Master of Science
Master's thesis in Human Centered Artificial Intelligence

DTU Compute
Department of Applied Mathematics and
Computer Science

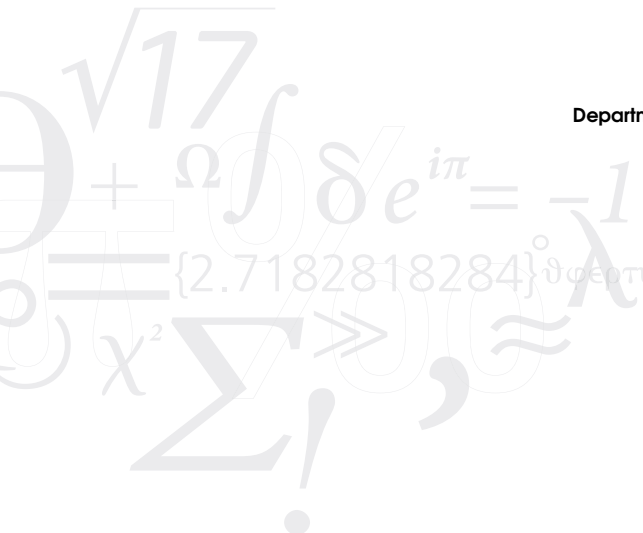
Analysts vs. Algorithms: Evaluating LLMs' Ability to Assess Credit Risk Like Human Analysts

A Master's thesis

Author: Pierre Høgenhaug

DTU

Supervisor: Nicki Skafte Detlevsen (DTU)
Co-supervisor: Nicolai Wiingreen (Capital Four)



DTU Compute

Department of Applied Mathematics and Computer Science

Technical University of Denmark

Richard Petersens Plads

Building 321

2800 Kongens Lyngby, Denmark

Abstract

This thesis explores how large language models (LLMs) can help detect fundamental risks in corporate bond prospectuses, a lengthy and sometimes cumbersome source of information for credit risk assessment. Traditionally, analysts read through hundreds of pages to identify warnings about an issuer's operations, market conditions, or industry challenges. As firms invest in a wide variety of bonds, maintaining a consistent review of each prospectus becomes increasingly time-consuming.

To address this challenge, we develop a proof-of-concept system that parses full prospectuses—including not just the "Risk Factors" section but also other areas like Business, Management, and Capitalization. We then apply an LLM-based pipeline to pinpoint specific negative risk factors, aligning its outputs with the existing "fundamental scores" used by credit analysts. We find that the model reliably flags all of the risks human experts highlight (achieving perfect recall), yet it also generates a fair amount of false positives. Providing brief "reference cases" in a second evaluation step reduces these false positives but does not eliminate them entirely.

Overall, our results suggest that an LLM-driven review can save analysts time by ensuring no critical risks go unnoticed, while also requiring a secondary check to filter out routine or boilerplate mentions. We discuss how fine-tuning prompts, incorporating specialized training data, and refining the model's reference examples could further improve accuracy. By automating the most labor-intensive parts of reading bond prospectuses, this approach holds promise for more efficient, transparent, and scalable credit risk analysis.

Contents

Part I

Theoretical Concepts and Data

CHAPTER 1

Introduction

In asset management, financial analysts generate returns for investors by identifying the most promising investment opportunities in the market. Financial analysts often carry out this task by reading and researching corporate documents, which is time-consuming given the large volume of data. As asset managers grow and diversify across different markets and industries, the number of documents and factors to track also increases. Human analysts, no matter how skilled, face natural limits in how many pages they can review and interpret each day.

Recent developments in technology especially with Large Language Models (LLMs), offer new ways to process and understand written text. These models can quickly scan large documents, figure out key points, and highlight possible risk factors. In this thesis, we focus on a proof-of-concept system for identifying specific risk factors in bond prospectuses. Today at Capital Four, a Danish credit asset management firm, human analysts assign "Fundamental Scores" to each issuer, paying special attention to risk factors like industry challenges or technology risk. Currently, these scores rely heavily on an analyst's personal reading and judgment. The goal of this work is to see if we can automate part of that process, using LLMs to read prospectuses and identify relevant risks. We hope that this automation can make the review process faster, more consistent, and less dependent on each analyst's reading style.

Before discussing the technical details, we first explain why LLMs are particularly relevant to credit risk assessment. Traditional methods of evaluating credit risk especially for high-yield bonds are labor-intensive. Bond prospectuses usually amount to several hundred pages, mixing legal, financial, and industry-specific jargon. We explore how LLMs' contextual understanding, combined with prompt engineering, may improve consistency and reduce the manual workload.

To provide a roadmap of this thesis: the next section (Section 1.1) clarifies our specific objectives and constraints. We then review relevant literature (Section 1.2) and lay out the theoretical foundations (Chapter 2), focusing on why bond prospectuses are central to credit risk evaluation and how LLMs can enhance text analysis. Subsequent chapters describe the Dataset and Data Processing pipeline, explaining how we gather and structure real-world prospectuses and then introduce our Experimental Methodology for evaluating LLM-based detection of risk factors. We conclude with the Experimental Results and final reflections on strengths, limitations, and potential

future work.

1.1 Scope

The main goal of this thesis is to design, build, and test a proof-of-concept technology that automates the detection of a specific risk factor in bond prospectuses. From the company's perspective, this proof of concept shows how LLMs can assist in screening long financial documents for fundamental risks, saving time and resources for analysts.

From an academic perspective, the thesis experiments with an LLM-based pipeline to analyze real-world text data and compares its performance with analyst-validated ground truth.

