analysis does not directly identify the author, it enables the identification of documents produced by the same individual or entity as presented in Blanco Lambruschini et al. [2023b].

In the case of SMEs, it is common for companies to rely on an accountant or accountancy firm whose have to prepare reports for multiple clients. These authors often use a standard template, modifying or adding specific details for each individual company. A similar pattern can be observed when extending this analysis to large corporations. These corporations either engage large consultancy firms that serve numerous clients or utilize a standardized format provided by the parent company, which is then shared across all subsidiaries and members of the corporate group.

The proposed model leverages this assumption to cluster documents based on their format and content. The algorithm follows a two-level autoclustering process. At the first level, documents are grouped based on similar formatting, while at the second level, they are clustered according to the author's distinctive fingerprint or writing style.

The proposed algorithm does not require the number of clusters to be specified in advance. Each document compares its features with those of existing clusters. If the features are similar, the document is included in the relevant cluster; otherwise, the document forms a new cluster.

The algorithm operates with three parameters: the minimum threshold for considering a document as part of a cluster (κ_c), the minimum threshold for merging clusters that appear similar (κ_m), and the minimum threshold for considering a document as part of a subcluster (κ_s).

Table 6.1: Threshold parameters used in the Autoclustering algorithm.

Threshold Parameter	**Notation**	**Explanation. Min similarity for:**
Clustering	κ_c	Appending a document into a cluster.
Merging	κ_m	Considering merging two similar clusters.
Sub-clustering	κ_s	Appending a document into a sub-cluster.

6.2.1 Autoclustering Algorithm

The algorithm is designed to process large volumes of documents, which is why the clustering process is performed in parallel, followed by the merging of clusters. As illustrated in Figure 6.2, the algorithm consists of four phases: (I) generation of cluster candidates, (II) feature cleaning, (III) cluster merging, and (IV) subclustering.

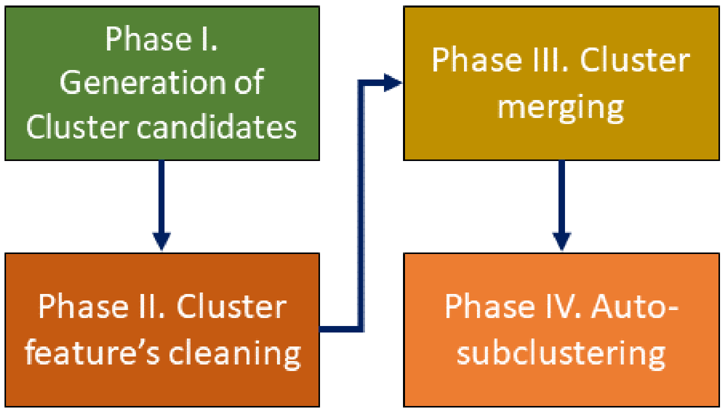

Phase I. Generation of Cluster candidates
Phase II. Cluster feature's cleaning
Phase III. Cluster merging
Phase IV. Auto-subclustering

Figure 6.2: Autoclustering algorithm's phases. The algorithm operates in four phases. First, a set of cluster candidates are created, where documents are assigned randomly to be processed in parallel. Then in each cluster, features are generated from both the text and metadata of the documents, then the autoclustering is applied. In the second phase, the less confident features of the clusters are removed. The third phase involves parallel cluster merging, with the number of workers gradually reduced in a loop until a single worker is left and maximum merging is achieved. Finally, in the last phase, the autoclustering algorithm is applied within each cluster to identify sub-clusters of authors.

6.2.1.1 Phase I: Generation of cluster candidates

As a preliminary step in this phase, the working documents are divided into Ψ parallel workers to accelerate the initial phase of feature extraction and clustering. Once the workload is distributed, features are extracted, and the documents are clustered based on these features within each worker's document set. As explained below.

The next step involves creating a set of features and ensuring they are comparable. Two types of features are proposed, derived from the text and metadata extracted from the PDF documents (as detailed in section 4.4): content-based and format-based features.

The algorithm uses the following format-based features:

- Enumerator patterns: Annual reports and annexes contain multiple sections and subsections, numbered with various patterns that mix letters and numbers, such as Arabic numerals, cardinal numbers, letters, parentheses, dots, and dashes. This is a key feature in these types of documents, as authors often use a specific, non-standard numbering style. To generalize this feature, the actual values are replaced with their types, allowing for frequency analysis. For example, a pattern like "V.a-2)" and "VI.b-1)" is translated to "roman dot lower-letter dash arabic parenthesis".

- Page orientation: The PDF text extraction tool provides page orientation (landscape or portrait). While most data is presented in portrait orientation, some authors use landscape for specific pages, such as tables, helping to cluster documents more effectively based on this distinction.
- Language: The language of the page, which in this case can be French, English, or German.
- Horizontal position of each line: Each text line's position is defined by the extraction tool's scale. The horizontal position helps indicate tabulation, which is important in identifying an author's template.
- Text line width and height: The width and height of text lines are also provided by the extraction tool. This feature helps identify margins and font sizes used by authors.

To avoid an excessive number of features resulting from special cases or tool errors, only features with a frequency of ≥ 2 are considered.

For content-based features, titles, subtitles, and first text lines are included as distinct features. These are identified using the following rules:

- Title: A short text line, either in all uppercase or with a horizontal position that exceeds 30% of the page width.
- Subtitle: A text line that begins with an enumerator pattern.
- Text-line: The first non-subtitle line that follows a subtitle.

Regular expressions are used to remove any numerators, dates, and numbers from titles and subtitles.

As illustrated in Figure 6.3, features such as language, page orientation, titles, and subtitles are utilized for first-level clustering (template-based), while position, text width, height, and text lines are used for sub-clustering (author's fingerprint).

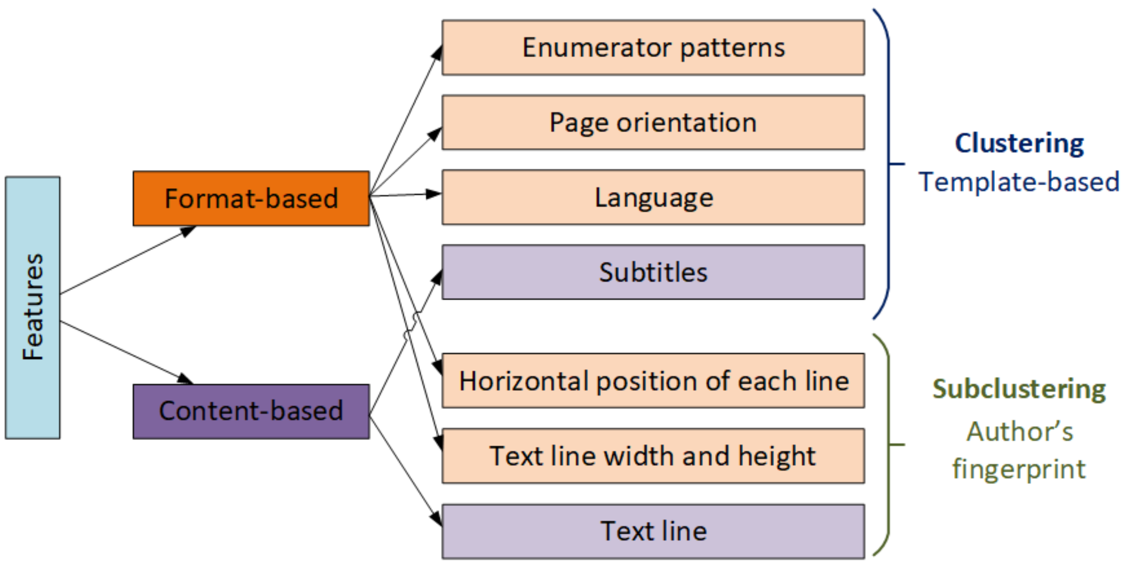


Figure 6.3: Autoclustering: List of Content- and Format-based Features. Four features are used for clustering at the template level, while three distinct features are applied for identifying the author's fingerprint at the second level.

The first stage of clustering attempts to assign a document to an existing cluster through feature comparison or create a new cluster if no suitable cluster exists. When a new cluster is created, all of the document's features are transferred to the cluster as its features. This process is repeated for each document.

In this phase only clustering first level features are used. Feature comparison involves comparing the cluster's features with the document's features, a process known as feature-set similarity (χ). If χ exceeds the clustering threshold (κ_c), the document is added to the cluster, and the cluster's features are merged with the document's features.

Feature merging involves adding any document features not already present in the cluster, assigning them a confidence score of 1 divided by the number of documents in the cluster. For features already present, only the confidence score is updated, representing the number of documents containing the feature relative to the total number of documents. The formula for calculating the confidence score (σ) is shown in Equation 6.1. The confidence score (σ) contributes to the overall similarity score (χ), allowing the algorithm to prioritize more frequent or common features, as outlined in Equation 6.2 and Equation 6.3.

$$\sigma_{(f_{t+1})} = (\sigma_{(f_t)} * n_t + 1)/n_{t+1} \tag{6.1}$$

$$\mu_{features}(A, B) = (|f_{(A)}| + |f_{(B)}|)/2 \tag{6.2}$$

$$\chi(A, B) = \sum_{i=0}^{|f_{(AUB)}|} \begin{cases} \sigma_i/\mu_{features} & , \exists f_i \varepsilon A \wedge f_i \varepsilon B \\ 0 & , \text{otherwise.} \end{cases} \tag{6.3}$$

Let $|f(K)|$ represent the number of features in cluster K, and $f_{(A\cup B)}$ be the set of features resulting from the union of features from A and B, with σ_i as the confidence score of the i^{th} feature in $f_{(A\cup B)}$.

The *feature-set similarity score* (χ) is based on the graph-probing distance, as shown in Equation 2.1, but incorporates the feature's confidence score (σ). This score (χ) is used in phase I for comparing the similarity between a document and a cluster, and in phase III for comparing two clusters. It measures the average contribution of each feature when the feature is present in both the cluster and the document (or another cluster). If a document contains most of the features in a cluster, especially those with higher confidence scores, it will be considered highly similar to the cluster.

After merging a document, the *cluster confidence* (τ_c) or sparsity of the documents within the cluster is calculated. τ measures the average confidence scores of the features, as shown in Equation 6.4. A higher τ indicates that most features have high confidence, suggesting that the documents in the cluster are highly similar to each other.

$$\tau = \sum_{i=0}^{n} (\sigma_{(f_i)})/n \tag{6.4}$$

6.2.1.2 Phase II: Cluster feature's cleaning

The second phase involves removing low-confidence features that are not representative of the cluster. These features tend to be special cases, and their removal creates more reliable clusters for the next phase. In this pruning phase, each cluster's feature confidence score $(\sigma_{(f)})$ is compared to the forgetting threshold (κ_f). If $\sigma_{(f)}$ is lower than κ_f, the feature is pruned, and the cluster confidence score (τ) is recalculated.

Pruning is applied only to clusters with more than four documents $(n_{docs} > 4)$. The value of κ_f was empirically determined, as shown in

Equation 6.5.

$$\kappa_f = \begin{cases} 1/n_{docs} & , n_{docs} < 10 \\ 2/n_{docs} & , \text{otherwise.} \end{cases} \tag{6.5}$$

6.2.1.3 Phase III: Cluster merging

In the first phase, documents were divided into several groups to accelerate processing. In the current phase, the resulting clusters are being collected, and two clusters are combined if they are found to be similar. It is expected that similar documents, which were placed in different working groups during the first phase, will now be grouped into the same cluster. The comparison is carried out in a manner similar to the first phase, but instead of individual documents, clusters are compared, with κ_m serving as the threshold parameter.

Then, the merging process is distributed among several k_m workers. As a result, all clusters (including those consisting of a single document) are divided into groups for parallel processing. Each group creates a similarity matrix, where each pair of clusters compares their feature sets.

To reduce the number of comparisons and improve the performance of constructing the similarity matrix M, only clusters with a similar number of features are compared. The comparison follows a process similar to that between a document and a cluster, but the resulting similarity score χ is stored in the matrix for later fusion. If *cluster* A is compared with *cluster* B, resulting in a similarity score χ_{AB}, then the matrix is updated such that $M[A, B] = M[B, A] = \chi_{AB}$.

Once the matrix is complete, the process proceeds by iterating through the rows. For each cluster, all of its high-confidence pairs are retrieved, and for each retrieved cluster, its high-confidence pairs $\tau_c \leq \kappa_m$ are also retrieved. This process is performed recursively, ensuring that previously included clusters are not reconsidered. The entire chain of clusters is then merged into a single cluster, combining their features and the cluster confidence τ_c.

$$\sigma_{A(f_{t+1})} = (\sigma_{A(f_t)} * n_{At} + \sigma_{B(f_t)} * n_{Bt})/(n_{At} + n_{Bt}) \tag{6.6}$$

To update the confidence score σ (Equation 6.6), Equation 6.1 is modified, where t represents the time prior to the merge, and A and B represent the clusters to be merged. The confidence score is adjusted by multiplying the existing confidence scores by the number of documents in each cluster, summing the results, and then dividing by the total number of documents after the merge.

This process is repeated until no further clusters remain to be merged.

6.2.1.4 Phase IV: Sub clustering

The final phase involves clustering within each individual cluster. The algorithm described in 6.2.1.1 is repeated for this sub-clustering process. In this stage, only second-level features are considered for comparison, and the sub-clustering threshold κ_s is applied.

The process is performed cluster by cluster, with no parallelization required due to the speed of the operation, eliminating the need to distribute the workload.

6.2.1.5 Results and Evaluation

The following results were obtained. Table 6.2 shows that there were 6,119 clusters, having 3,953 clusters (10.2%) with one single document (or unclustered documents). Also similarly 8.4% of clusters were discarded (because they have less than 5 features) and almost 70% of the clusters have a high confidence (τ_c bigger than 0.85).

Table 6.2: Auto-Clustering Results

Total clusters	One document	More than 2 docs		discarded	High confident clusters	
		clusters	documents		clusters	documents
6,119	3,953 10.2%	2,165	31,264 81.3%	3,238 8.4%	2,070	25,735 66.9%

Table 6.3 shows the evaluation of the clusters and subclusters with the Rand Indexes of 1,000 manually labeled documents. Having a Rand Index of 87% for first level or template based clustering. For the second level clustering, subclustering or author's fingerprint clustering was around 55%.

Table 6.3: Rand Index (RI) results

Metric	Clustering	Sub-Clustering
Rand Index	87%	55%

6.3 Text-based bankruptcy prediction

As explained in Section 2.2, most current state-of-the-art methods for bankruptcy prediction are based on financial ratios and financial statements. Some studies that incorporate text analysis also include an examination of the risk section in 10-K forms from publicly traded companies in the United States. For the majority of annual accounts not governed by SEC regulations, such as the dataset in this study, the analysis focuses on the information provided in the annexes of the annual accounts to predict bankruptcy risk.

Among the two leading technologies for NLP, a BERT-based model is better suited for this task, given the classification nature of the problem and the potential for high hallucination if GPT-based models were used. However, as previously mentioned, a limitation of BERT is its restricted ability to handle long information, capped at 512 tokens. For this reason, the bankruptcy prediction will be conducted using the best-performing model from the proposed methodology and architecture in Section 5.4.1.

As part of the multidimensional analysis, the annexes of the Annual Accounts were analyzed to detect risk alerts, excluding numerical information. The predicted categories were *Risky* and *No Risky*. On average, each document in this dataset contains eight pages and approximately 2,000 tokens. For this task, δ and κ were set to 100 and 3, respectively.

For fine-tuning the BERT models, the following hyperparameters were used: a dropout rate of 20%, a learning rate of $1e^{-5}$, the AdamW optimizer with a weight decay of $1e^{-2}$, a batch size of 5, and 4 epochs. To reduce the trainable model size, 10 out of 12 encoder layers were frozen.

Detailed results from Blanco Lambruschini and Brorsson [2024b] is shown in Table 6.4. The base model was a random selection of the whole text which fits into a BERT instance and is fed into a final classification layer.

Due to the highly imbalanced data, the focus is on precision for predicting risk based on the text, as shown in Figure 6.4. The best-performing model for bankruptcy prediction was the m4 model with 80 LSTM layers, achieving a risk prediction precision of over 70%.

Table 6.4: Results of Risk Precision of the models with respect to the different number of hidden layers (h).

Case	Accuracy	F1 score	Precision	Recall
base	0.810	0.772	**0.258**	0.951
$m_1\_h_{30}$	0.899	0.888	0.518	0.920
$m_1\_h_{50}$	0.899	0.888	0.520	0.916
$m_1\_h_{80}$	0.925	0.922	0.513	0.909
$m_1\_h_{120}$	0.890	0.875	0.464	0.922
$m_1\_h_{200}$	0.937	0.933	**0.643**	0.850
$m_2\_h_{30}$	0.811	0.729	0.014	0.941
$m_2\_h_{50}$	0.811	0.730	0.017	0.905
$m_2\_h_{80}$	0.820	0.751	0.065	0.949
$m_2\_h_{120}$	0.912	0.905	**0.608**	0.902
$m_2\_h_{200}$	0.919	0.914	**0.651**	0.898
$m_3\_h_{30}$	0.909	0.901	0.589	0.903
$m_3\_h_{50}$	0.915	0.908	**0.621**	0.903
$m_3\_h_{80}$	0.911	0.904	**0.617**	0.904
$m_3\_h_{120}$	0.919	0.914	0.526	0.911
$m_3\_h_{200}$	0.917	0.911	**0.641**	0.895
$m_4\_h_{30}$	0.907	0.899	0.584	0.896
$m_4\_h_{50}$	0.902	0.893	0.555	0.892
$m_4\_h_{80}$	**0.809**	0.723	**0.733**	0.813
$m_4\_h_{120}$	0.890	0.875	0.473	0.920
$m_4\_h_{200}$	0.872	0.850	0.523	0.903
m_5	0.914	0.907	**0.602**	0.904
m_6	0.893	0.880	0.486	0.918

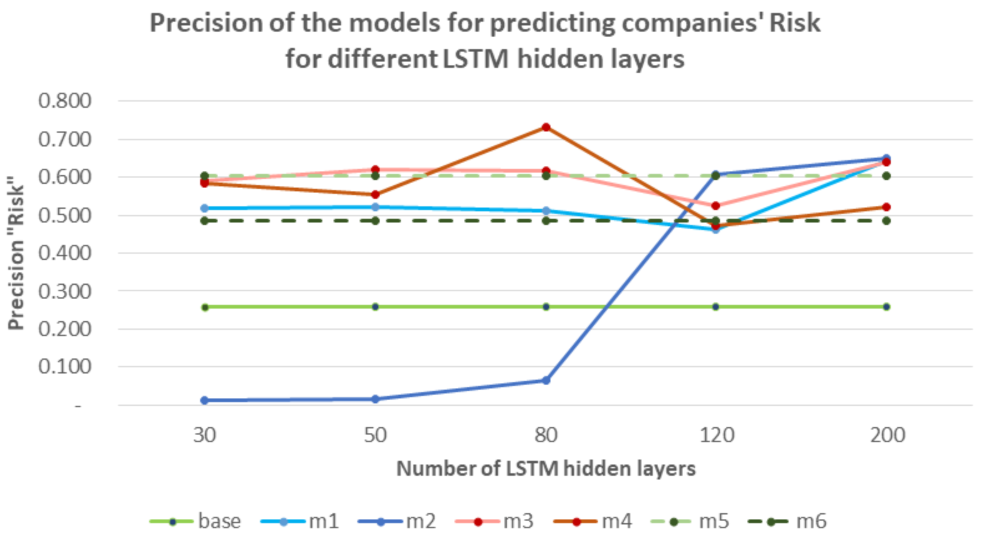


Figure 6.4: Data Analysis: The different models vs numbers of LSTM hidden layers. The best-performing model for bankruptcy prediction is m4 (80 LSTM hidden layers). Image taken from Blanco Lambruschini and Brorsson [2024b]

Figure 6.5 illustrates the performance of bankruptcy prediction using different sequence lengths, which involves providing the network with more information. The best-performing model for this task was achieved with $\kappa = 5$ using the m6 model, reaching nearly 80% precision in predicting a company's risk.