Even though the dataset and assumptions may not be perfect for a full production solution, this project lays the foundation for future work, where analysts can improve data quality, refine model prompts, and expand the approach to other risk factors. With these boundaries clarified, the next section positions this thesis within the broader context of automated text processing in finance and credit risk.

1.2 Related Work

1.2.1 Automated Text Processing in Finance

Early efforts to apply NLP in finance relied on rules-based methods and small datasets, but the field expanded significantly with the emergence of deep learning and large-scale corpora. This evolution is illustrated by surveys that show how finance advanced from basic text parsing for sentiment toward more sophisticated pipelines that incorporate machine learning and neural architectures [1, 2]. Beyond simple sentiment extraction, Ke et al. [3] propose a supervised-learning approach for detecting sentiment terms predictive of asset returns.

Meanwhile, readability and sentiment remain focal points in analyzing financial documents, especially in bond markets. Less readable annual reports correspond to higher credit spreads, trade costs, and volatility [4]. Lower prospectus readability is linked to higher issuance spreads in Chinese bond offerings [5]. Overly optimistic language can skew bond risk premiums, underscoring the complexity of issuer disclosures [6]. Overall, studies confirm that even relatively simple textual features—readability and tone—can strongly correlate with market outcomes.

Alongside these topic-specific investigations, systematic reviews detail how cloud-based NLP pipelines can classify and cluster large amounts of financial data [7]. Domain-specific corpora are becoming more common, as illustrated by REFinD for financial relation extraction [8]. ChatGPT-based methods in finance have been shown to automate tasks like reporting and risk analysis [9]. Collectively, these works highlight how automated text processing now plays a central role in financial data analytics, offering coverage far beyond what manual reading can achieve.

1.2.2 Large Language Models in Credit Risk Assessment

Large Language Models (LLMs) such as GPT variants have recently gained attention for their potential in credit risk applications. Generative AI is being explored to accelerate underwriting, monitor portfolios, and standardize credit decisions, though scalability and data-quality issues remain key barriers [10]. "Generalist credit scoring" with LLMs has been described as capable of outperforming classical models across multiple tasks while also introducing new biases if not calibrated properly [11]. Broader surveys of LLM research in finance underscore domain-specific adaptation strategies, performance benchmarks, and critical challenges such as interpretability and scalability [12, 13].

ChatGPT and GPT-4 have been shown to handle tasks like credit classification, although specialized pretraining may still excel in narrow contexts [14]. Concerns regarding reasoning and transparency in high-stakes decisions like credit approvals have been pointed out [9, 15]. It is further noted that while LLMs have significantly boosted NLP coverage in finance—from sentiment analysis to compliance monitoring—thorough evaluation and domain alignment remain critical [2]. Together, these studies position LLMs as a promising approach for credit risk analysis while emphasizing the need for bias mitigation, regulatory compliance, and transparent deployment.

1.2.3 Challenges in Applying NLP to Regulatory and Legal Documents

Applying NLP to regulatory or legal texts introduces unique difficulties, including specialized jargon and complex statutory structures. The effort needed to convert dense regulations into structured rules, often requiring domain experts to produce controlled English statements, has been underscored [16]. Automated reviews can miss subtle legal obligations in financial advice [17]. Integrating phrase-structure grammars with domain ontologies can improve extraction accuracy in construction

regulations, but this also highlights the labor-intensive nature of building ontologies [18].

In the context of bond prospectuses, an NLP-based decision-support system for Germany's central bank has been developed to automate the classification of eligibility criteria, reducing the time analysts spend manually reviewing prospectuses [19]. Still, bridging the gap between raw text processing and domain-specific rule interpretation remains challenging. The scarcity of complex–simple parallel corpora in legal text simplification, and the failure of off-the-shelf simplification methods on intricate legal writing, have been noted [20]. Overall, these studies show that finance-specific and legal knowledge must be deeply integrated with technical NLP methods for reliable results.

1.2.4 Comparisons with Traditional Manual Approaches

Many processes in finance—from risk factor identification to compliance checks—are still performed through human reading. Recent studies comparing automated or LLM-based approaches to manual baselines find that GPT-4 can equal or outperform humans in specific classification tasks [21] and that ChatGPT frequently matches human labels in stance detection and sentiment analysis [22]. However, prompt variations and multiple runs can lead to inconsistencies [23]. In financial advice compliance, although automation handles more data faster, analysts must watch for edge cases [17]. These findings suggest that while LLM-based pipelines can substantially reduce workload, they should be paired with human oversight to address accuracy and bias in critical decisions like credit evaluations or regulatory compliance.

1.2.5 Concluding Remarks and Gap

Taken together, these four subsections highlight how NLP and LLM methods have evolved for financial data, the specialized challenges in regulatory/legal texts, and the importance of human oversight. They also confirm a key gap that underlies our thesis: the difficulty of thoroughly scanning high-yield bond prospectuses to identify relevant risks in a reliable and scalable manner. In the pages ahead, we propose to address this gap with an LLM-based pipeline for automating risk factor detection. We now turn to Chapter 2, where we lay out the foundations of credit risk analysis and detail how LLMs' textual capabilities may fit into an automated workflow.