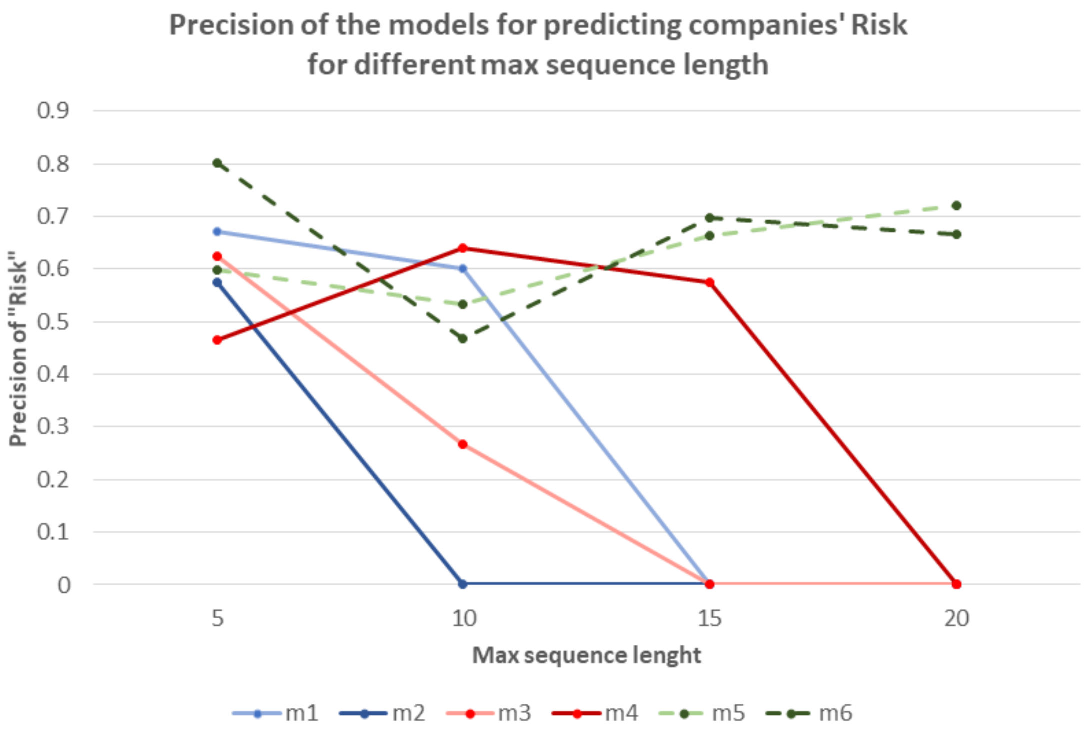


Figure 6.5: Data Analysis: Precision for Bankruptcy prediction for different Sequence Length. The sequence length ranges from 5 to 20 segments. As the number of segments increases, more information is fed into the network. However, the performance of LSTM-based layers degrades as the sequence length grows. In contrast, models utilizing concatenation layers maintain consistently high performance, even with longer sequences. Image taken from Blanco Lambruschini and Brorsson [2024b]

For very long sequences (15 or more segments), the recommended architecture utilizes only the BERT network in the segment analysis phase, followed by a fully concatenated layer, intermediate dense layers, and a final classification layer.

As explained by Blanco Lambruschini and Brorsson [2024b], for processing very long documents with concatenation layers as integration is possible to have approximately 8,000 tokens. This enables us to process around 7,600 English words, 6,600 French words, or 5,100 German words.

6.4 Companies' Cluster Analysis

6.4.1 Data ingestion

Graphs are build using data collected from the different sources. Figure 6.6 illustrates the base model used to develop our primary graph structure. While the database contains numerous tables and columns, only a subset is included in the diagram for clarity.

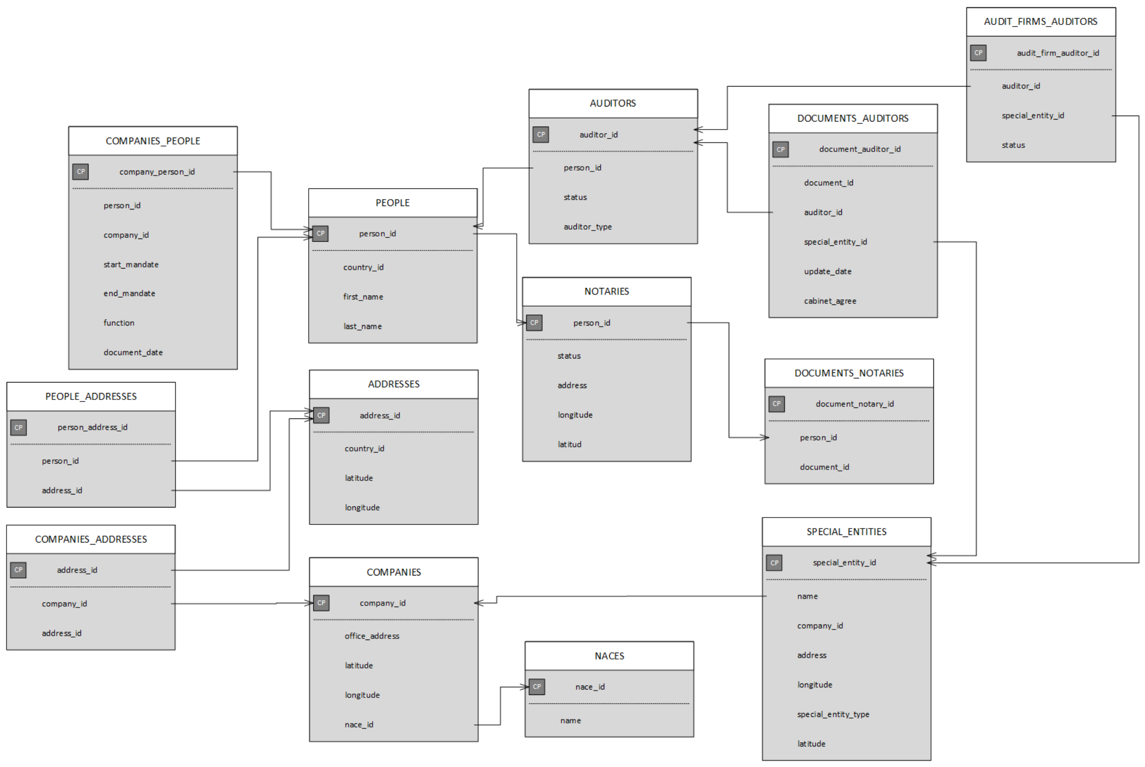


Figure 6.6: Database model for Graph. The base database model consists on 13 tables which has information from the different datasources such as NACE, list of auditors, list of notaries and the information collected from the LBR.

Figure 6.7 presents the proposed graph model. Most of the tables have been consolidated into two primary nodes: Company and Person. The relationships between these nodes are distinguished by edge types. Additionally, the Address node integrates address data from various tables and includes geolocation information.

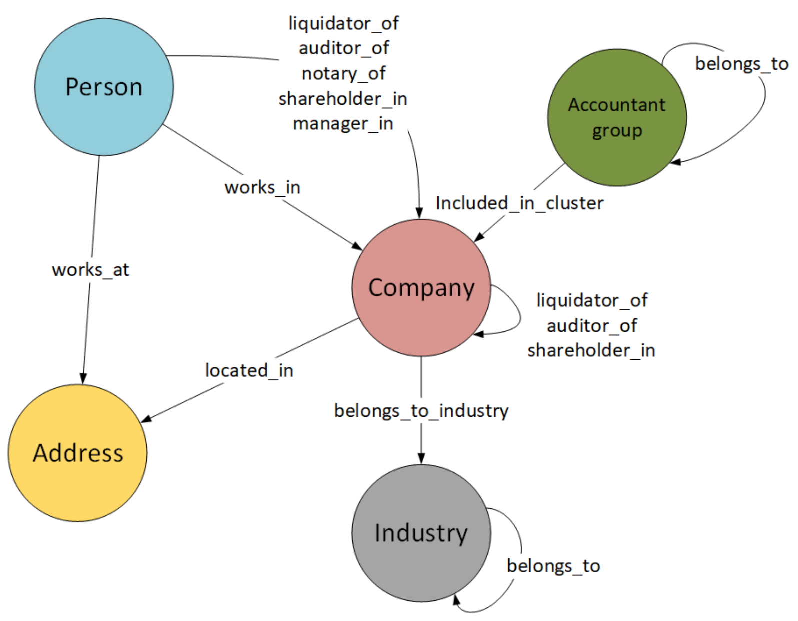


Figure 6.7: Graph Model: Based on the different tables from the database, this model consolidates 13 tables them into 5 nodes. The most important nodes are Company and Person.

Figure 6.8 illustrates a workflow that highlights the primary entities queried from the RDBMS and the processes used to establish relationships between nodes in the graph. Each color corresponds to the nodes defined in Figure 6.7, providing a clear representation of the entity types and their connections.

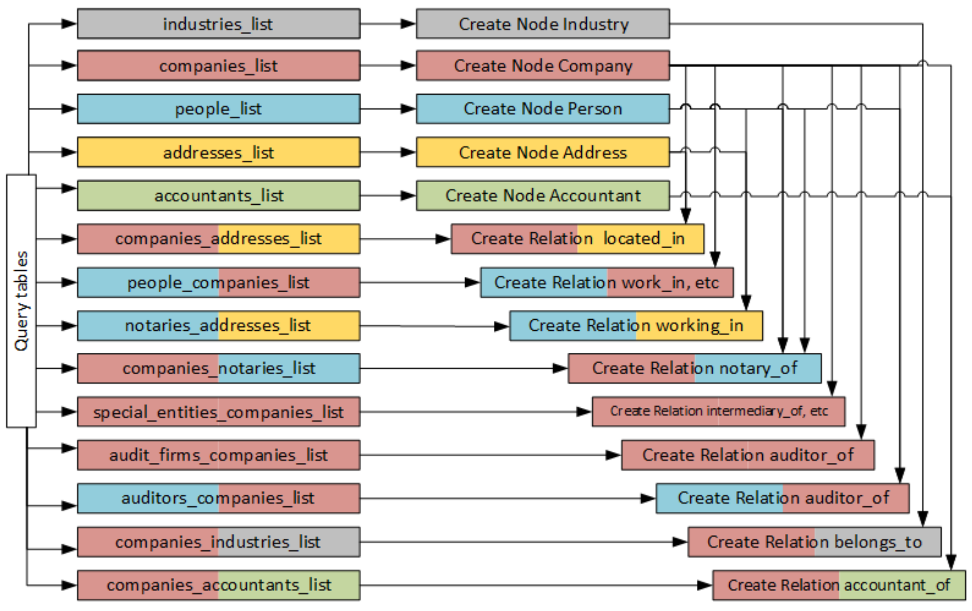

Query tables
industries_list
companies_list
people_list
addresses_list
accountants_list
companies_addresses_list
people_companies_list
notaries_addresses_list
companies_notaries_list
special_entities_companies_list
audit_firms_companies_list
auditors_companies_list
companies_industries_list
companies_accountants_list
Create Node Industry
Create Node Company
Create Node Person
Create Node Address
Create Node Accountant
Create Relation located_in
Create Relation work_in, etc
Create Relation working_in
Create Relation notary_of
Create Relation intermediary_of, etc
Create Relation auditor_of
Create Relation auditor_of
Create Relation belongs_to
Create Relation accountant_of

Figure 6.8: Based on the previous tables, nodes and edges, a workflow was created to ingest the data to the graph.

Figure 6.9 illustrates the different services used to load data into the graph database. The Orchestrator server (Dagster) is the central component. A workflow in the Orchestrator server retrieves data from the RDBMS via SQL queries, processes it, and prepares it for upload to the Graph Server using Cypher queries. An intermediate storage server temporarily holds the files before they are uploaded. This storage server is accessed via an API service with REST endpoints for creating, querying, and deleting files. Consequently, the storage service's REST APIs are accessible from the graph server, and data is uploaded through Cypher queries. This approach significantly accelerates the upload process, reducing it from hours to minutes. To prevent concurrency issues, Dagster workflows generate batch files, and Cypher insert queries (MATCH) are executed sequentially.

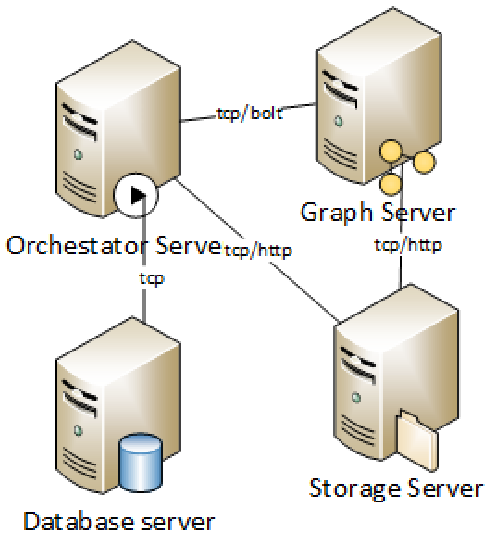


Figure 6.9: To create nodes and relationships, four servers work together. The orchestrator server initiates the workflow by querying the database tables. It then generates a list of node and relationship objects and saves them into files through a REST service. Using Neo4J, these files are uploaded in bulk to the database. Once the upload is complete, the files are deleted to optimize storage.

Once the data is migrated and the nodes and edges are created, a verification process is conducted. The following images provide examples of random data, showcasing various nodes and edges.

Figure 6.10 shows an example of two NACE codes at the lowest level, along with the companies connected to them. The upper NACE levels are also shown but not expanded in the figure to avoid overloading the image.

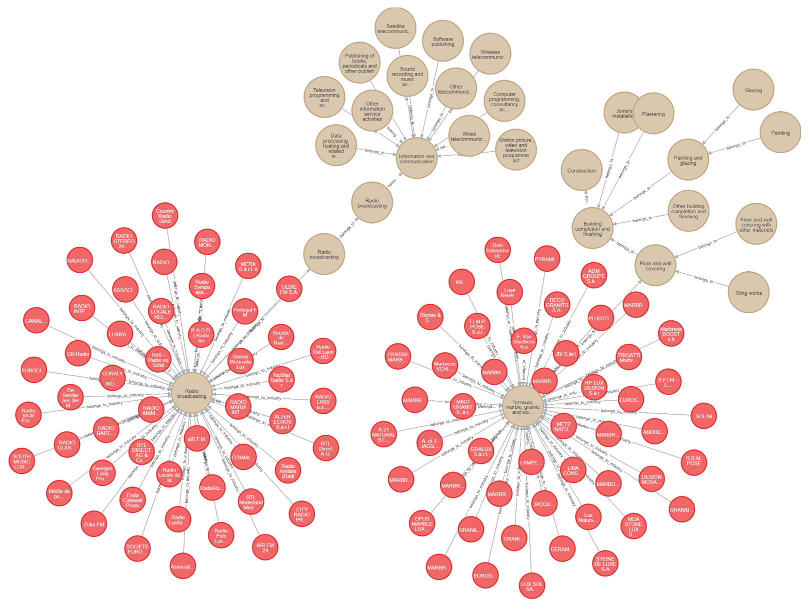


Figure 6.10: An example of two NACE codes (Purple) at the 4th level with the corresponding NACE hierarchy (Brown). These two NACE codes expands the companies connected with them (Red)

Figure 6.11 shows an example of two main address points, each connected to the companies registered at that address. The addresses are linked to other addresses based on geolocation proximity. Using Neo4J's geodesic distance function, three types of edges were created based on proximity (latitude and longitude): the *same_location* edge, which represents identical coordinates or a distance of less than one meter; the *very_close_to* edge, for distances between one and 10 meters; and the *close_to* edge, for distances between 10 and 50 meters (Table 6.5).

Table 6.5: Parameters to define the edge type based on the distance between two company addresses.

Potential Risk	**Range in meters**
same_location	[0,1]
very_close_to	]1,10]
close_to	]10,50]

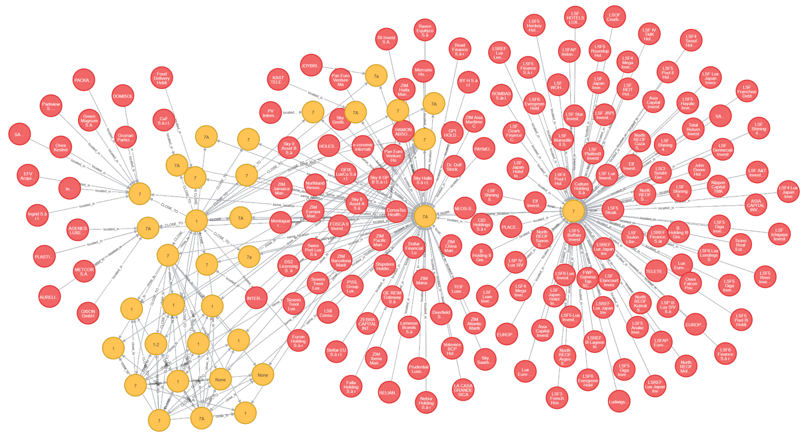


Figure 6.11: An example of two addresses (yellow) and their connections to the companies registered at the same address (red). The addresses are linked to other nearby addresses through proximity edges based on their latitude and longitude.

Figure 6.12 shows three audit firms and their associated companies. Since all the nodes represent companies, they are all in red. However, it is clear from the diagram that each audit firm is the center of a cluster. As companies tend to change audit firms over time, this is represented by the connections between clusters, with companies having two or more edges to different audit firms.

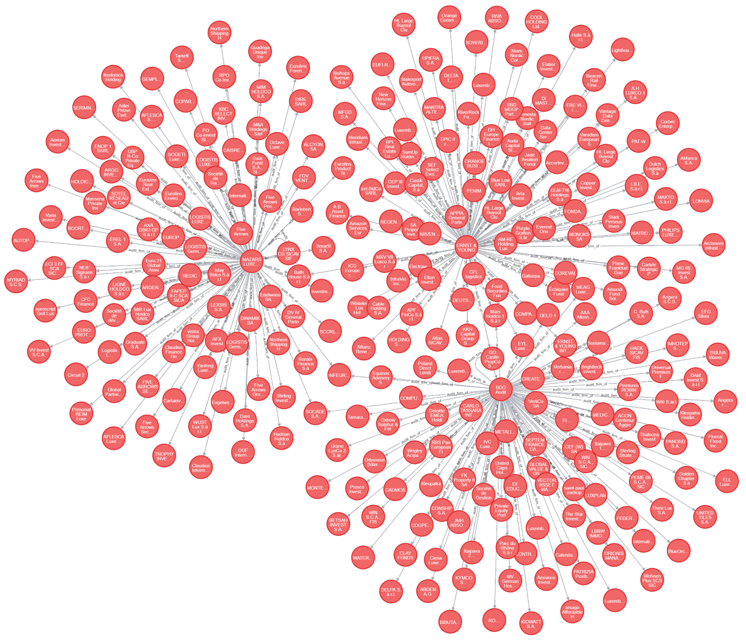


Figure 6.12: This network example might not be easy to differentiate between a company and an audit firm, as both are represented as company nodes in red. The audit firms are the three main centers of the clusters, and these clusters are not isolated but connected to each other through shared audit firms.

Figure 6.13 shows two auditors and their corresponding companies. The auditors may or may not be linked to an audit firm, and the companies may or may not be linked to audit firms as well. As in the previous audit firm diagram, the center of the clusters in this diagram is formed by the auditors and the audit firms.

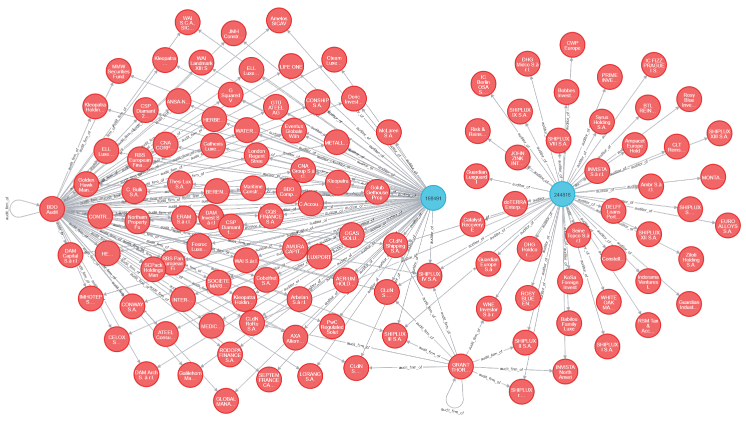


Figure 6.13: An example of two auditors (light blue) and their associated companies (red) is shown, along with the corresponding audit firms (also red). Similar to the audit firms graph example, a company can be associated with more than one audit firm. The central nodes are dominated by the auditors and audit firms.

Figure 6.14 shows four clusters of notaries and their associated companies. Some connections are shared between clusters, indicating that certain companies are linked to more than one notary over time. However, this behavior is not common, as reflected in the graph.

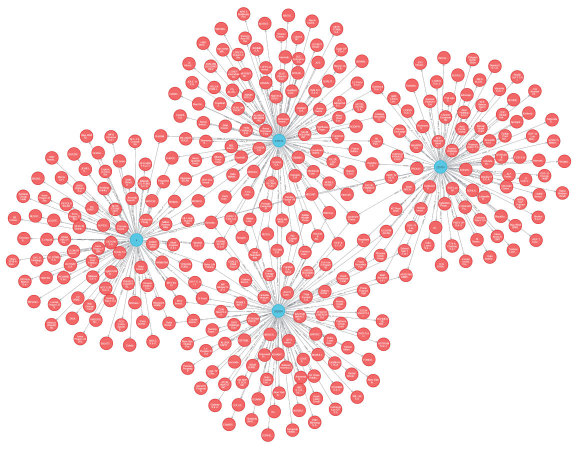


Figure 6.14: An example of four notaries (blue) and the companies related to them (red). The graph shows that some companies are associated with more than one notary, but it is clear that notaries tend to form centralized clusters with a few shared connections.

Figure 6.15 The graph illustrates the relationships between companies and their managers or shareholders. It highlights that individuals (managers/shareholders) can be associated with one or multiple companies. Notably, some individuals manage or own several companies. In a detailed check, the person at the top left is related to 25% of bankrupted companies.

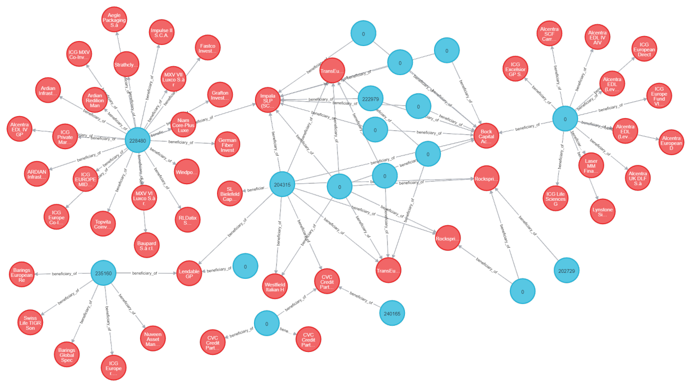


Figure 6.15: This graph shows a group of companies (red) connected to other companies through shared managers or shareholders (light blue). As in other graphs, companies may share managers and shareholders over time.

Figure 6.16 shows a cluster based on the templates used by companies in their annual accounts, along with subclusters representing the authors fingerprint or the accountant. This cluster, along with its subclusters, is notably large relative to the number of companies involved.

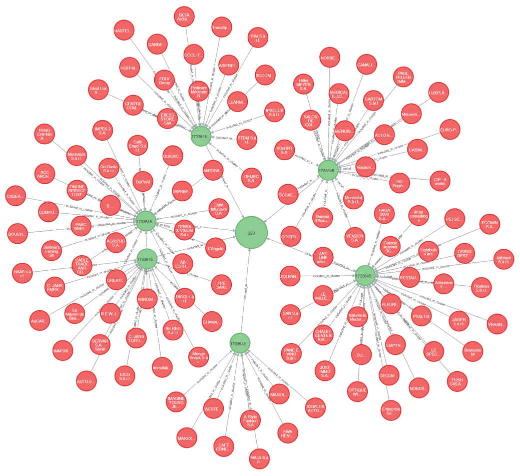


Figure 6.16: The graph depicts an accountant cluster (large green node) along with its corresponding accountant subclusters (smaller green nodes), related to their associated companies (red nodes). In this example, most subclusters are independent, with connections occurring primarily at the cluster level.

Figure 6.17 provides a concise example featuring nodes such as individuals, companies, addresses, and NACE classifications. To maintain clarity and reduce complexity, only a subset of the nodes has been expanded in the graph.

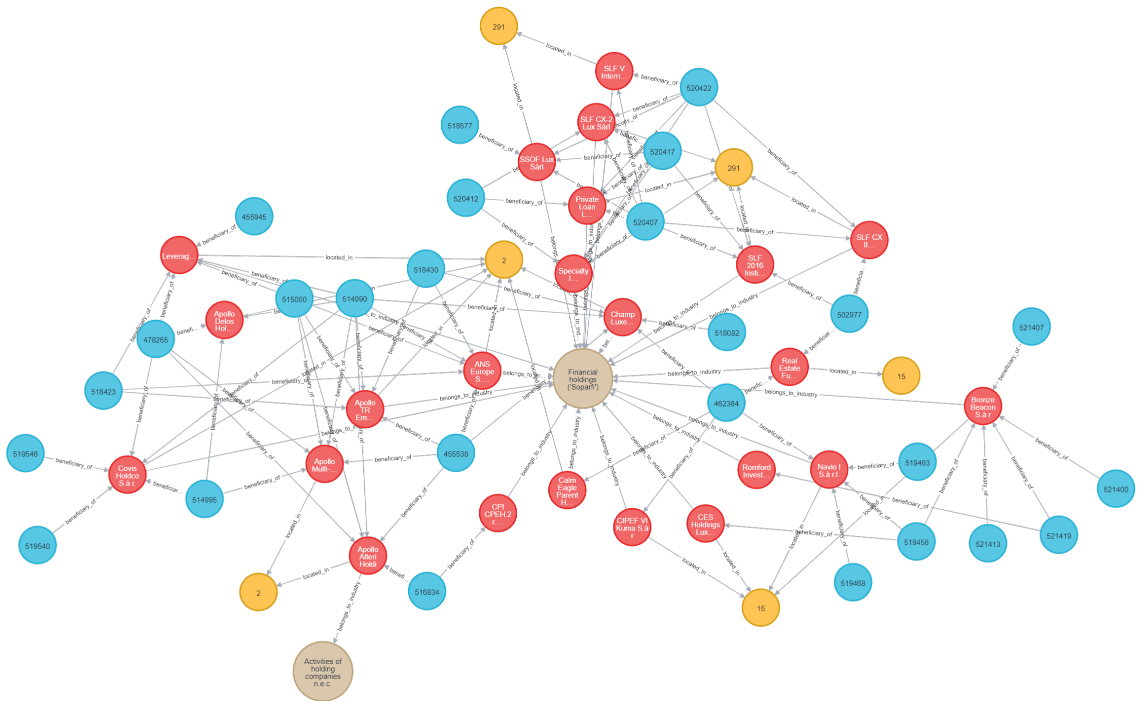


Figure 6.17: This final graph illustrates the relationships between companies (red), people (light blue), addresses (yellow), and other related nodes. To maintain clarity and avoid creating an overly dense visualization, most nodes were not expanded.

6.4.2 Node Graph Centrality

As explained in section 2.12, authors who use graphs have highlighted the importance of analyzing the density of the network between a company and its shareholders as a critical factor in assessing company risk. For the purposes of this study, the centrality analysis is conducted using PageRank and various actors related to the company. The approach involves the creation of four isolated networks to measure PageRank, representing the relationships between companies and: (a) Managers and Shareholders (Figure 6.15), (b) Audit Firms (Figure 6.12), (c) Auditors (Figure 6.13), and (d) Notaries (Figure 6.14).

6.5 Graph Risk Analysis: Multidimensional Risk Proposed Model

This paper proposes and develops a unified multidimensional risk model to analyze data from various sources, along with typical financial ratios. This approach provides decision-makers with a more comprehensive view, enabling them to make better-informed decisions. Figure 6.18 illustrates the five proposed perspectives: management risk, partner relationship risk, environmental and industry risk, sector risk, and financial reporting risk. By considering these five perspectives as potential sources of risk, a deeper understanding of the results can be achieved.

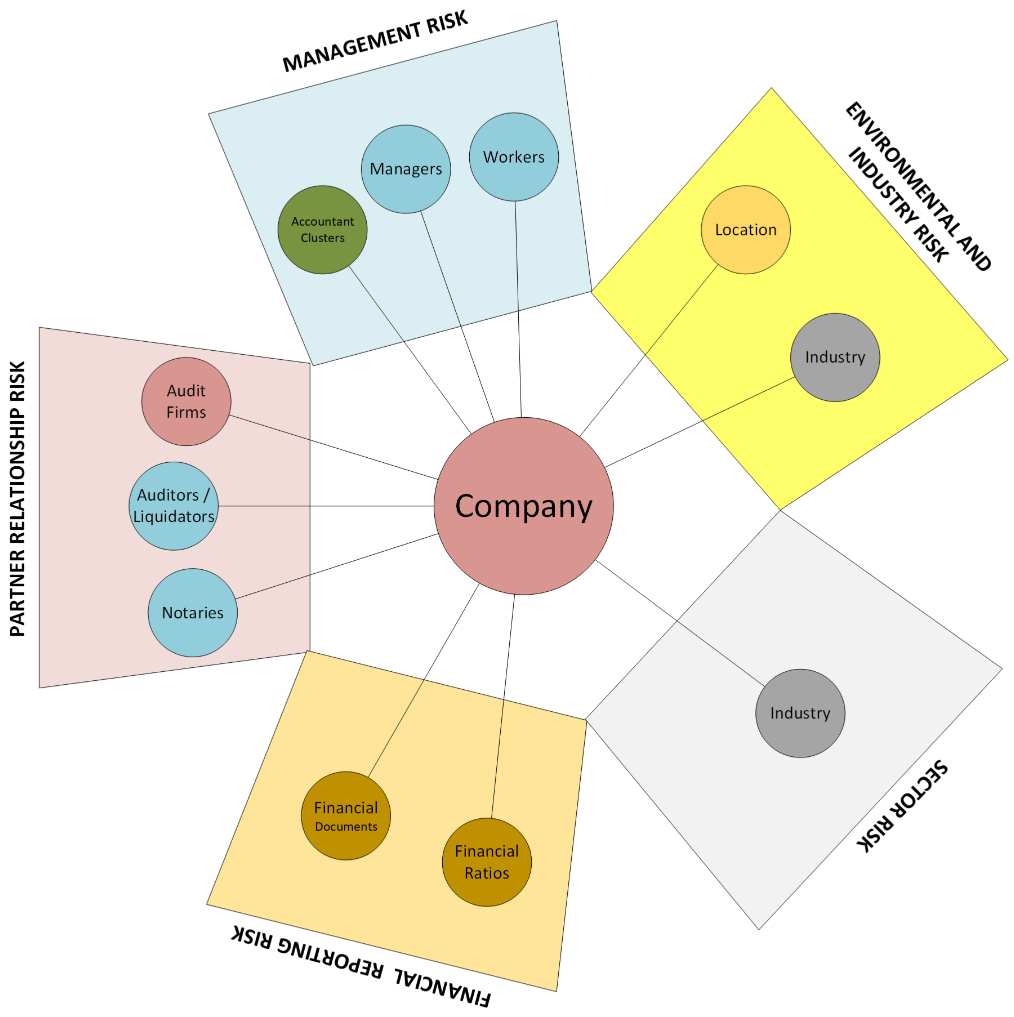


Figure 6.18: The proposed multidimensional risk model consists of five perspectives. By integrating data from the graph, along with inputs from the textual bankruptcy prediction model and the financial ratios model, we aim to build a unified model.

6.5.1 Proposed dimensions

As shown in Equation 6.7, the potential risk of an entity (such as a person, company, address, NACE, etc.) is calculated by dividing the number of related bankrupt companies by the total number of related companies. These relations can be directly or indirectly. It is important to highlight that most relationships are direct, such as those between shareholders and companies, notaries and companies, audit firms and companies, or addresses and companies. However, in some cases, the relationship is indirect. For example, an address is linked to *close* addresses that are associated with companies, and in the case of NACE, only the fourth level is directly related to the companies, while the higher levels have an indirect relationship.

$$\text{Risk\_Entity\_X} = \frac{\sum(\text{Related bankrupted companies})}{\sum(\text{Related companies})} \tag{6.7}$$

6.5.1.1 D1: Management Risk

This perspective focuses on the risk stemming from internal decisions, particularly those involving managers, shareholders, or key employees. The potential risk is associated with the establishment of a company or the hiring of high-level decision-makers, a factor closely linked to companies that have gone bankrupt.

Figure 6.19 illustrates an example of potential management risk. Company M is analyzed but not included in the count. In this example, the potential risk for Company A (Shareholder) is 0 (0/3), while the potential risk for Person Y (Manager) is 0.33 (1/3). Finally, the risk for Person X is the highest among the related entities, with a potential risk of 0.66 (2/3).

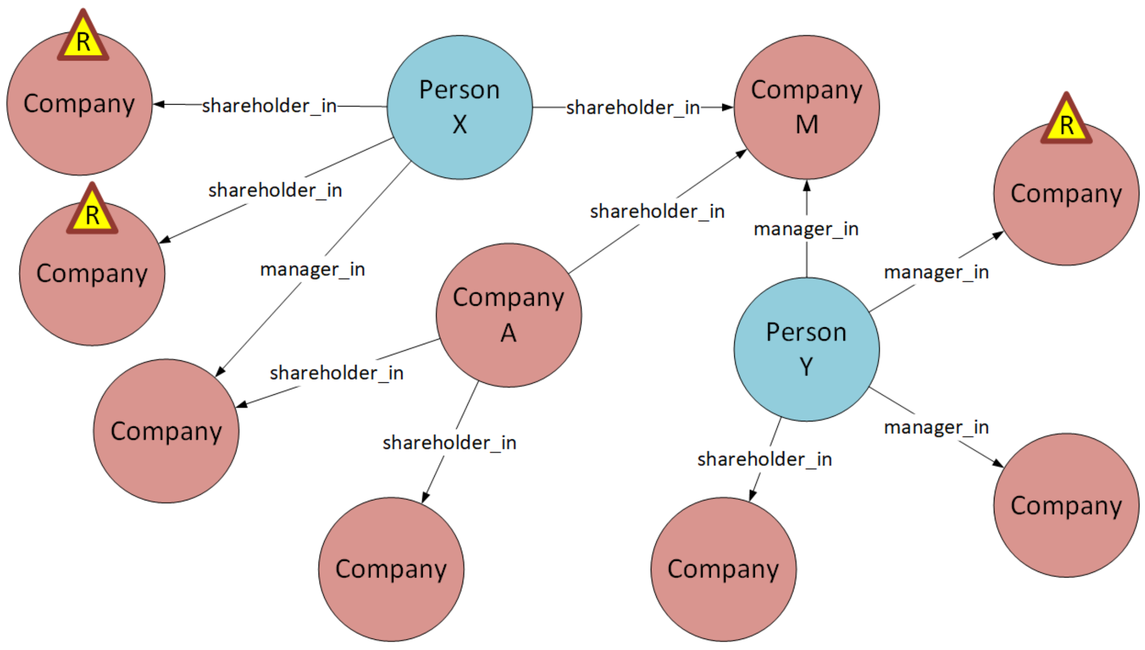


Figure 6.19: The graph illustrates an example of the relationships between a Shareholder, Manager and companies. The associated potential risk for each entity is determined by the number of bankrupt companies linked to it.