CHAPTER 2

Theory

In modern asset management, the shift from manual reviews to computer-based processing is transforming how investment risks are identified and managed. Capital Four faces the challenge of systematically extracting information that fits its internal scoring system. To address this, the next sections explain how to integrate large language models (LLMs) into the credit risk assessment pipeline. We begin by providing an overview of corporate bonds and their related disclosure documents. We then highlight Capital Four's approach to risk assessment and the potential for automation. Finally, we discuss the theoretical background of LLMs particularly their ability to handle context-rich, domain-specific texts [24,25] and connect these capabilities to the dataset, data processing, and experimental methodology detailed in later chapters.

2.1 Corporate Bonds and Credit Risk Assessment

Corporate bonds represent debt securities that companies issue to raise capital (see e.g. Investopedia). Investors effectively lend money to the issuer with the expectation of receiving periodic coupon payments and principal repayment at maturity. Unlike stocks, where shareholders hold ownership stakes, bondholders rely on contractual obligations spelled out in the bond's terms, interest rates, maturities, and covenants to protect their investments. When evaluating these terms, analysts must consider both quantitative and qualitative factors, from the issuer's financial health to overall market conditions.

A key part of bond investing is measuring credit risk, or the likelihood that an issuer might default on its payments. Credit rating agencies like Standard & Poor's and Moody's offer a preliminary lens on this risk by assigning letter-grade ratings. Investment-grade bonds (BBB- or higher) are considered less likely to default but offer lower yields. In contrast, high-yield or "junk" bonds compensate investors with higher coupons to offset their elevated risk of default (Investopedia). Historically, however, these ratings have not captured all risks [26]. Thus, asset managers, including Capital Four, conduct their own in-depth credit risk assessments to identify risks

that ratings may miss.

2.2 Corporate Bond Prospectuses

2.2.1 Prospectus Requirements and Global Frameworks

Corporate bond prospectuses are legal documents that provide details about a bond's features and the issuer's financial health (Investopedia). These documents are often long, running hundreds of pages, and must follow the regulatory requirements of the issuer's region. In Europe, the Prospectus Regulation (EU) 2017/1129 sets out the required structure and content of these disclosures (see ESMA's Prospectus Regulation page), supervised by the European Securities and Markets Authority (ESMA). In the United States, the Securities and Exchange Commission (SEC) offers similar guidelines through the Securities Act of 1933 and the Securities Exchange Act of 1934.

2.2.2 The Risk Factors Section

The "Risk Factors" section of a prospectus typically describes potential threats that could affect the issuer's ability to make its payments such as operational challenges, market volatility, or industry cycles. For asset managers like Capital Four, who specialize in high-yield bonds, accurately identifying risk factors is especially critical. Nevertheless, traditional approaches remain labor-intensive and time-consuming, raising the potential for inconsistencies, missed details, and subjective biases. These issues are precisely where modern NLP methods, such as LLMs, may help.

2.2.3 Beyond Risk Factors: Other Relevant Sections

Although the Risk Factors section is the primary source for identifying issuer-specific threats, other parts of the prospectus can also provide valuable context for a risk assessment. Sections on Business, Management and Governance, Capitalization, and Legal Details may reveal hidden vulnerabilities or crucial background about the company and its sector. This thesis thus considers the entire prospectus rather than only the "Risk Factors" section when scanning for textual clues about fundamental risks. For an idea of the typical sections included in a bond prospectus, refer to Figure 2.1.

Typical Prospectus Table of Contents

1. **Preliminary**
   a) TABLE OF CONTENTS
   b) SUMMARY
   c) IMPORTANT NOTICE
2. **Offering Details**
   a) THE OFFERING
   b) USE OF PROCEEDS
   c) PLAN OF DISTRIBUTION
   d) DESCRIPTION OF THE NOTES
3. **Company Overview**
   a) BUSINESS
   b) MANAGEMENT
   c) CAPITALIZATION
   d) INDEPENDENT AUDITORS
   e) LISTING AND GENERAL INFORMATION
4. **Risk & Legal Considerations**
   a) RISK FACTORS
   b) LEGAL MATTERS
   c) TRANSFER RESTRICTIONS
   d) NOTICE TO INVESTORS
   e) CERTAIN RELATIONSHIPS & RELATED PARTY TRANSACTIONS
   f) CERTAIN ERISA CONSIDERATIONS
   g) LEGAL ADVISORS TO THE INITIAL PURCHASERS
5. **Market Measures**
   a) STABILIZATION

Figure 2.1. *An example layout of a bond prospectus, showing major sections such as Risk Factors, Offering Details, and Management.*

2.3 Capital Four's Investment Approach

Capital Four is an asset manager that manages approximately 19 billion EUR for institutional investors worldwide. Its strategies include fixed-income instruments such as high-yield bonds, leveraged loans, private debt, multi-asset credit, and collateralized loan obligations (CLOs). Capital Four's investment philosophy relies on fundamental credit analysis, ESG integration, and sector-specific expertise.

A key element of Capital Four's strategy involves repeatable assessment frameworks, such as their Fundamental Score, which ranks bonds on parameters like market dynamics, structural trends, and issuer stability. While some scores, like ESG metrics, come from numerical data, the Fundamental Score often depends on subjective judg-

ments based on the text from the bond prospectus.

2.3.1 The Challenges of Manual Fundamental Scores

Since Capital Four's credit portfolios cover multiple industries and regions, manually maintaining accurate risk assessments quickly becomes resource-intensive. Even with a robust analyst team, ensuring consistent and timely detection of risk parameters can be difficult. Differences in legal terminology, industry-specific language, and document structure add complexity. Hence, the firm faces two major challenges:

1. **Scalability:** Manually reading and annotating prospectuses takes time, limiting the number of companies that can be monitored.
2. **Consistency:** Subjectivity in interpreting qualitative text can cause inconsistencies between analysts, making it harder to compare risk profiles across issuers.