6.5.1.2 D2: Partner Relationship Risk

This perspective is related to the relationships a company establishes with external entities such as notaries, auditors, and audit firms, most of which are independent fiscal entities. It also addresses the potential risk associated with selecting external firms or individuals who may negatively impact the company's risk profile due to their associations with other high-risk companies.

Figure 6.20 shows an example of this potential risk. Company M is analyzed but not included in the count. For example the associated potential risk for the Company A (Audit Firm) is 0.67 (2/3) and this is going to be transferred as potential risk to the Company M. On the other hand the potential risk of the Person X (Notary) is also 1.0 (1/1). Finally the potential risk for Person Y (Auditor) is 0.0 (0/2).

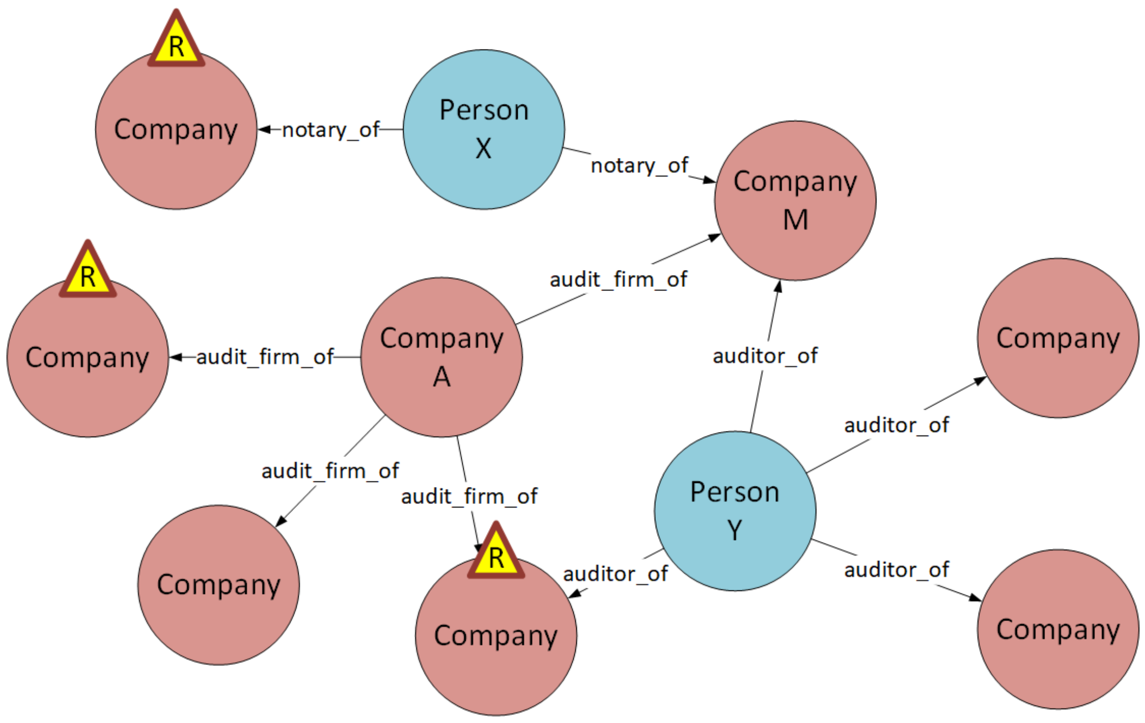


Figure 6.20: The graph illustrates an example of the relationships between an audit firm, auditor, notary, and a company. The associated potential risk for each entity is determined by the number of bankrupt companies linked to it.

6.5.1.3 D3: Environmental and Industry Risk

This potential risk is associated with the risks tied to a specific location, such as a high concentration of bankrupt companies in a particular area due to factors like low market demand or the risk of natural disasters or accidents in the surrounding environment. This perspective not only analyzes the location itself but also evaluates the risks posed to companies within the same sector or NACE classification operating in that location.

Figure 6.21 illustrates an example of potential environmental risk. Company M is analyzed but not included in the count.. In this example, the potential *direct location risk* for Address A is 0 (0/1), the potential *close_to risk* for Address A is 1 (3/3), and the potential *very_close risk* for Address A is 0.66 (2/3). On the other hand, the potential *close_to risk* for Address A within the same industry (NACE L4) is 0 (0/3), and the potential *very_close_to risk* for Address A in the same industry is 0.5 (1/2). The risk associated with an address varies depending on whether the related companies are in the same industry or not.

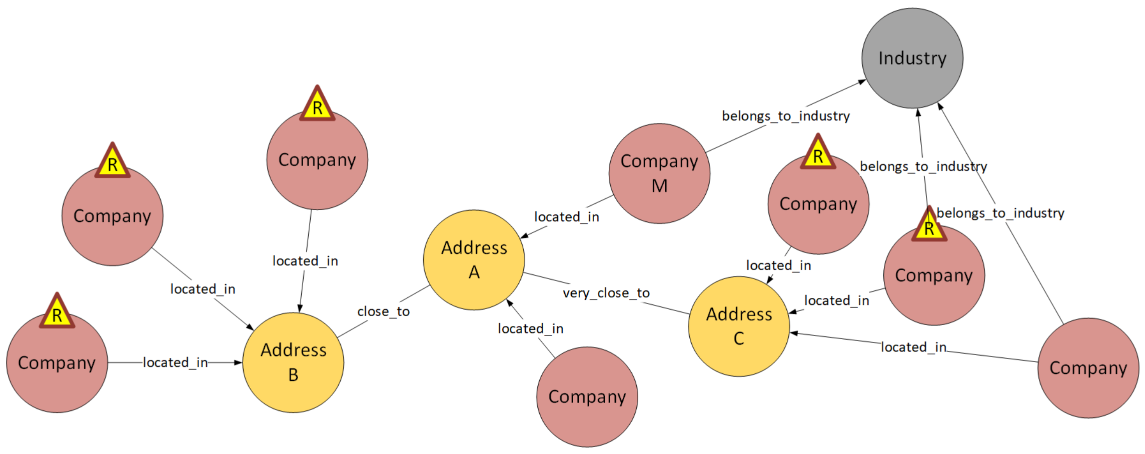


Figure 6.21: The graph illustrates an example of the relationships between an address and a company, as well as between the address and other nearby addresses. The risk is determined by the companies directly and indirectly associated with bankruptcies. Additionally, as with other proposed factors, only companies within the same industry are considered in the analysis.

Figure 6.22 presents a map illustrating how potential environmental risk is influenced by nearby companies and their proximity. In this example, for node 1, the potential *direct location risk* is 0.0 (0/2), the potential *very_close_to risk* is 0.0 (0/0), and the potential *close_to risk* is 1.0 (1/1). To avoid redundant counting, companies are not considered multiple times across different proximity categories.

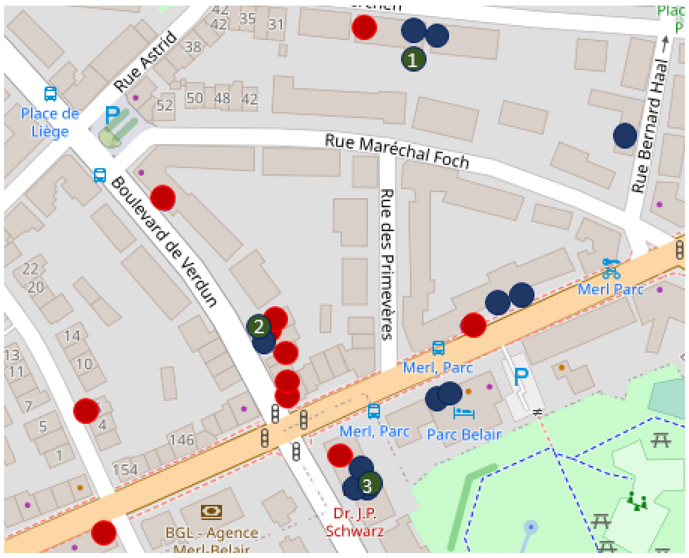


Figure 6.22: The figure illustrates an example of environmental risk, where a company can be surrounded by non-risky, neutral, or risky companies. The potential location risk for nodes 1 and 3 is low (0.333), while it is high for node 2. The degree of risk changes when considering only nodes within the same location, very close proximity, or close proximity.

6.5.1.4 D4: Sector Risk

This perspective analyzes the potential risks inherent to the sector itself, regardless of location, while considering the four levels of the NACE classification. A high number of bankrupt companies within the same industry can increase the risk for the company being analyzed.

Figure 6.23 illustrates an example of the relationships between a company and two levels of NACE classifications. Company M is analyzed but not included in the count. For instance, the potential risk for Industry A at L4 is 1.0 (2/2), while the potential risk for Industry B at L3 is 0.8 (4/5), considering that NACE code L3 (Industry B) consists of two classifications: Industry A and Industry C. The same logic applies to the analysis of the upper levels.

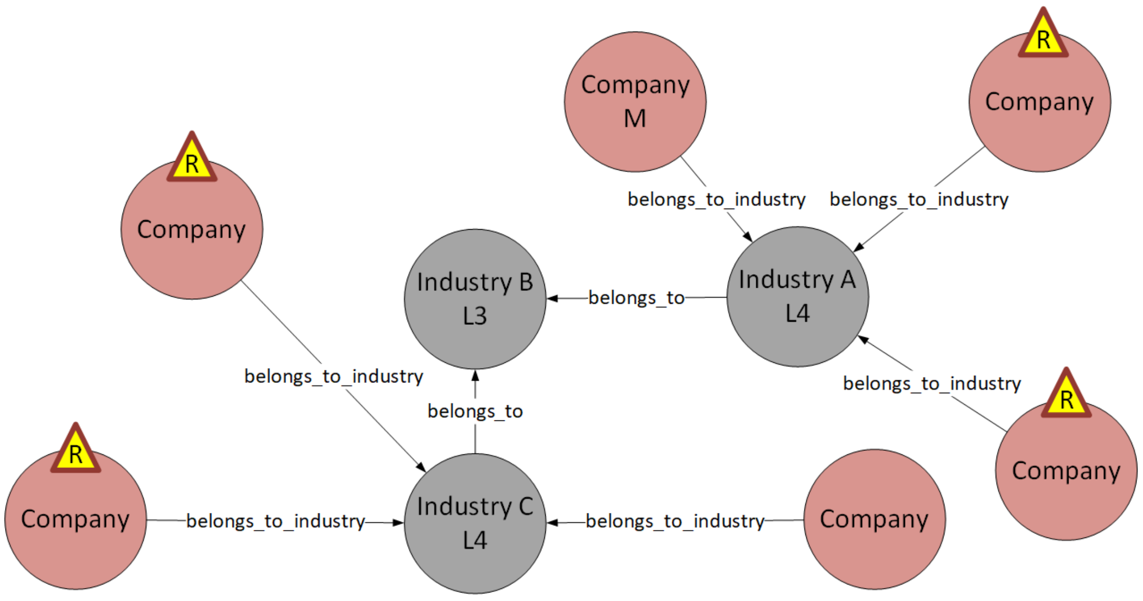


Figure 6.23: The graph illustrates an example of the relationships between two levels of the NACE classification (L4 and L3) and the related companies. The potential risk is determined by the companies connected to the entire downward hierarchy of the analyzed level.

6.5.1.5 D5: Financial Reporting Risk

Finally, this perspective evaluates the risk derived from analyzing financial ratios and the risk prediction based on the textual analysis of financial documents. It aggregates the predictions, ranging from 0 to 1, obtained from both the financial ratios prediction model from Wang et al. [2023] and the text-based prediction model section 6.3.

6.5.2 Proposed sub-dimensions

Based on the five proposed perspectives, Table 6.6 outlines the various components of the multidimensional risk model. As previously explained, each sub-dimension can be related to the company either directly or indirectly, and is calculated using Equation 6.7. Additionally, the sub-dimensions include four separate PageRank results for each company, reflecting the relationships with the corresponding entities. Furthermore, the analysis incorporates the number of occurrences for specific entities directly associated with the company. For example, in the case of notaries, the value represents the total number of notaries associated with the company over time. In total the proposed model is composed of 28 sub-dimensions divided into 5 dimensions or perspectives.

Table 6.6: Perspectives or dimensions and its components or sub-dimensions.

Perspective	N.	Sub-dimension Potential Risk
D1: Partner Relationship Risk	1	D11: Audit Firm Risk
	2	D12: Auditor Risk
	3	D13: Notary Risk
	4	D14: Total Audit Firms Risk
	5	D15: Total Auditors Risk
	6	D16: Total Notaries Risk
	7	D17: Rank of Audit Firms
	8	D18: Rank of Auditors
	9	D19: Rank of Notaries
D2: Management Risk	10	D21: Manager and Shareholders Risk
	11	D22: Accountant Cluster Risk
	12	D23: Accountant Sub-Cluster Risk
	13	D24: Total Managers and Shareholders Risk
	14	D25: Total Accountant Clusters Risk
	15	D26: Total Accountant Sub-clusters Risk
	16	D27: Rank of Manager and Shareholders
D3: Environmental and Industry Risk	17	D31: Same-Location Address Risk
	18	D32: Very-close location Address Risk
	19	D33: Close-location Risk
	20	D34: Same-Location Address within Industry Risk
	21	D35: Very-close location within Industry Risk
	22	D36: Close-location within Industry Risk
D4: Sector Risk	23	D41: Industry L4 Risk
	24	D42: Industry L3 Risk
	25	D43: Industry L2 Risk
	26	D44: Industry L1 Risk
D5: Financial Reporting Risk	27	D51: Financial Ratio Risk
	28	D52: Text-base Reporting Risk

6.5.3 Evaluation of the Integrated proposed multidimensional Risk Model

Once the risk information for the 28 sub-dimensions is calculated for each company, the next step is to assess the predictive capacity based solely on these values. After training and testing the models, the feature importance of the best-performing model is then evaluated. For this, we are evaluating 5 different ML models such as Random Forest, Logistic Regression, SVM, MultiLayer Perceptron (MLP) and LightGBM.

6.5.3.1 Dataset

The training and testing datasets will be used for the five different models. One of the key characteristics of this data is its high sparsity, particularly for SMEs. To address this, records with excessive sparsity were filtered out. The performance of the models was evaluated with different parameters, and it was decided that a minimum of 50% non-null data should be required for a record to remain in the dataset. Additionally, the models were trained using different dataset sizes. Since the performance did not show significant variation beyond 8,000 records, this size was chosen as the final dataset size. Finally, missing data was replaced with -1, and the data was normalized by column. The same dataset was used as input for all five models.

6.5.3.2 Models and metrics to Evaluate

The five models to evaluate the sub-dimensions are:

- **Random Forest**: The model uses 100 estimators and Gini criteria and the other default values of sklearn RandomForestClassifier library.
- **Logistic Regression** The model used for Logistic Regression uses the default parameters of sklearn LogisticRegression library.
- **SVM** For this model, a RBF non linear kernel was used and used an scaler.The regularization parameter is set to 1. sklearn SVC library is used.
- **MLP** The library sklearn MLPClassifier is used and the size of the hidden layers which best performs is (20,5) with 500 max iterations.
- **LightGBM** In this case the library used is LightGBM with the boosting type of GBDT, learning rate of 0.05, feature fraction of 0.9 and binary objective.

The metrics to be used to compare are accuracy, precision, recall, F1-score and AUC.

6.5.3.3 Comparison of models performance

After executing each model three times with the parameters previously mentioned, the results are shown in the Table 6.7

Table 6.7: Results of testing the proposed sub-dimensions with different ML models.

Model	Accuracy	Precision	Recall	F1-score	AUC
Random Forest	**0.8387**	0.8309	0.8387	0.8319	0.7456
Logistic Regression	0.8100	0.8005	0.8100	0.7814	0.6484
SVM	0.8281	0.8185	0.8281	0.8123	0.7004
MLP	0.8356	0.8286	0.8356	0.8305	0.7497
LightGBM	0.8362	**0.8553**	**0.9429**	**0.8970**	**0.8922**

The best-performing model was consistently LightGBM, which also demonstrated the highest performance in the study by Wang et al. [2023], achieving a maximum AUC of 0.8713. This indicates that the integrated model developed in this study improved performance to an AUC of 0.8922, providing enhanced information to support decision-making.

Ablation studies were conducted by systematically removing components such as the financial ratios score, total counters, centrality rankings, and dimensions. The results of these studies are presented in Table 6.9.

Table 6.8: Ablation studies of the different groups of the proposed sub-dimensions.

Model	Accuracy	Precision	Recall	F1-score	AUC
Baseline	0.8362	0.8553	**0.9429**	0.8970	**0.8922**
D5: Financial Reporting Risk	0.8281	0.8185	0.8281	0.8123	0.7004
D2: Management Risk	0.7943	0.8275	0.8198	0.8712	0.8043
D1: Partnership Risk	0.8181	0.8454	0.8392	0.8898	0.8395
D3: Environmental Risk	0.8262	0.8560	0.9394	0.8958	0.8607
D4: Sector Risk	**0.8418**	**0.8802**	0.9278	0.9033	0.8671
Centrality Ranking	0.8450	0.8750	0.9374	**0.9051**	0.8754

From the previous table, it is important to highlight that, for the AUC analysis, each individual dimension contributes to improving the model's performance. However, for other metrics, removing the sector risk dimension yields even better results. This outcome may be attributed to the nature of industry classification. A company can register multiple business purposes, but only one is assigned by the LBR, which may not necessarily reflect the company's primary source of income.

6.5.3.4 Final weighted proposition

Figure 6.24 illustrates the top 18 most important features for the LightGBM model in predicting a company's risk based on multiple sources of information. As anticipated, the Financial Ratio Risk is among the most significant features. Notably, the accountant cluster emerges as the single most important feature for the model. Other critical features include information related to managers, shareholders, auditors, and nearby locations. Additionally, features derived from rank centrality algorithms, particularly those associated with auditors, managers, and shareholders, are also deemed important for the model's performance.

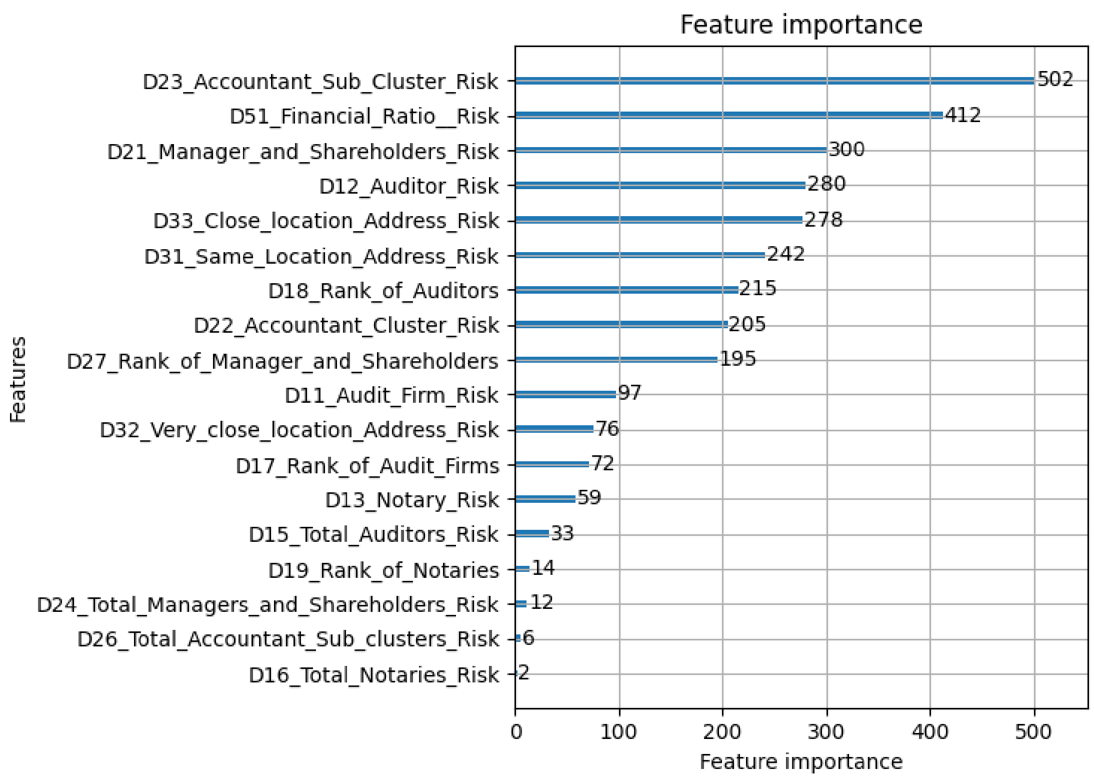


Figure 6.24: The figure shows the most important sub-dimensions for the LightGBM predictive model

Based on the importance of the proposed sub-dimensions, the following weights are suggested for the multidimensional risk analysis. This approach involves taking the feature importance of each sub-dimension and adding an error margin of 5 to each individual sub-dimension. The feature contributions of all components within a dimension are then summed, and the total is averaged across the number of sub-dimensions. The resulting average contribution to importance represents the percentage of contribution for the overall integrated analysis.

Table 6.9: Contribution analysis of the different groups of the proposed sub-dimensions.

Dimension / Perspective	**Weight**
D1: Partner Relationship Risk	15%
D2: Management Risk	30%
D3: Environmental and Industry Risk	18%
D4: Sector Risk	01%
D5: Financial Reporting Risk	36%

It is important to note that, even though the Sector Risk does not directly contribute to risk prediction, including it in a Potential Risk report can provide decision-makers with additional insights to support more informed decision-making.

6.5.4 Multidimensional Risk Report

As part of the research, a report is proposed to provide detailed insights into the risk score. As shown in Figure 6.25, the proposed graph illustrates the five dimensions along with statistical values for each dimension within the same industry as the company. The data is visualized using a box plot, allowing for a clear comparison of the variability and central tendencies of the data from different dimensions.

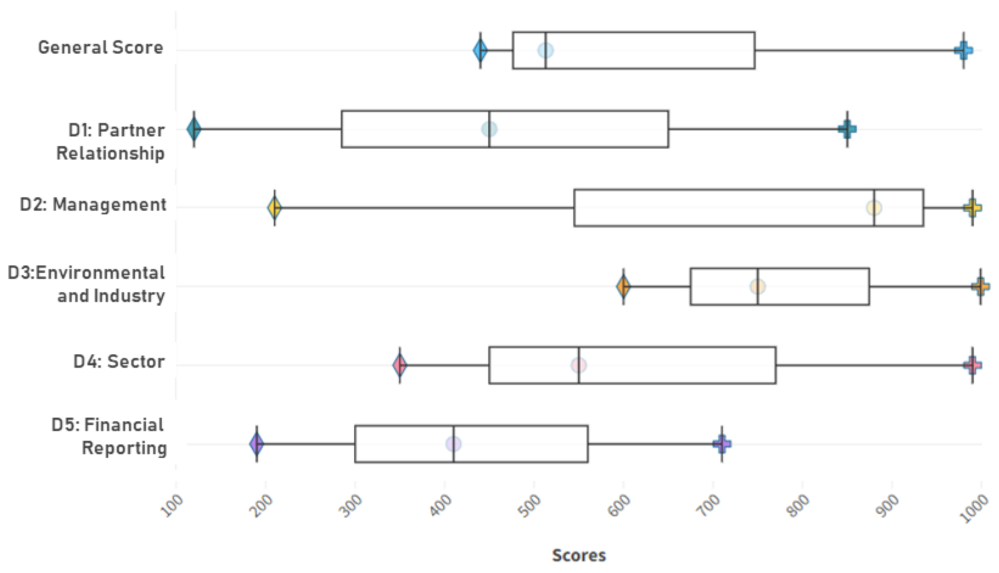


Figure 6.25: The proposed report contains the potential risk score in the five dimensions and the comparison with the companies in the same industry.

When clicked, the explanatory graph or chart is displayed. For instance, as shown in Figure 6.26, a graph network is proposed for dimensions D1 and D2. This graph illustrates the network and node assessments while ensuring the anonymity of related company names. Nodes are color-coded to indicate their status: red for risky, green for non-risky, and gray for no data available.

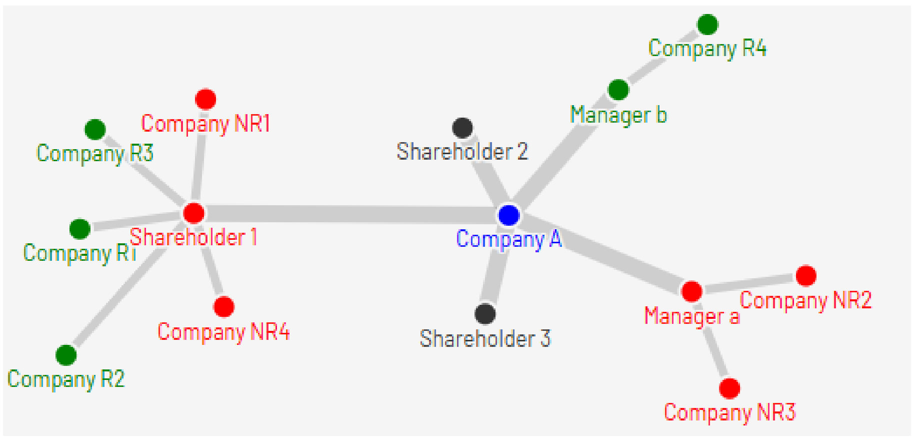
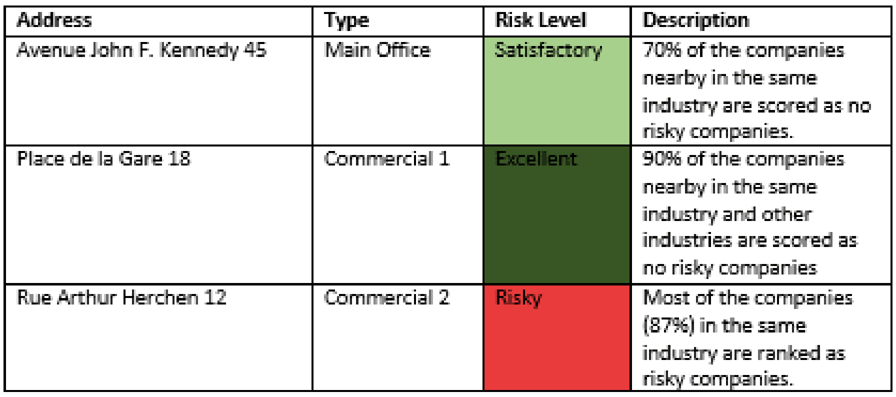


Figure 6.26: The proposed report contains the potential risk score in the five dimensions and the comparison with the companies in the same industry.

For the remaining dimensions, such as D3, D4, and D5, a potential risk matrix is proposed, accompanied by a detailed evaluation presented in natural language. The matrix incorporates color-coding to visually represent the data and highlights key insights accordingly.



Address	Type	Risk Level	Description
Avenue John F. Kennedy 45	Main Office	🟩 Satisfactory	70% of the companies nearby in the same industry are scored as no risky companies.
Place de la Gare 18	Commercial 1	🟩 Excellent	90% of the companies nearby in the same industry and other industries are scored as no risky companies
Rue Arthur Herchen 12	Commercial 2	🟥 Risky	Most of the companies (87%) in the same industry are ranked as risky companies.

Figure 6.27: The proposed report contains the potential risk score in the five dimensions and the comparison with the companies in the same industry.

6.6 Summary

In this section, the final step of the data pipeline is presented, where most of the utilized models are Deep Learning models, along with one Machine Learning model. The proposed autoclustering model is a Machine Learning clustering model that processes the annexes of the annual accounts and uses textual information to cluster the documents according to their format and subcluster them based on the authors fingerprint. The performance of the model is measured using the Rand Index, which assesses the accuracy of document assignment to specific clusters. Initial results indicated a Rand Index of 87% for template-based clustering and 55% for the authors fingerprint, or the *hidden author*. Using the proposed long-text BERT architecture, a model was fine-tuned to predict the risk of bankruptcy based solely on the annexes of the Financial Reports. The model achieved a precision of 73% in predicting a company as risky. Finally, all previously extracted information and predictions were integrated into a Graph Network to conduct a comprehensive multidimensional risk assessment. Utilizing a straightforward LightGBM model, the integrated approach achieved a bankruptcy risk prediction with an F1-score of 89.70%. The proposed multidimensional model comprises five dimensions or perspectives, each assigned a suggested weight of importance. These weights can be applied to evaluate a company's overall risk effectively.

Chapter 7

Data Presentation and exploitation

Contents

7.1 Introduction 141

7.2 Business Dashboard 142

7.2.1 Proof of Concept 142

7.2.2 Solution Design 146

7.2.3 Implementation, Testing and Deployment 147

7.3 Implemented Dashboard 149

7.4 Summary 152

7.1 Introduction

This chapter presents the results of the current project through a user-friendly interface. The initial version of the interface was designed to assist users in navigating the data, analyzing each company, and gaining valuable insights. There are two main user interfaces tailored to three different user roles. The first is the Business Dashboard, intended for the end user as well as the platform's IT business manager, who oversees access management. The second is the IT Dashboard, designed for the IT platform manager to manage the data pipeline. This section will focus on presenting the Business Dashboard.

7.2 Business Dashboard

As explained in Section 3.5.1, after analyzing the data and user interface requirements, a set of features was developed for implementation in a proof of concept, which was then presented to the partner for approval.

7.2.1 Proof of Concept

For the proof of concept, a list of requirements was created based on the available data. At the time of the proof of concept, the available data primarily consisted of company information provided by the business partner, the initial risk score, the geolocation of company addresses, and the processed extraction of text information from a limited number of available PDFs. The requirements are specified in Table 7.1.

Table 7.1: Business Dashboard: Business Requirements for Proof of concept

Code	**Category**	**Description (The dashboard should ...)**
REQ1	Industries	Display information about industries, showing the number of companies categorized as either active or inactive, as well as the total number of companies per industries.
REQ2	Industries	Allow users to drill down through sub-industries up to the fourth level according to the NACE hierarchy. This functionality will enable users to view detailed information about companies at progressively more specific levels within each industry.
REQ3	Companies	Provide a list of companies with essential details, including Name, Code, Industry, Status, Rating, and Progress of Data Processing. This will enable users to easily access and review key company information in a structured format.
REQ4	Companies	Include options to access a detailed profile, view a detailed risk evaluation, print the report, check the company's location, and highlight the companys status. These features will enhance user interaction by providing comprehensive access and functionality directly from the company list.
REQ5	Companies	Display detailed information about the company's location and a list of associated documents. This functionality will provide users with comprehensive insights and access to relevant company details upon request.
REQ6	General	Display the information by country and allowing the user to check the data according to the data source.

Based on the initial requirements, a set of tools to conduct a straightforward proof of concept were analyzed. The selected base languages were Python and HTML. Given these technological constraints, they decided to create a proof of concept for the following tools Table 7.2.

Table 7.2: Business Dashboard: Evaluation of web-tools

Technologies	Description
Plotly Dash[1]	This tool is specifically designed for web-based dashboards and integrates easily with pandas and Plotly. It allows to create applications with minimal code. No need for interacting at code level with HTML, CSS and javascript. The compilation and execution is at server side, which makes light front ends.
Flask[2]	This tool is suitable for small and big applications that requires custom solutions. It allows to build applications with fine-grained control over the components. Can be complemented with HTML, CSS and javascript libraries. Also server-side execution.

Two main components were identified: an interactive donut chart to display industries and an interactive table for companies. Both technologies offer the option for an interactive donut chart, but neither provides robust support for an interactive table. Dash has fewer options for creating interactive and visual tables compared to Flask. To achieve both components with Flask, it is necessary to incorporate Bootstrap[3], D3[4] and Tabulator[5].

Flask allows for working with templates and updating only specific parts of the panel. Based on this, the proposed dashboard was divided into four sections, as illustrated in Figure 7.1. The main components include the main options bar, where the country of analysis can be selected, the search bar, the options panel that changes based on the selected main option, and the main panel where the data is displayed.

As shown in Figure 7.2, the main options bar includes, in addition to country selection, the ability to view different data sources and the status of data collection, as well as a list of companies, which is displayed later. The last two options, data flow and models, are intended to be viewed at a general level but can also be accessed at the record or company level through the list of companies.

Once one country is selected the left panel is showed accordingly. In the proposed page, is showed the information divided into the different dataset and type of data as shown in Figure 7.3.

[3]https://bootstrap-flask.readthedocs.io/en/stable/
[4]https://d3js.org/
[5]https://www.tabulator.info/

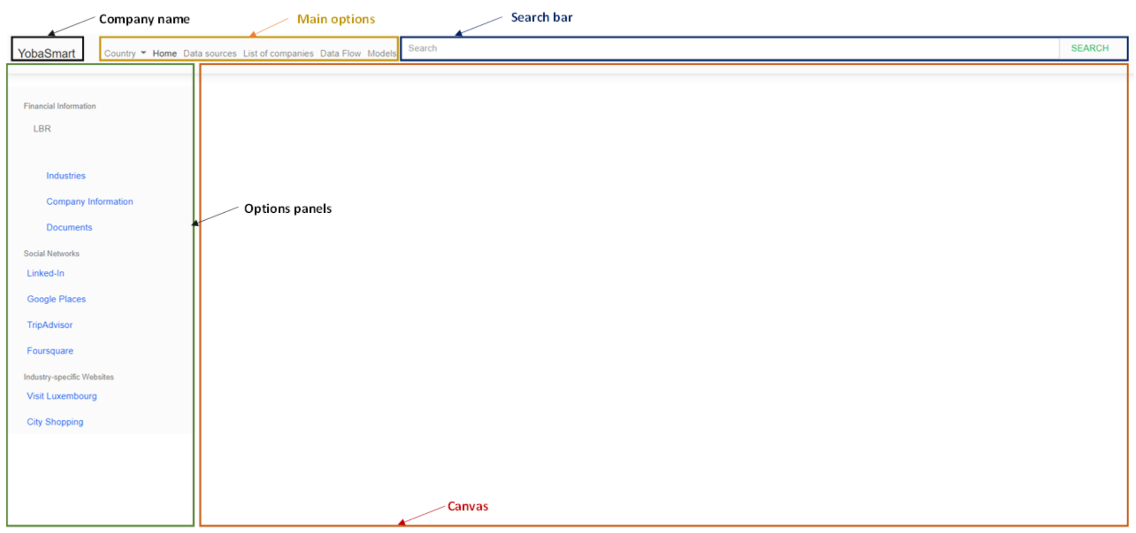


Figure 7.1: Business Dashboard: Organization Layout. The image illustrates the primary layout of the proposed dashboard, highlighting four key components: the main options bar, the search bar, the option panels, and the central canvas.