These challenges motivate our work: to explore if LLMs can make identifying specific risk factors more efficient and consistent for Capital Four's Fundamental Scores.

2.4 Toward an Automated Assessment Framework

To address the limitations of manual prospectus reviews, this thesis proposes using transformer-based neural models (LLMs) trained on large amounts of text. The automation pipeline is meant to work with analysts, not replace them. By quickly scanning prospectus sections for potential red flags, LLMs can highlight or prioritize areas that need further human review.

2.4.1 Connection to the Dataset and Data Processing

This approach relies on building a robust and representative dataset of bond prospectuses. Chapter 3 ("Dataset") describes how these documents were collected from Capital Four's SharePoint, regulatory websites (e.g., ESMA), and third-party providers (e.g., FinDox). Chapter 4 ("Data Processing") explains how each PDF was converted into structured text while preserving headings and subheadings for context.

2.4.2 Alignment with Capital Four's Fundamental Scores

LLMs must map prospectus text to the same risk parameters that Capital Four's analysts currently use. Each risk parameter becomes a yes/no question (e.g., "Does the text indicate this issuer is vulnerable to rapid technological shifts?"). The pipeline's results are then validated against the firm's ground truth data.

2.5 Large Language Models

Large Language Models (LLMs) are advanced neural networks, typically based on the Transformer architecture, that generate text with strong contextual understanding. They are trained on large text datasets, which helps them learn linguistic patterns and even some domain-specific terms without extensive retraining.

2.5.1 Key Advancements in LLMs

The move from earlier rule-based systems to LLMs is based on two key breakthroughs:

1. **Self-Attention Mechanism:** Transformers can dynamically focus on the most relevant parts of an input sequence, capturing both local and long-range dependencies in text more effectively than older recurrent or convolutional architectures.

2. **Pretraining at Scale:** Training on massive text datasets lets LLMs learn language structures and even domain-specific terms. As a result, these models can adapt to new tasks with little extra tuning, often using "few-shot" prompting.

These properties are very useful in financial text analysis. Financial documents, such as bond prospectuses, include industry-specific jargon and complex legal language. Traditional NLP tools often have difficulty with this complexity, requiring a lot of manual tuning or specialized lexicons. In contrast, LLMs are pre-trained on diverse sources including news articles, regulatory filings, and legal documents which makes them more familiar with financial terms. This makes them well-suited for extracting insights without extensive customization.

2.5.2 LLMs in Credit Risk Assessment

Recent studies demonstrate LLM success in domain-specific tasks like legal clause extraction, identifying compliance gaps in regulatory texts, and performing sentiment analysis on earnings call transcripts. In a credit risk context, an LLM can detect clues in the text about a company's reliance on key suppliers, exposure to volatile input costs, or potential legal disputes. By consistently identifying these references, whether they appear in the formal "Risk Factors" section or buried elsewhere, an LLM-based pipeline can reduce omissions and expedite the process. Prompting strategies like chain-of-thought prompting, which encourages the model to explain its reasoning step by step, and few-shot prompting, which provides a few annotated examples, can further improve accuracy and clarity [25, 27].

2.5.3 Prompt Engineering and Few-Shot Learning

An interesting aspect of LLMs is their responsiveness to targeted "prompt engineering". By designing prompts carefully, one can guide the model to produce better outputs. For example, "chain-of-thought" prompts encourage the model to explain its reasoning step by step, which helps improve clarity and trust in the results.

Few-shot learning is another useful technique. It involves giving the model a few examples, either real or synthetic, that show what a particular risk factor looks like. These examples help the model understand what counts as a real red flag versus a normal mention. In this thesis, we use such reference cases to show what a present or missing risk parameter looks like in practice. The experimental methodology (Chapter 5) explores how these examples can improve detection performance.

2.5.4 LLM Consistency and Reliability

Despite the growing enthusiasm surrounding Large Language Models, recent research highlights that these models may produce variable outputs across repeated prompts, especially when dealing with complex or ambiguous questions [28, 29]. This variability can be caused by random sampling in the model, small changes in prompt phrasing, and alignment mechanisms that adjust the model's tone or content based on instructions.

From a methodological standpoint, work by Gallo et al. [28] shows how repeat prompting can produce correlated responses rather than truly independent samples, highlighting the need to manage variability in LLM outputs carefully. Moore et al. [29] further

demonstrate that LLMs' consistency can differ by topic, with more controversial subjects leading to more inconsistent answers. Both studies suggest that it is important to evaluate an LLM's stability by repeating queries or using different paraphrases.

In this thesis, we address LLM consistency by running multiple experiments under identical conditions and measuring agreement with Fleiss' Kappa. This statistic provides insight into how reliably the model identifies specific risk factors in bond prospectuses from run to run. If the model often gives different labels for the same text, it may not be effective in production without further calibration (such as adjusting temperature or fine-tuning). On the other hand, high agreement (a high Kappa score) suggests that the model could be a reliable risk-screening tool. Along with the reproducibility strategies proposed by Kim et al. [30], these measures help ensure that our LLM-based pipeline can be validated consistently, reinforcing the reliability of its predictions in credit risk assessment.

2.6 Summarizing the Theory

In summary, credit risk analysis is fundamental to bond investing, especially in high-yield markets. Meanwhile, LLMs have proven capable of handling context-rich, domain-specific documents, making them strong candidates for automated text analysis. By leveraging self-attention and large-scale pretraining, these models can detect subtle risk signals in lengthy regulatory filings. In the following chapters, we test how well LLMs can support credit risk detection in real prospectuses. The next chapter (Chapter 3) describes the documents used to validate these ideas.

CHAPTER 3

Dataset

This chapter provides an overview of the data used in this study, bridging the theory section and the data processing steps that follow in Chapter 5. We begin by describing the sources of the bond prospectuses, as well as Capital Four's proprietary Fundamental Score data. Next, we show how we combined these data streams into one dataset for Large Language Model (LLM) analysis. Finally, we discuss how well the dataset represents different industries, regions, and time periods.

3.1 European Bond Prospectus

3.1.1 Data Origin and Scope

We built a complete collection of European bond prospectuses using several data sources from Capital Four's system data ecosystem. The Capital Four **SharePoint Portal** served as the primary repository, where each company's documents are stored under a unique internal company ID. This repository contained all available corporate bond prospectuses and was scraped using a custom Python script (`scrape_sharepoint.py`). Any missing prospectuses were retrieved from the European Securities and Markets Authority (ESMA) website. If the Capital Four SharePoint did not have a prospectus, we scraped it from the **European Securities and Markets Authority** (ESMA) website using a custom Python script (`scrape_esma.py`). This served as a supplementary source to ensure comprehensive coverage. When prospectuses could not be automatically linked between FinDox and Capital Four's data, we had to extract them manually. The mapping uses asset IDs, and if no match is found, it uses strict string matching. However, small differences in naming or formatting can stop the automatic linking. For these companies, we manually checked the FinDox website to find the correct prospectuses. In most cases, we could identify the correct documents even without automatic mapping, so they were included in the dataset.

3.1.2 Risk Factors Section

Every prospectus has a "Risk Factors" section that lists potential threats affecting the issuer's ability to repay its debt. Using the PDF-to-text process described in Chapter 4, each "Risk Factors" section was extracted and segmented into discrete entries, each corresponding to a named or sub-named risk factor. Metadata linking ensures that each entry retains information about the Capital Four Company ID, the original document name, and the publication date.

3.1.3 Beyond Risk Factors: Other Key Sections (Dataset)

While the "Risk Factors" section is important, other parts of a bond prospectus also provide insights into an issuer's risks. For example, Business and Industry sections may show vulnerabilities like relying on one product or being exposed to fast-changing technology. Management and Governance sections can highlight key-person risk, and Capitalization and Use of Proceeds sections often show a company's financial structure or debt refinancing plans. These extra sections widen the view of risk, offering a more complete assessment than just looking at the "Risk Factors" section. By noting these risks, analysts can spot "red flags" that may not appear in the "Risk Factors" section. In Figure 2.1, we gave an overview of typical sections in a bond prospectus. Below is a brief overview of what information the common sections in bond prospectuses contain.

Business and Industry-Related Sections

Sections like *Business* and *Industry and Market Data* shed light on the company's core operations, revenue drivers, and competitive landscape. They frequently disclose details about product lines, target markets, barriers to entry, and emerging industry trends.

Management and Governance

The *Management* section typically discusses the experience, track record, and structure of the executive team and board of directors.

Capitalization and Use of Proceeds

Sections such as *Capitalization* and *Use of Proceeds* reveal the issuer's financial structure and strategic intentions behind the bond issuance.

Legal and Transactional Details

Prospectus parts like *Legal Matters*, *Transfer Restrictions*, and *Certain Relationships and Related Party Transactions* may point to potential legal or regulatory entanglements that could impact a firm's operations or reputation.

Other Relvant Disclosures

Lastly, sections like *Plan of Distribution*, *Description of the Notes*, and *Description of Certain Financing Arrangements* offer information about underwriting strategies, covenant structures, and future obligations that might influence a company's financial flexibility.

Taken together, these extra sections widen the view of risk, offering a more complete assessment than just looking at the "Risk Factors" section. Beyond the formal "Risk Factors" segment, they offer richer context for understanding a company's strategic posture, managerial structure, financial health, and industry dynamics. Incorporating insights from these parts of the prospectus can therefore enhance an LLM's ability to detect and categorize fundamental risks, ultimately leading to a more robust and nuanced investment analysis.

3.2 Fundamental Score Data

Capital Four uses a proprietary Fundamental Score metric to assess 7 key areas:

1. Business Model
2. Competitive Positioning
3. Management & Ownership
4. Intra-Industry Competition
5. Market Dynamics
6. Regulatory Framework
7. Technology Risk

to form an overall credit opinion.

Each company gets a score from 1 to 5 (1 is best, 5 is worst). In addition, analysts add short text descriptors, or "parameters", to note details like exposure to cyclical markets, reliance on a niche product, or vulnerability to demographic shifts.

In this thesis, we focus on whether a risk parameter is present (a negative exposure) instead of the numerical score. While the 1-5 scale is important for asset management, our goal is to see how well LLMs can spot fundamental risks by converting

the analysts' text descriptors into a binary risk label. For instance, an analyst might note the parameter "Is the industry susceptible to rapid technological advances or innovations?" under the broader category of "Technology Risk". If the analyst thinks the issuer is susceptible, the parameter is marked (indicating a risk). Otherwise, it is omitted, indicating minimal or mitigated risk.

Each of the seven fundamental risk categories in Capital Four's framework (e.g., Technology Risk, Regulatory Risk, etc.) has two to five specific parameters. Each parameter is labeled as positive or negative. For this research, we focus on negative indicators as signs of risk. Focusing on negative parameters provides a clear target for model evaluation. In the future, positive parameters could be included to assess strengths and improve risk assessment.