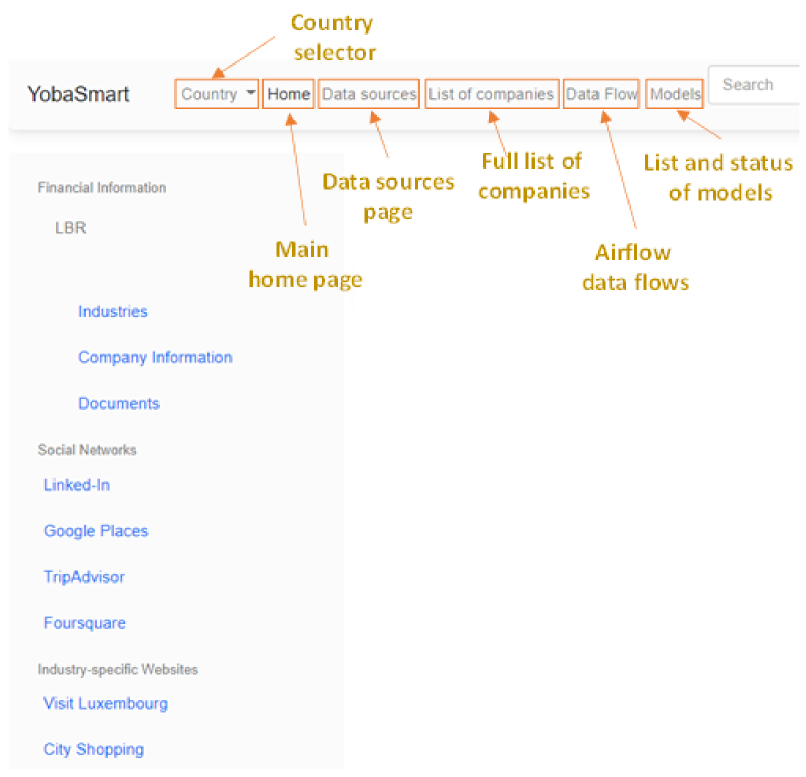


Figure 7.2: Business Dashboard: Main Menu Layout. The main menu includes a dropdown list of countries, a home page link, detailed views of data sources, a list of companies, data flows, and models.

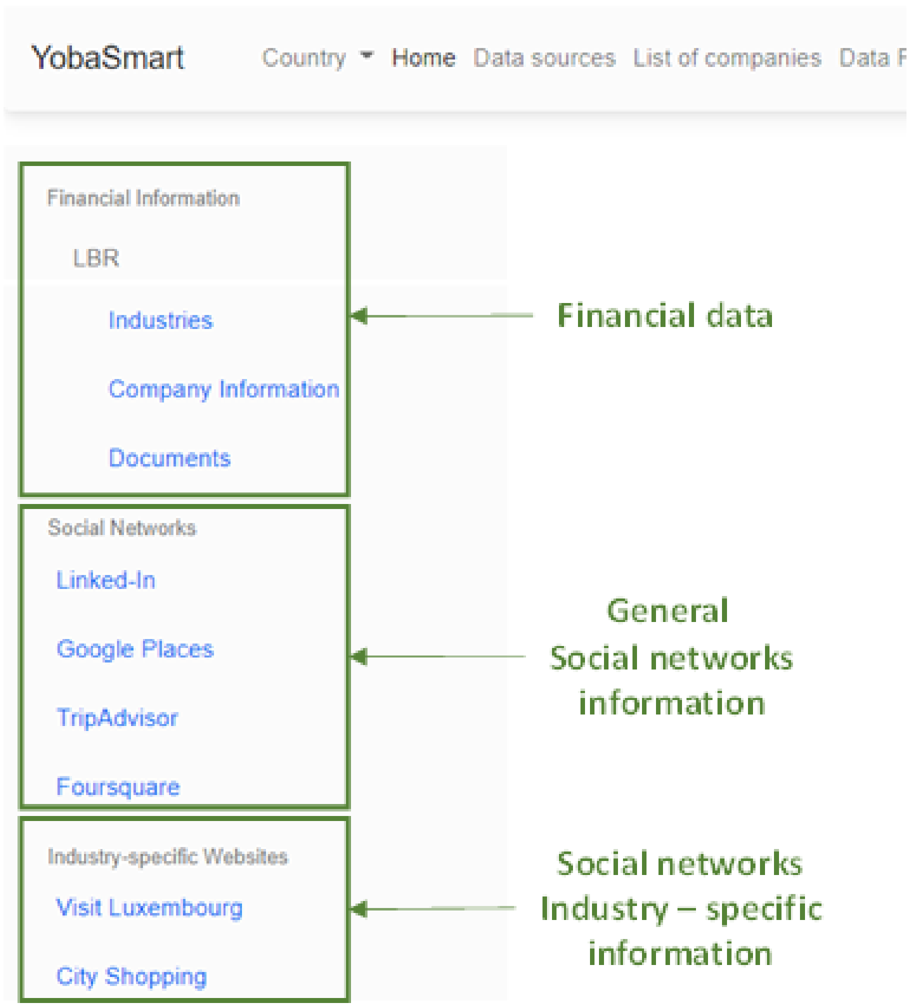


Figure 7.3: Business Dashboard: Datasources Layout. The detailed view of data sources categorizes them by type. This approach encourages the use of non-official sources, such as social networks, alongside traditional data sources.

In the following charts, the data displayed is randomly selected and augmented for proof of concept purposes.

Figure 7.4 illustrates the section for Industry F: Construction of buildings, which includes a total of 31 registered companies. The donut chart displays four levels of hierarchy according to the NACE standard. Similarly, other industries can be selected in the lower part of the donut chart, and the corresponding section of the donut will be highlighted.

Figure 7.5 presents a paginated list of companies, displaying the necessary information. A progress bar indicates the status of the data pipeline processing for each company, with additional details accessible via the other row buttons. Companies that are bankrupt are highlighted in red.

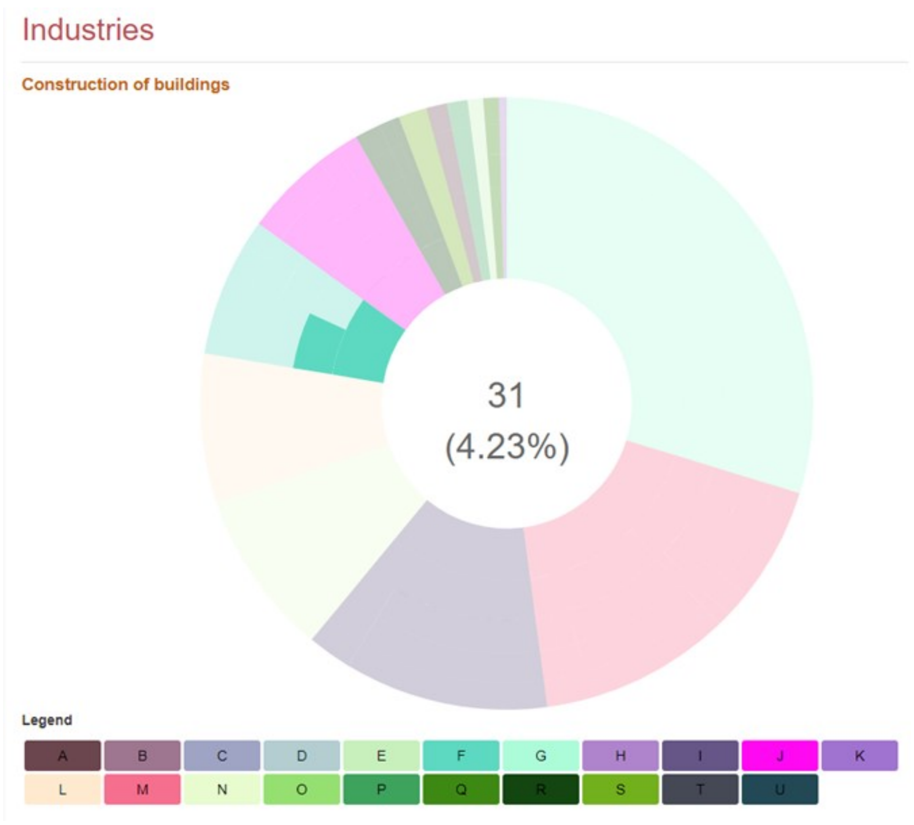


Figure 7.4: Business Dashboard: Industries. This chart displays the distribution of companies across various NACE codes, spanning four levels and 21 industries.

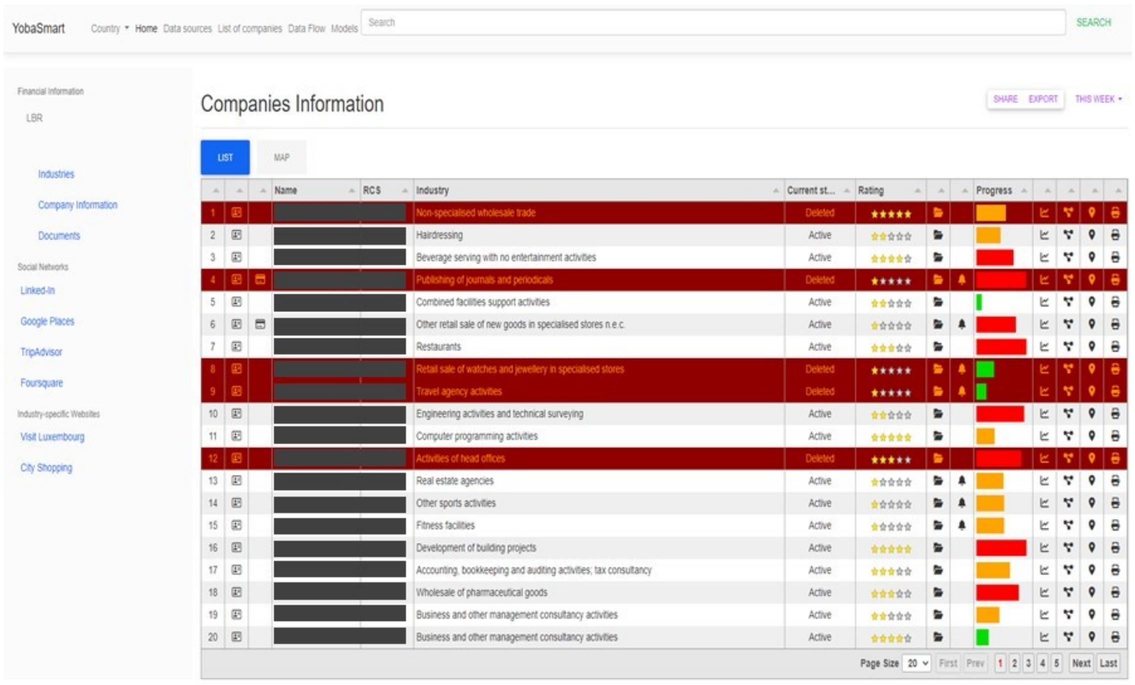
When a company is selected, as shown in Figure 7.6, its basic information is displayed along with additional details such as risk level and financial ratios. Furthermore, other relevant data, including social network metrics and location-based ratings, are also presented.

Another available option is to view public documents, with a link provided to the corresponding public URL. Additionally, the table displays calculated information, including the document's language and the progress of the OCR and extraction process.

As a final page, the location information is diplayed using the geolocated information from the company registered office as shown in Figure 7.8.

7.2.2 Solution Design

The tool was selected, and brainstorming meetings were held to define the list of requirements and divide them into phases for incremental implementation. Roles were assigned, and tasks for the various compo-



Companies Information

	Name	RCS	Industry	Current st...	Rating		Progress
1	[redacted]	[redacted]	Non-specialised wholesale trade	Deleted	★★★★★		🟧
2	[redacted]	[redacted]	Hairdressing	Active	★★☆☆☆		🟧
3	[redacted]	[redacted]	Beverage serving with no entertainment activities	Active	★★★★☆		🟥
4	[redacted]	[redacted]	Publishing of journals and periodicals	Deleted	★★★★★	🔔	🟥
5	[redacted]	[redacted]	Combined facilities support activities	Active	★★☆☆☆		🟩
6	[redacted]	[redacted]	Other retail sale of new goods in specialised stores n.e.c.	Active	★☆☆☆☆	🔔	🟥
7	[redacted]	[redacted]	Restaurants	Active	★★★☆☆		🟥
8	[redacted]	[redacted]	Retail sale of watches and jewellery in specialised stores	Deleted	★★★★★	🔔	🟩
9	[redacted]	[redacted]	Travel agency activities	Deleted	★★★★★	🔔	🟩
10	[redacted]	[redacted]	Engineering activities and technical surveying	Active	★★☆☆☆		🟥
11	[redacted]	[redacted]	Computer programming activities	Active	★★★★★		🟧
12	[redacted]	[redacted]	Activities of head offices	Deleted	★★★★★		🟥
13	[redacted]	[redacted]	Real estate agencies	Active	★☆☆☆☆	🔔	🟧
14	[redacted]	[redacted]	Other sports activities	Active	★☆☆☆☆	🔔	🟧
15	[redacted]	[redacted]	Fitness facilities	Active	★☆☆☆☆	🔔	🟧
16	[redacted]	[redacted]	Development of building projects	Active	★★★★★		🟥
17	[redacted]	[redacted]	Accounting, bookkeeping and auditing activities; tax consultancy	Active	★★★☆☆		🟧
18	[redacted]	[redacted]	Wholesale of pharmaceutical goods	Active	★★★☆☆		🟥
19	[redacted]	[redacted]	Business and other management consultancy activities	Active	★★☆☆☆		🟧
20	[redacted]	[redacted]	Business and other management consultancy activities	Active	★★★★☆		🟩

Figure 7.5: Business Dashboard: Companies list. The proposed interface for the company list includes indicators for inactive companies, rankings, and other useful links.

nents were delineated.

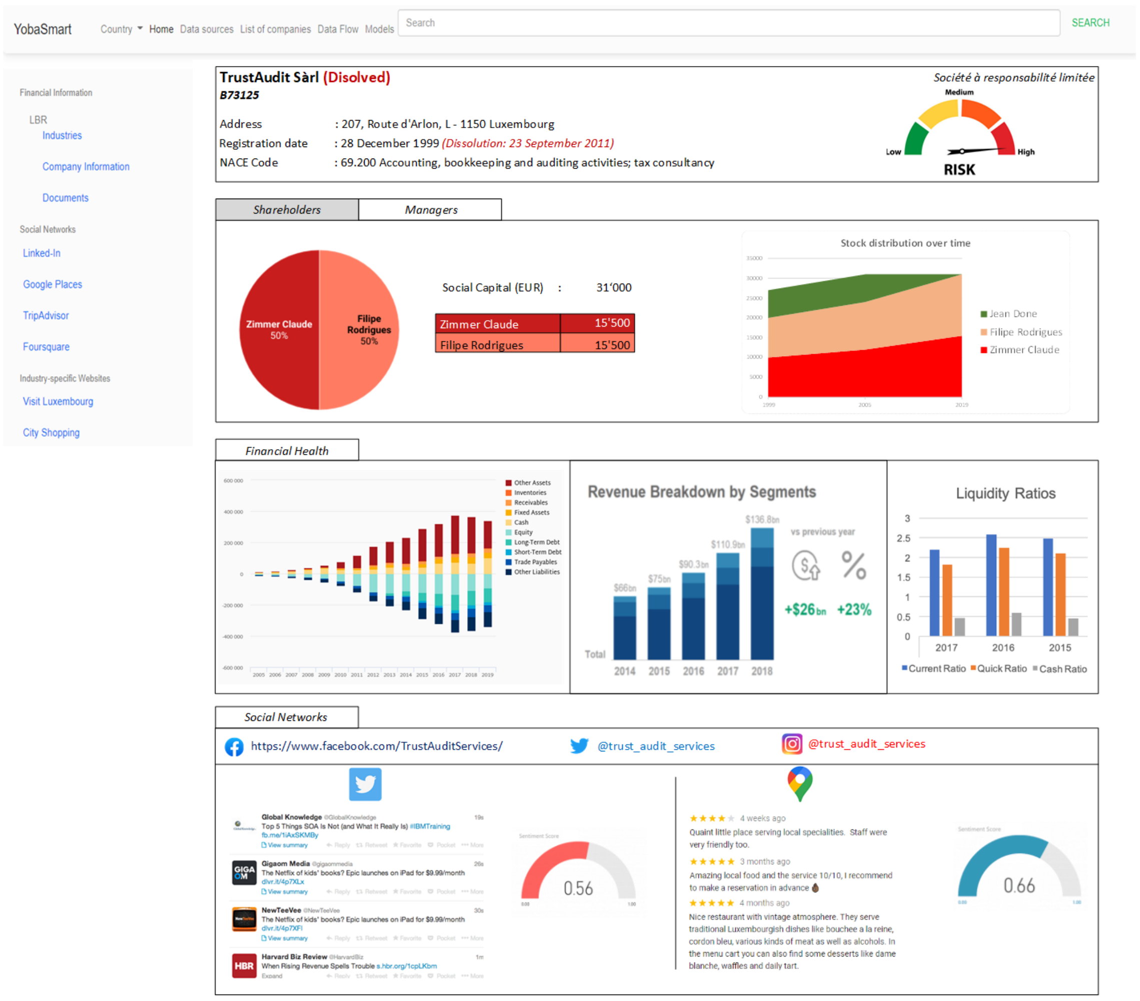
Based on the proof of concept and an analysis of the possibilities for the business dashboard, brainstorming meetings resulted in four groups of requirements. The first group, defined as the MVP was presented at the SNT Partnership Day held at the European Convention Center Luxembourg (ECCL) in Kirchberg, Luxembourg City, on May 7, 2024[6].

As illustrated in Figure 3.8, the web server is connected to the API server, which in turn is linked to both the cache server and the database server. Thus, two main components are identified for the design of the business dashboard.