3.3 Ground Truth Labels

The ground truth labels used in this thesis are derived from the text-based descriptors appended to each Fundamental Score record. When an analyst marks a particular fundamental risk parameter (e.g., "Industry susceptible to rapid technological innovation"), that parameter is stored in the database and thereby signals that the risk is present. If no label is attached, we assume that risk is absent.

In our final ground truth dataset, each company's text parameters are converted to a binary outcome for each risk:

- **Present (1):** The analyst flagged the risk parameter, indicating a negative exposure.
- **Absent (0):** The analyst did not attach the risk parameter.

By linking these binary labels to each prospectus, we create a clear way to evaluate the LLM. If the LLM flags a risk that matches an analyst's label, it is a true positive; if it flags a risk not noted by the analyst, it is a false positive. This structure sets the stage for the precision and recall metrics discussed in Chapter 5.

3.4 Reference Cases

In addition to the ground truth labels, Capital Four keeps reference cases (mini-case studies) that show when and how each risk parameter appears in real situations. These reference cases come in pairs (negative example vs. positive/mitigated exam-

ple) and are used internally to ensure that new or less experienced analysts identify consistent signals across different issuers.

In the first part of our experiments, we test the LLM's ability to detect risk parameters without using these examples—a naive approach with a simple prompt. In the second part, we give the LLM some reference cases as few-shot examples, showing clear examples of what counts as a risk. To give the reader an example, we provide one of the raw reference cases (Industry Competition - a: tendency to price rationally) in Figure 3.1. This approach lets us measure if the reference cases improve the LLM's ability to identify fundamental risks in new prospectuses.

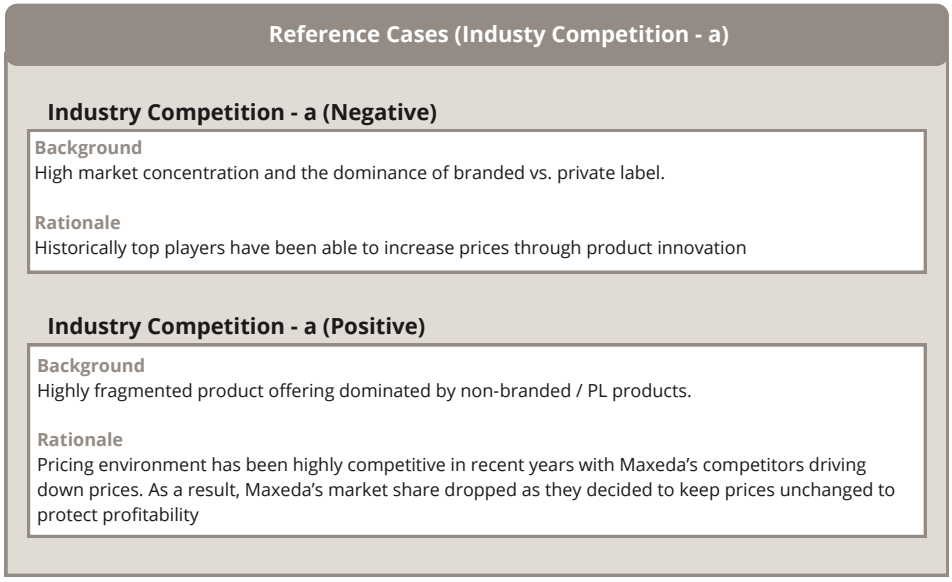


Figure 3.1. *An example of an untouched reference case (Internal document from Capital Four).*

Since some of these reference cases contain proprietary information and mention specific companies in Capital Four's portfolio, they had to be anonymized for the purposes of this thesis. In certain cases, the underlying scenarios were lightly altered, but the central characteristics such as technological disruption in the industry or a major operational dependency were preserved. This approach ensured adherence to the non-disclosure agreement (NDA) in place while still conveying relevant context for the LLM to learn from.

3.5 Dataset Assembly

After merging the bond prospectus data with the Fundamental Score records, our final dataset comprises the following:

- **Number of Companies:** 171 unique companies (selected from an initial pool of 228).
- **Time Span:** Documents span from 2007 to 2024, with 2021 appearing most frequently.
- **Geographical Distribution:** Prospectuses originate from 20 countries, with the highest representation from the US (63), the UK (26), and Germany (15).
- **Industries:** The dataset covers 16 distinct industries, ensuring broad sectoral diversity; the most frequent sectors include Basic Industries (21), Services (21), and Retail (16).
- **Risk Factor Entries:** On average, each prospectus contains about 73 risk-factor entries, totaling 12,483 entries across all documents.
- **(Sub)Subsection Entries:** Each prospectus averages 441 pages, with roughly 795 (sub)subsections per document, amounting to 135,945 entries in total.

Having compiled all sections from each prospectus, we now describe how this data is cleaned, parsed, and prepared in the next chapter on Data Processing.

CHAPTER 4
Data Processing

In this chapter, we turn bond prospectus PDF files into a structured format that LLMs can handle more easily. By keeping the original headings and hierarchy, we preserve important context. We also link each prospectus to Capital Four's "fundamental scores" database so we have a ground truth for whether a particular risk factor actually applies to a company. These steps prepare both the "Risk Factors" section and other relevant parts for automated analysis. As shown in Figure 4.1, the pipeline starts with raw PDFs, merges them with fundamental score data, and adds reference cases for few-shot examples to create the final dataset.