7.2.3 Implementation, Testing and Deployment

The initial list of features and components required for the web server, API server, and database modifications was defined, though it has been continuously updated based on testing and incremental implementation of the Business Dashboard. The development of the web server was handled by a third-party company hired by the partner, while the API

[6] https://www.uni.lu/snt-en/partnership-day/



TrustAudit Sàrl (Disolved)
B73125

Address	: 207, Route d'Arlon, L - 1150 Luxembourg
Registration date	: 28 December 1999 *(Dissolution: 23 September 2011)*
NACE Code	: 69.200 Accounting, bookkeeping and auditing activities; tax consultancy

Social Capital (EUR)	:	31'000
Zimmer Claude		15'500
Filipe Rodrigues		15'500

Global Knowledge @GlobalKnowledge
Top 5 Things SOA Is Not (and What It Really Is) #IBMTraining fb.me/1AxSKMBy

Gigaom Media @gigaommedia
The Netflix of kids' books? Epic launches on iPad for $9.99/month dlvr.it/4p7XLx

NewTeeVee @NewTeeVee
The Netflix of kids' books? Epic launches on iPad for $9.99/month dlvr.it/4p7XFl

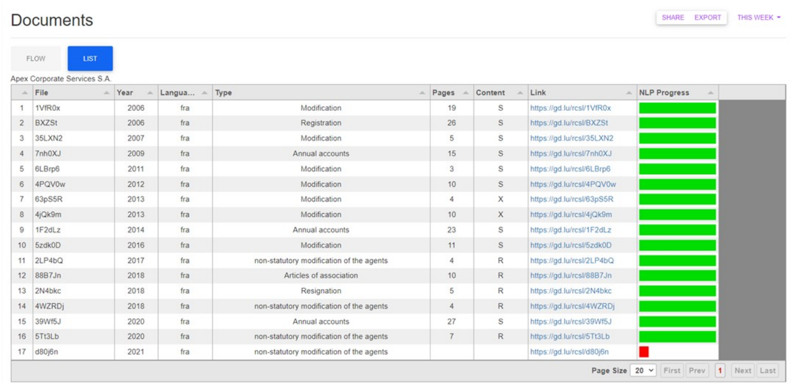
Harvard Biz Review @HarvardBiz
When Rising Revenue Spells Trouble s.hbr.org/1cpLKbm

★★★★ 4 weeks ago
Quaint little place serving local specialities. Staff were very friendly too.

★★★★★ 3 months ago
Amazing local food and the service 10/10, I recommend to make a reservation in advance

★★★★★ 4 months ago
Nice restaurant with vintage atmosphere. They serve traditional Luxembourgish dishes like bouchee a la reine, cordon bleu, various kinds of meat as well as alcohols. In the menu cart you can also find some desserts like dame blanche, waffles and daily tart.

Figure 7.6: Business Dashboard: Company Profile. The proposed company profile includes basic company information, financial charts, social network metrics, and an overall risk evaluation.



Documents

Apex Corporate Services S.A.

	File	Year	Langua...	Type	Pages	Content	Link	NLP Progress
1	1VfR0x	2006	fra	Modification	19	S	https://gd.lu/rcsl/1VfR0x	🟩
2	BXZSt	2006	fra	Registration	26	S	https://gd.lu/rcsl/BXZSt	🟩
3	35LXN2	2007	fra	Modification	5	S	https://gd.lu/rcsl/35LXN2	🟩
4	7nh0XJ	2009	fra	Annual accounts	15	S	https://gd.lu/rcsl/7nh0XJ	🟩
5	6LBrp6	2011	fra	Modification	3	S	https://gd.lu/rcsl/6LBrp6	🟩
6	4PQV0w	2012	fra	Modification	10	S	https://gd.lu/rcsl/4PQV0w	🟩
7	63pS5R	2013	fra	Modification	4	X	https://gd.lu/rcsl/63pS5R	🟩
8	4jQk9m	2013	fra	Modification	10	X	https://gd.lu/rcsl/4jQk9m	🟩
9	1F2dLz	2014	fra	Annual accounts	23	S	https://gd.lu/rcsl/1F2dLz	🟩
10	5zdk0D	2016	fra	Modification	11	S	https://gd.lu/rcsl/5zdk0D	🟩
11	2LP4bQ	2017	fra	non-statutory modification of the agents	4	R	https://gd.lu/rcsl/2LP4bQ	🟩
12	88B7Jn	2018	fra	Articles of association	10	R	https://gd.lu/rcsl/88B7Jn	🟩
13	2N4bkc	2018	fra	Resignation	5	R	https://gd.lu/rcsl/2N4bkc	🟩
14	4WZRDj	2018	fra	non-statutory modification of the agents	4	R	https://gd.lu/rcsl/4WZRDj	🟩
15	39Wf5J	2020	fra	Annual accounts	27	S	https://gd.lu/rcsl/39Wf5J	🟩
16	5Tt3Lb	2020	fra	non-statutory modification of the agents	7	R	https://gd.lu/rcsl/5Tt3Lb	🟩
17	d80j6n	2021	fra	non-statutory modification of the agents			https://gd.lu/rcsl/d80j6n	🟥

Page Size 20 First Prev 1 Next Last

Figure 7.7: Business Dashboard: Company documents. The proposed interface displays the document list, including the language, a public link to the document, and the processing progress percentage.

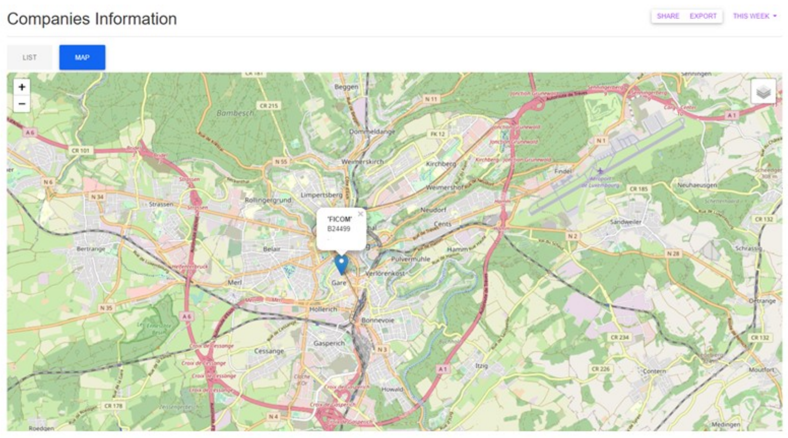


Figure 7.8: Business Dashboard: Company Location's map. This interface displays one or multiple addresses registered for a specific company.

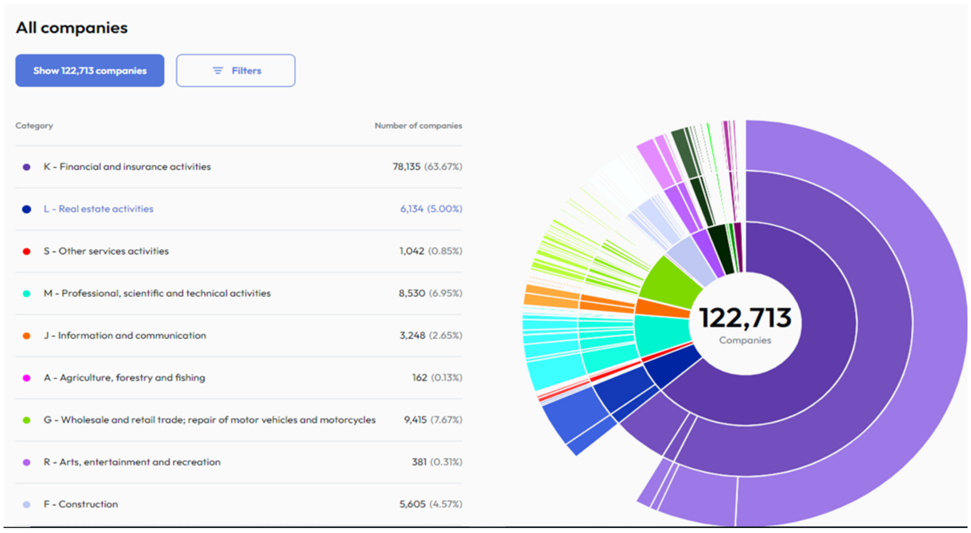
server was developed in collaboration with the partner company.

The API server was developed in C# using Entity Framework and integrated with a cache server to optimize performance. When a request is made, a hash key is generated from the API query payload (String) and used to query the cache server. If the same request has been made previously, the cached results are returned; otherwise, a query is performed on the dataset. Additionally, security measures are implemented at each component level to ensure data protection.

The solution was deployed in Azure, and users from the partner company began testing the application. Based on their feedback, a list of improvements was compiled and added to the feature backlog, which led to a reprioritization of the existing requirements.

7.3 Implemented Dashboard

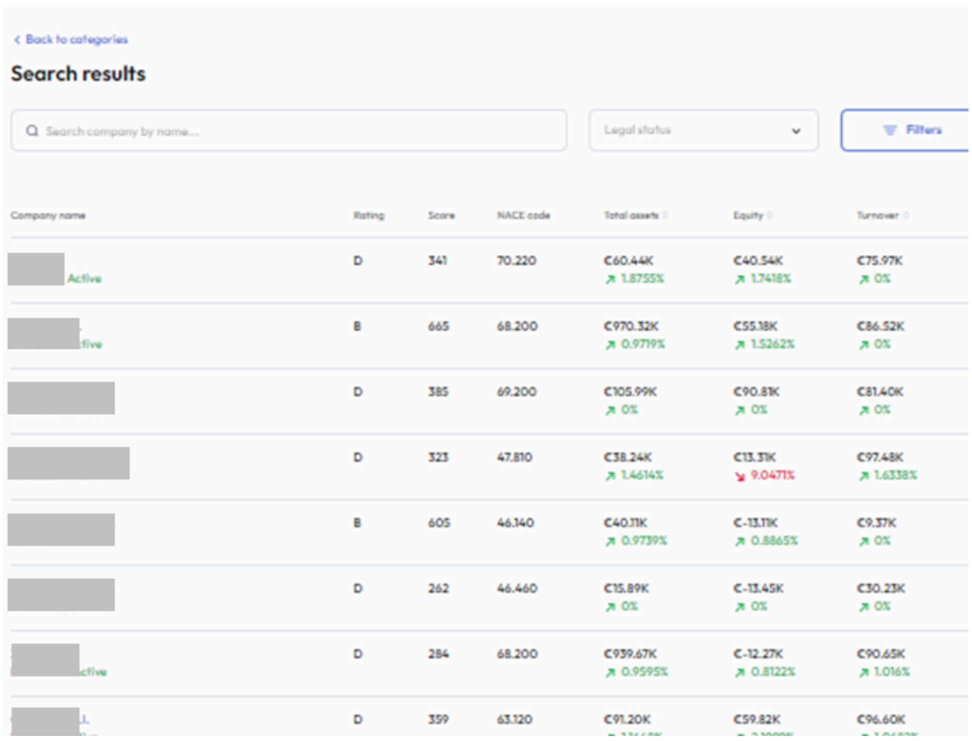
The following figures (Figure 7.9, Figure 7.10, Figure 7.11, Figure 7.12) show some screenshots of the implemented and deployed Client Dashboard, where is showed the industry chart, list of companies, score and rating and the detailed page view.



All companies

Category	Number of companies
● K - Financial and insurance activities	78,135 (63.67%)
● L - Real estate activities	6,134 (5.00%)
● S - Other services activities	1,042 (0.85%)
● M - Professional, scientific and technical activities	8,530 (6.95%)
● J - Information and communication	3,248 (2.65%)
● A - Agriculture, forestry and fishing	162 (0.13%)
● G - Wholesale and retail trade; repair of motor vehicles and motorcycles	9,415 (7.67%)
● R - Arts, entertainment and recreation	381 (0.31%)
● F - Construction	5,605 (4.57%)

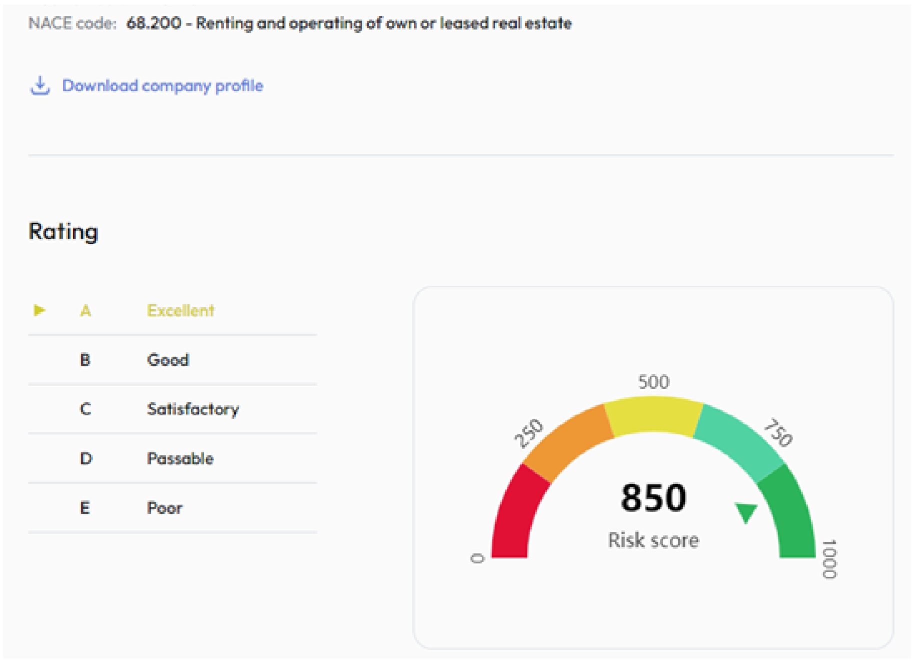
Figure 7.9: Implemented Business Dashboard. This figure shows the implemented and deployed industry chart.



Search results

Company name	Rating	Score	NACE code	Total assets	Equity	Turnover
[redacted] Active	D	341	70.220	€60.44K ↗ 1.8755%	€40.54K ↗ 1.7418%	€75.97K ↗ 0%
[redacted] tive	B	665	68.200	€970.32K ↗ 0.9719%	€55.18K ↗ 1.5262%	€86.52K ↗ 0%
[redacted]	D	385	69.200	€105.99K ↗ 0%	€90.81K ↗ 0%	€81.40K ↗ 0%
[redacted]	D	323	47.810	€38.24K ↗ 1.4614%	€13.31K ↘ 9.0471%	€97.48K ↗ 1.6338%
[redacted]	B	605	46.140	€40.11K ↗ 0.9739%	€-13.11K ↗ 0.8865%	€9.37K ↗ 0%
[redacted]	D	262	46.460	€15.89K ↗ 0%	€-13.45K ↗ 0%	€30.23K ↗ 0%
[redacted] ctive	D	284	68.200	€939.67K ↗ 0.9595%	€-12.27K ↗ 0.8122%	€90.65K ↗ 1.016%
[redacted] .L.	D	359	63.120	€91.20K	€59.82K	€96.60K

Figure 7.10: Implemented Business Dashboard. This interface shows the list of companies and some relevant ratios information in the first sight.



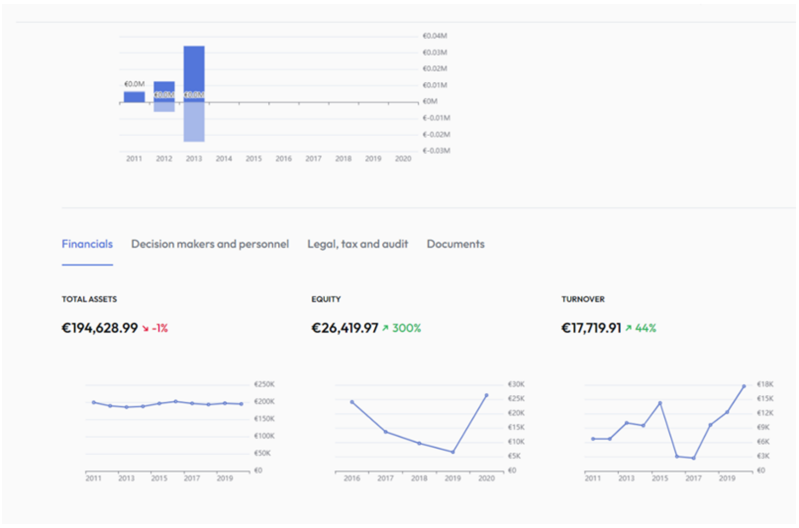
NACE code: 68.200 - Renting and operating of own or leased real estate

Download company profile

Rating

▶	A	Excellent
	B	Good
	C	Satisfactory
	D	Passable
	E	Poor

Figure 7.11: Implemented Business Dashboard. This interface shows the final risk rating calculated for the project and the corresponding score.



TOTAL ASSETS
€194,628.99 ↘ -1%

EQUITY
€26,419.97 ↗ 300%

TURNOVER
€17,719.91 ↗ 44%

€250K €200K €150K €100K €50K €0
2011 2013 2015 2017 2019
€30K €25K €20K €15K €10K €5K €0
2016 2017 2018 2019 2020
€18K €15K €12K €9K €6K €3K €0
2011 2013 2015 2017 2019

Figure 7.12: Implemented Business Dashboard. This interface shows the detailed information per company with information about financial ratios, decision makers, legal and audit partners and documents.

7.4 Summary

This section presented the Business Dashboard, a user interface designed to offer a user-friendly experience for interactive data exploration. Key components of the project included the proof of concept and the API services for the Minimum Viable Product (MVP), with the API services developed in partnership with the business partner.

Chapter 8

Results and Discussions

Contents

8.1 Introduction 153
8.2 Results 153
8.2.1 Data Gathering and Refining 153
8.2.2 Information Enrichment 154
8.2.3 Data Analysis and Risk Assessment 154
8.2.4 Data presentation and Exploitation 155
8.3 Discussion 155
8.4 Recommendations for Future Research 157

8.1 Introduction

This chapter presents the main results of the current research. The results are grouped into the corresponding data pipeline phases. Then is presented a brief discussion about the results and recommendation for the next steps.

8.2 Results

8.2.1 Data Gathering and Refining

For the first stage of the data pipeline, the research has been focused on the collection of various official data sources that are complementary. An OCR/HTML extractor tool was developed, which, using different existing Python tools, is able to extract text from both PDF-readable and scanned documents. The tool generates information that can be used to fine-tune a BERT-based model to automatically correct errors arising from the OCR process in images. The text extracted from these documents is cleaned and inserted into the relational database for use in subsequent stages. The primary data source is the Luxembourg Business Register (LBR), complemented by information from the List of NACE Codes from the European Commission, the list of auditors from the Institute of Business Auditors (IRE), and the list of notaries from the Chambre des Notaires of Luxembourg. The final dataset used in this study comprises 10,000 companies from 19 different industries. On average, each company has 10 documents, and these documents are predominantly in French (87.3%), German (6.6%), and English (5.1%).