4.1 PDF Parsing

The data processing starts by collecting bond prospectuses in PDF format and converting them into parseable text. Initially, we focused on extracting the "RISK FACTORS" section because of its standard format and direct link to Capital Four's risk assessment. However, we found that other sections like Business and Industry, Management and Governance, Capitalization and Use of Proceeds, Legal and Transactional Details, and financial arrangements also include important risk information.

To capture all these sections systematically, we developed a single Python script, `parse_prospectus.py`, that applies a unified parsing pipeline to each PDF. The script also handles errors, checks the text accuracy, and flags documents that need manual review.

4.1.1 Parsing Workflow Overview

1. **File Identification:** Each PDF is uniquely associated with a company in Capital Four's database using an internal ID. The script checks a designated folder for valid PDF files, ignoring duplicates or files known to be corrupted.

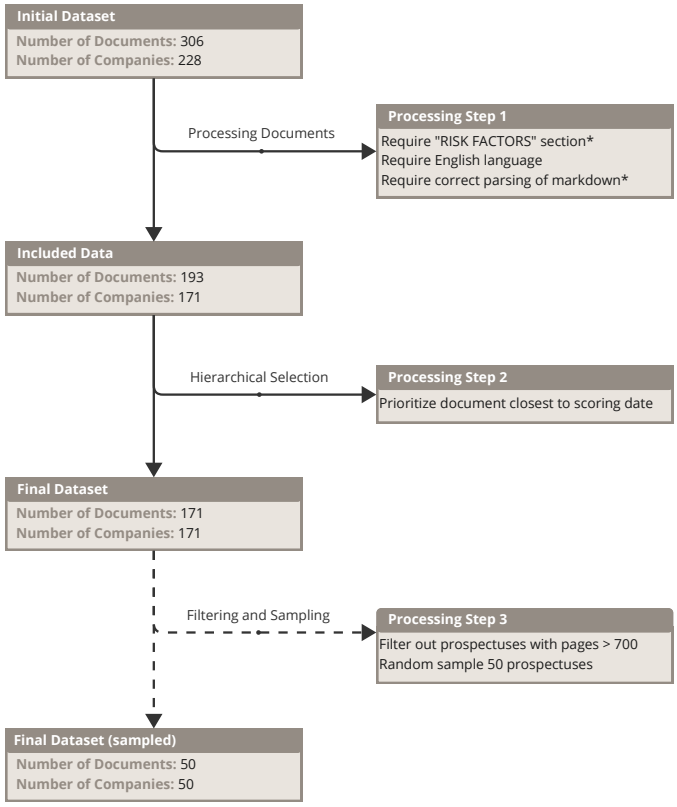


Figure 4.1. *High-level diagram showing how raw PDFs, Fundamental Score data, and reference cases flow through each processing step to produce a structured dataset for LLM analysis.*

2. **Language Detection:** The script detects the text language to avoid non-English documents. Non-English documents are flagged and set aside.
3. **Exception Handling:** If the parser finds malformed PDFs or unexpected formatting, it logs the error and skips the file. Partially parsed outputs are discarded to prevent errors.

4.1.2 PDF-to-Text Conversion

Once the script identifies a valid PDF, it transforms the document's contents into a human-readable Markdown representation to capture both text and the basic formatting cues used to distinguish headings, subheadings, and emphasized text. This

conversion is done using the `pymupdf4llm` library and works as follows:

1. **Page-by-Page Extraction:** Each PDF page is parsed to extract its raw text and any associated style markers (e.g., **bold**, *italic*).
2. **Markdown Structuring:** The text is saved as a Markdown file, keeping visual cues like headings (**HEADING**), subheadings (**Subheading**), and bold-italic markers (**_Sub-subheading_**). Compared to raw text output, Markdown allows us to keep a simplified version of the document's original hierarchy.
3. **Global Formatting Rules:** For non-standard formatting (like tables, footnotes, or disclaimers), the script uses regular expressions and line-merging to maintain text flow.
4. **Inclusion of All Sections:** While the "RISK FACTORS" section remains vital, the parser also keeps headings from other prospectus sections (e.g., Business, Management, Capitalization, etc.). This ensures that any text with potential relevance to fundamental risks does not get lost simply because it is outside the formal risk disclosures.

With these steps, each prospectus first becomes a single Markdown file, and later a Pandas DataFrame, which is easier to search, process, and use with LLMs.

4.1.3 Hierarchical Text Extraction

After converting the PDF documents to Markdown, the parsing script reassembles the text into a nested hierarchy of sections, subsections, and (in some cases) sub-subsections. This helps preserve the context of each text block while standardizing different formatting styles.

1. **Identifying Major Sections:** The parser scans for headings in uppercase bold (e.g., **SECTION TITLE**) to define top-level sections. For example, the script seeks "RISK FACTORS" explicitly but also flags any other uppercase bold headings such as "BUSINESS," "MANAGEMENT," or "CAPITALIZATION."
2. **Subsection Detection:** Within each top-level section, the parser looks for bold headings that are not in all-caps (e.g., **Subsection Title**) to create subsections. This captures segmentations like "Reliance on Key Customers" or "Senior Management Team," which can fall under the broader categories mentioned above.
3. **Sub-Subsection Handling:** A bold-italic heading pattern (e.g., **_Additional Considerations_**) is treated as a sub-subsection. While more common

in risk disclosures, where certain sub-risks are broken out, similar headings occasionally appear in other sections (e.g., disclaimers or footnotes).