8.2.2 Information Enrichment

The second stage of the data pipeline focuses on generating new information based on the extracted text from the data collection. A web application for labeling was developed, which displays the extracted text and provides functionality to view the original document and navigate through different document types. Using a fine-tuned BERT model and the content of each page from non-financial documents, page categories were predicted with approximately 97% accuracy. Additionally, a high-performing LLaMA 3.1 model was fine-tuned for extraction tasks, enabling the extraction of structured JSON information from unstructured and diverse formats with an accuracy of approximately 78% and no hallucination, while maintaining a consistent JSON output structure. Existing tools were utilized to convert textual addresses into geographical coordinates for use in subsequent stages. Lastly, data from various sources and documents were successfully unified, resolving discrepancies caused by OCR errors, typos, or variations in free-form text, allowing accurate identification of the same business entity across multiple sources. The accuracy of the proposed model for entities such as Companies and People was about 80%.

8.2.3 Data Analysis and Risk Assessment

The final stage of the data pipeline generation process focuses on generating meaningful insights that cannot be extracted using simple rules, aiming to uncover hidden information. An autoclustering algorithm was developed that utilizes the data from the OCR tool, creating a set of features based on document format and content. Clustering is then performed at the template level, followed by subclustering at the author's fingerprint level. Using the Rand Index as a clustering metric, the algorithm achieved approximately 87% accuracy for template-level clustering and around 55% for author's fingerprint or "hidden accountant" identification. Additionally, with the same long-BERT architecture, the risk of bankruptcy for companies was predicted using only the textual information from the annexes of the Annual Accounts, with a risk prediction precision exceeding 70%. The proposed multidimensional risk model takes information from the different datasources and models and classifies them into five dimensions, then using graphs, calculates the risk of each dimension using sub-dimension metrics. The proposed model is able to predict the risk of bankruptcy of a company with an F1-score of 89.7% and AUC of 89.2%.

8.2.4 Data presentation and Exploitation

To make the information and generated insights useful for users, a user web interface was created following an incremental and agile methodology. As a result, a user-friendly interface was released, allowing users to interact with the system and provide feedback for improvements and identifying opportunities for further enhancements.

8.3 Discussion

Based on the previous findings, publishing an anonymized dataset is currently not feasible due to GDPR requirements. While the Text Extraction tool can be employed to generate data for clustering analysis, it cannot be used for the initial step of anonymization. Ensuring that no sensitive data is shared remains a priority. The dataset must be anonymized as a whole, meaning that after extracting all the documents, a unique code must be assigned to each individual entity and each of its attributes to preserve meaning while maintaining privacy.

The workflow tool used for data pipeline implementation facilitates the control of various steps, from data collection to analysis and insight generation. This allows for a user-friendly web application interface. The tool also enables web scraping for new data sources, provided that interface emulation is not required. Each data flow is independent and can be automated using cron daemons or manually triggered. This underscores the importance of an IT dashboard for managing the integration of these diverse data flows.

Regarding trained or fine-tuned models for information enrichment, the fine-tuned BERT model processes segments of a document in parallel and integrates them using a concatenation layer. This approach outperforms the current state of the art for long text categorization, as it avoids the hallucination problems often seen in GPT-based models. It can handle more than 20 segments, representing approximately 8,000 tokens or 7,600 English words. Additionally, the fine-tuned LLaMA 3.1 model offers companies a lightweight solution that can run on a single GPU, extracting information in JSON format with accuracy comparable to leading subscription-based models like GPT-3.5. Moreover, it outperforms models like GPT-3.5 and Claude in reducing empty fields and adhering to the required two-level structure. Importantly, a method was proposed to generate synthetic data for sensitive information, enhancing the model's security and preventing data leakage.

In the final stage of the data pipeline, which focuses on generating new knowledge and insights, the auto-clustering algorithm is capable of identifying crucial elements such as hidden accountants. This is significant for detecting risky accountants and ensuring that all companies within the cluster are assessed with a lower risk for the networking dimension. Furthermore, the parallel long BERT architecture demonstrated the ability to predict bankruptcy risk, even when such risk was not explicitly mentioned in the annexes of the annual accounts.

Consolidating all of this information into a graph for clustering analysis offers a more comprehensive perspective on company risk, enabling a more thorough and multidimensional risk assessment. The data is categorized into five distinct dimensions or perspectives, integrating information from various data sources. While the inclusion of this additional information does not negatively impact the quality of bankruptcy predictionsin fact, it results in a slight improvementorganizing the data within these dimensions, along with the proposed importance weights for each dimension, provides decision-makers with enhanced insights to support more informed and effective decisions.

8.4 Recommendations for Future Research

This research project is considered a foundational step toward a multidimensional risk analysis. As such, there are numerous opportunities for improvement, particularly in the initial stage of the proposed data pipeline. It is recommended to gather unofficial data sources, such as LinkedIn, Twitter (X), news outlets, Facebook, Google Places, Foursquare, and TripAdvisor. Leveraging this information could allow for the inclusion of reputation as an additional dimension.

Additionally, advancements in OCR technologies, especially those based on GPT models, offer the potential to directly utilize or fine-tune newer models to correct OCR errors in scanned documents. Furthermore, enhancing the quality of the information extractor or developing a complex dataflow that integrates complementary technologies could enable the anonymization and publication of the dataset.

Another key improvement involves implementing an embeddings generator using a GPT-based model that retains context in its embeddings. This would facilitate the creation of a vector database for the entire dataset and support a contextual search engine tailored to the projects specific needs, mimicking Google's capabilities.

Incorporating temporal data into the graph will enable a more detailed analysis of a company's risk over time, as well as the relationships between individuals, locations, companies, and other entities. This addition has the potential to enhance the performance of predictive models. Furthermore, implementing a more sophisticated approach that directly integrates the graph with other modelssuch as combining GPT, BERT, and GNN architecturescould significantly improve prediction accuracy and overall model performance.

Lastly, the integration of an IT dashboard is recommended to serve as a centralized point for monitoring and managing the IT services and platform which currently is only in Dagster.

Chapter 9

Conclusions

The current thesis project proposes a new risk assessment framework for companies, with a focus on Small and Medium Enterprises (SMEs) in Luxembourg. It has been demonstrated that by utilizing also non-financial information from official public websites, along with Machine Learning tools, the existing data can be enriched to provide a broader perspective for decision-makers. The research involved the use and development of IT components, applying data science and engineering methods to integrate all developed components into a unified value chain for the end user.

As a result of this research, we have contributed to the state of the art with:

- Creation of a unified OCR tool that can be used for everyone to extract text information from PDF-Readable documents and scanned documents. Providing of more information such as page language, indicators if the page is encoded or not, scanned or not and other metadata information.
- A fine-tuned GPT-based model, built on LLaMA 3.1, has been developed to be installed on a single GPU server. It is capable of returning information in JSON format with performance comparable to GPT-3.5 and Claude 3 online services, and even surpasses them in terms of the quality of data retrieved, such as the reduction of empty fields.
- A new tokenization method has been developed for Entity Resolution, which is faster and more cost-effective compared to GPT-based methods, while effectively addressing OCR errors and typographical mistakes.

- We proposed a new process to add multiword entries to a tokenizer dictionary to reduce the number of generated tokens for domain-specific tasks in BERT-based models. Being able with this to increment the input information without the need of changing the model itself.

- The proposed autoclustering method has been designed to cluster documents based on their format and the author's fingerprint. This approach is useful for identifying risky clusters associated with high-risk accountants.

- Two architectures have been proposed to handle long text predictions or categorization using a BERT-based model. For text sequences of fewer than five pages, it is recommended that the input document be segmented and processed in parallel BERT instances, which are then summarized with two dense intermediate layers and integrated with an LSTM network, followed by two additional summarization layers. For text sequences exceeding five pages, or more than 8,000 English words or 16 pages, it is advised that the LSTM layers be replaced with concatenation layers.

- An integrated multidimensional risk assessment framework utilizing Graph Networks provides enhanced insights by aggregating data from diverse sources. The five proposed dimensions effectively predict the risk of bankruptcy, achieving an F1-score of 89.70%. This approach not only delivers high predictive accuracy but also equips decision-makers with comprehensive and actionable information.

Bibliography

A. Adhikari, A. Ram, R. Tang, and J. Lin. Docbert: Bert for document classification, 2019. URL https://arxiv.org/abs/1904.08398.

K. Adnan and R. Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771, 2019a. doi: 10.1177/1847979019890771. URL https://doi.org/10.1177/1847979019890771.

K. Adnan and R. Akbar. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, 6(1):1–38, 2019b.

K. Adnan and R. Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771, 2019c.

K. Adnan, R. Akbar, and K. S. Wang. Information extraction from multifaceted unstructured big data. *International Journal of Recent Technology and Engineering (IJRTE)*, 8:1398–1404, 2019.

M. S. Alkaissi H. Artificial hallucinations in chatgpt: Implications in scientific writing, 2023.

G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, 2015.

D. Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019. URL https://arxiv.org/abs/1908.10063.

E. C. Bank. Economic and monetary developments. small and medium-sized enterprises in the euro area: Economic importance and financing conditions. 2013. https://www.ecb.europa.eu/pub/pdf/other/mb201307_focus06.en.pdf.

I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL https://aclanthology.org/D19-1371.

O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18:255–276, 2009.

J.-G. Bernard, B. Aubert, S. Bourdeau, Éric Clément, C. Debuissy, M.-J. Dumoulin, M. Laberge, N. de Marcellis-Warin, and I. Peignier. Le risque : un modèle conceptuel d'intégration. CIRANO Project Reports 2002rp-16, CIRANO, Oct. 2002. URL https://ideas.repec.org/p/cir/cirpro/2002rp-16.html.

S. Bhatore, L. Mohan, and Y. Reddy. Machine learning techniques for credit risk evaluation: a systematic literature review. *J BANK FINANC TECHNOL 4*, 2020. doi: https://doi.org/10.1007/s42786-020-00020-3.

W. Bi, B. Xu, X. Sun, Z. Wang, H. Shen, and X. Cheng. Company-as-tribe: Company financial risk assessment on tribe-style graph with hierarchical graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 27122720, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539129. URL https://doi.org/10.1145/3534678.3539129.

O. Binette and R. C. Steorts. (almost) all of entity resolution. *Science Advances*, 8 (12):eabi8021, 2022.

B. C. Blanco Lambruschini and M. Brorsson. Transforming unstructured sensitive information into structured knowledge. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 831–838, 2024a.

B. C. Blanco Lambruschini and M. Brorsson. A novel architecture for long-text predictions using bert-based models. In K. Arai, editor, *Intelligent Systems and Applications*, pages 105–125, Cham, 2024b. Springer Nature Switzerland. ISBN 978-3-031-66428-1.

B. C. Blanco Lambruschini, P. Becerra-Sanchez, M. Brorsson, and M. Zurad. Reducing tokenizers tokens per word ratio in financial domain with t-mufin bert tokenizer. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 94–103, 2023a.

B. C. Blanco Lambruschini, M. Brorsson, and M. Zurad. Auto-clustering of financial reports based on formatting style and author's fingerprint. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 112–127, Cham, 2023b. Springer Nature Switzerland. ISBN 978-3-031-23633-4.

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

I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. Legal-bert: The muppets straight out of law school, 2020. URL https://arxiv.org/abs/2010.02559.

M.-Y. Chen, M.-Y. Chen, and M.-Y. Chen. Predicting corporate financial distress based on integration of decision tree classification and logistic regression. *Expert Systems With Applications*, 2011. doi: 10.1016/j.eswa.2011.02.173.

K. Choe and S. Garas. The graph theoretical approach to banruptcy prediction. *The Journal of Accounting and Management*, 2021.

V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys (CSUR)*, 53(6):1–42, 2020.

J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

C. Clement et al. Machine learning in bankruptcy prediction–a review. *Journal of Public Administration, Finance and Law*, (17):178–196, 2020.

C. Crovini, G. Ossola, and B. Britzelmaier. How to reconsider risk management in smes? an advanced, reasoned and organised literature review. *European Management Journal*, 39(1):118–134, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2018.

M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang. Distributed representations of tuples for entity resolution. *Proc. VLDB Endow.*, 11 (11):14541467, July 2018. ISSN 2150-8097. doi: 10.14778/3236187.3236198. URL https://doi.org/10.14778/3236187.3236198.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

I. E. Fisher, M. R. Garnsey, and M. E. Hughes. Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, 23(3):157–214, 2016.

Q. Gao, M. Lin, and R. W. Sias. Words matter: The role of texts in online credit markets. *Journal of Financial and Quantitative Analysis, forthcoming*, 2018.

R. Geng, I. Bose, and X. Chen. Prediction of financial distress: An empirical study of listed chinese companies using data mining. *European Journal of Operational Research*, 2015. doi: 10.1016/j.ejor.2014.08.016.

Q. Grail, J. Perez, and E. Gaussier. Globalizing bert-based transformer architectures for long document summarization. In *Proceedings of the 16th conference of the European chapter of the Association for Computational Linguistics: Main volume*, pages 1792–1810, 2021.

Y. Guo, J. Liu, W. Tang, and C. Huang. Exsense: Extract sensitive information from unstructured data. *Computers Security*, 102:102156, 2021. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2020.102156. URL https://www.sciencedirect.com/science/article/pii/S0167404820304296.

H. Hamza, Y. Belaid, A. Belaid, and B. B. Chaudhuri. An End-to-End Administrative Document Analysis System. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 175–182.

S. Herath, M. Roughan, and G. Glonek. Generating name-like vectors for testing large-scale entity resolution. *IEEE Access*, 9:145288–145300, 2021. doi: 10.1109/ACCESS.2021.3122451.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

ISO-31000. risk management - a practical guide for smes. 2015. https://www.iso.org/publication/PUB100367.html.

B. Jehangir, S. Radhakrishnan, and R. Agarwal. A survey on named entity recognitiondatasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017, 2023.

Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu. Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172, 2020. doi: 10.1080/23270012.2020.1756939.

A. Kim and S. Yoon. Corporate bankruptcy prediction with domain-adapted bert. In *EMNLP 2021, 3rd Workshop on ECONLP*, 2021.

LBR. Luxembourg business registry. https://www.lbr.lu/mjrcs-lbr/jsp/webapp/static/mjrcs/en/mjrcs-lbr/pdf/faq.pdf.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL https://doi.org/10.1093/bioinformatics/btz682.

Y. Liu. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318, 2019. URL http://arxiv.org/abs/1903.10318.

F. Mai, S. Tian, C. Lee, and L. Ma. Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, 274(2):743–758, 2019. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2018.10.024. URL https://www.sciencedirect.com/science/article/pii/S0377221718308774.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

E. B. Musanovic and S. Halilbegovic. Financial statement manipulation in failing small and medium-sized enterprises in bosnia and herzegovina. *Journal of Eastern European and Central Asian Research*, 2021. doi: 10.15549/jeecar.v8i4.692.

D. L. Olson, D. Delen, and Y. Meng. Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, 52(2):464–473, 2012. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2011.10.007. URL https://www.sciencedirect.com/science/article/pii/S0167923611001709.

G. Papadakis, M. Fisichella, F. Schoger, G. Mandilaras, N. Augsten, and W. Nejdl. Benchmarking filtering techniques for entity resolution. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 653–666, 2023. doi: 10.1109/ICDE55515.2023.00389.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, 2018.

J. Qu, W. Hua, D. Ouyang, and X. Zhou. An efficient and effective approach for multi-fact extraction from text corpus. *World Wide Web*, pages 1–24, 2022.

Y. Qu, P. Quan, M. Lei, and Y. Shi. Review of bankruptcy prediction using machine learning and deep learning techniques. *Procedia Computer Science*, 162:895–899, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.12.065. URL https://www.sciencedirect.com/science/article/pii/S1877050919320769. 7th International Conference on Information Technology and Quantitative Management (ITQM 2019): Information technology and quantitative management based on Artificial Intelligence.

A. Radford. Improving language understanding by generative pre-training. 2018.

W. M. Rand. Objective criteria for the evaluation of clustering methods. 66(336): 846–850, 1971. doi: 10.1080/01621459.1971.10482356.

B. Ribeiro, N. Chen, and A. Kovacec. Shaping graph pattern mining for financial risk. *Elseiver Neurocomputing*, 2019.

P. Rust, J. Pfeiffer, I. Vuli, S. Ruder, and I. Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models, 2020. URL https://arxiv.org/abs/2012.15613.

R. Sawhney, A. Wadhwa, S. Agarwal, and R. R. Shah. Fast: Financial news and tweet based time aware network for stock trading. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2164–2175, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.185. URL https://aclanthology.org/2021.eacl-main.185.

F. Sovrano, K. Ashley, and A. Bacchelli. Toward eliminating hallucinations: Gpt-based explanatory ai for intelligent textbooks and documentation. In *CEUR Workshop Proceedings*, number 3444, pages 54–65. CEUR-WS, 2023.

StanfordNLP. Openie, 2020. URL https://stanfordnlp.github.io/CoreNLP/openie.html.

R. Stefko, J. Horvathova, and M. Mokrisova. The application of graphic methods and the dea in predicting the risk of bankruptcy. *Journal of Risk and Financial Management*, 2021.

J. Tang, Y. Zuo, L. Cao, and S. Madden. Generic entity resolution models. In *NeurIPS 2022 First Table Representation Workshop*, 2022.

T. M. Toprak Ahmet. Enhanced named entity recognition algorithm for financial document verification. *The Journal of Supercomputing*, 2023. doi: 10.1007/s11227-023-05371-4.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

X. Wang, Z. Kräussl, M. Zurad, and M. Brorsson. Effective automatic feature engineering on financial statements for bankruptcy prediction. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–8. IEEE, 2023.

A. Wong. Corporate sustainability through non-financial risk management. *Corporate Governance*, 14(4):575–586, 2014.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 267–273. Association for Computing Machinery. ISBN 978-1-58113-646-3.

D. Yao, Y. Gu, G. Cong, H. Jin, and X. Lv. Entity resolution with hierarchical graph attention networks. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 429442, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392495. doi: 10.1145/3514221.3517872. URL https://doi.org/10.1145/3514221.3517872.

B. Yu, Z. Zhang, J. Li, H. Yu, T. Liu, J. Sun, Y. Li, and B. Wang. Towards generalized open information extraction, 2022.

G. Zaman, H. Mahdin, K. Hussain, and A. Rahman. Information extraction from semi and unstructured data sources: A systematic literature review. *ICIC Express Letters*, 14(6):593–603, 2020.

R. Zhang, Z. Wei, Y. Shi, and Y. Chen. {BERT}-{al}: {BERT} for arbitrarily long document understanding, 2020. URL https://openreview.net/forum?id=SklnVAEFDB.