4. **Text Normalization:** Lines with the same formatting are merged to avoid breaking up a discussion. This helps combine bullet points, lists, or broken paragraphs into coherent text blocks.

4.1.4 Structured Output

Finally, once the script finishes parsing and organizing the text, it stores the results in a structured tabular format. Each row in the output corresponds to the smallest coherent text unit in many cases, a subsection or sub-subsection body.

1. **Metadata Fields**

   Every row is tagged with identifying information, such as:

   - **Prospectus ID** (unique code that is linked to internal Company ID)
   - **Original Filename**
   - **Section and Subsection Titles**
   - **Hierarchy Levels** (e.g., Section ID, Subsection ID)
   - **Error Flags** (e.g., language mismatch or missing “RISK FACTORS”)

2. **Body Text**

   The parser stores the combined text from each subsection or sub-subsection in a column called ”Subsubsection Text”. This keeps the complete content of that segment for LLM analysis.

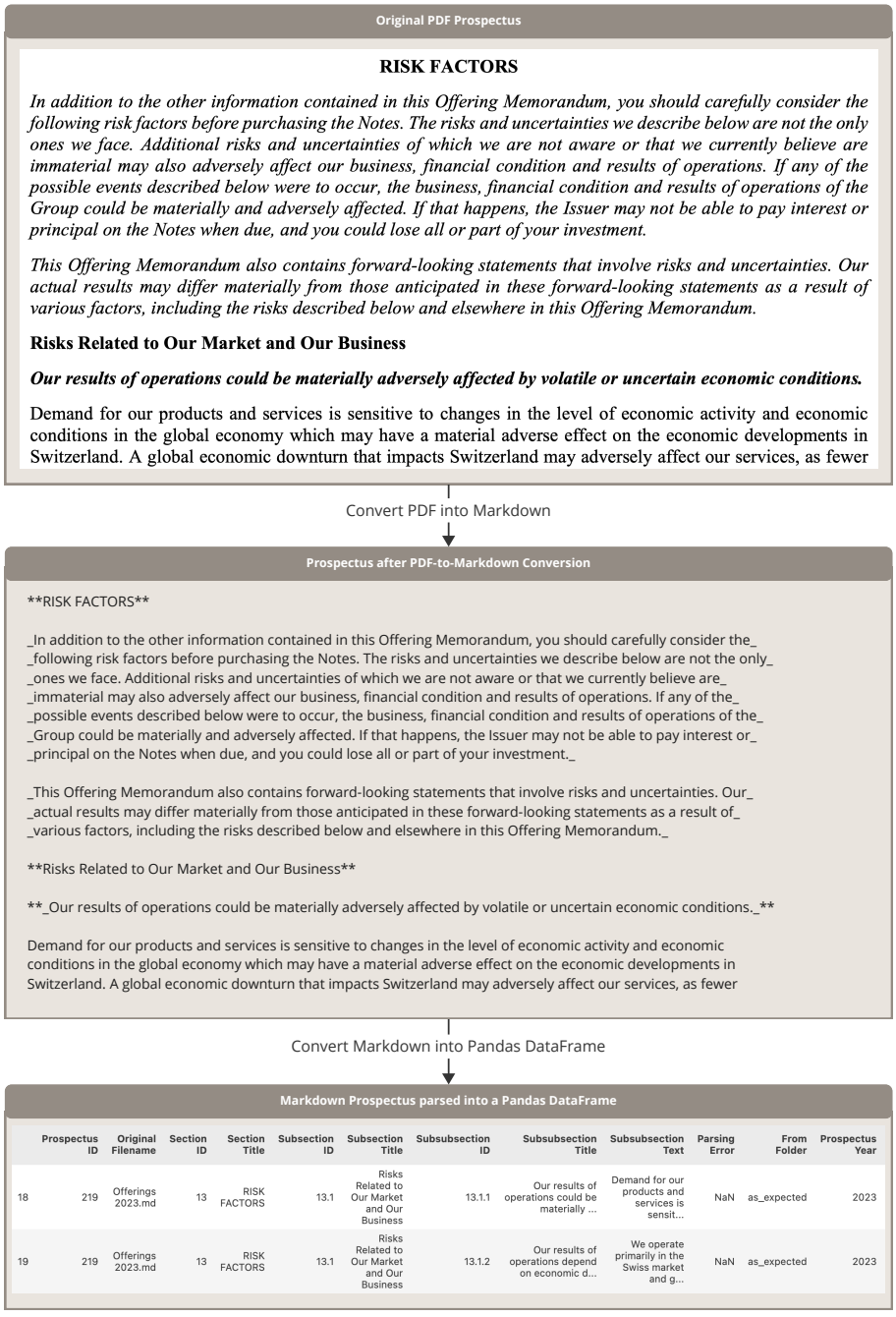
3. **Handling Non-standard Sections**

   If the script finds unexpected headings (like disclaimers or footnotes in bold uppercase), it still records them, so no content is lost.

4. **Integration with ”Ground Truth”**

   By storing the text for all relevant sections (not just “RISK FACTORS”), we can later match each subsection with the analyst-assigned labels from Capital Four’s Fundamental Score. This lets us measure how well an LLM identifies fundamental risks inside and outside formal risk disclosures.

Figure 4.2 shows the step-by-step process from PDF to Markdown to hierarchical text blocks, capturing each section while keeping context for LLM analysis.



	Prospectus ID	Original Filename	Section ID	Section Title	Subsection ID	Subsection Title	Subsubsection ID	Subsubsection Title	Subsubsection Text	Parsing Error	From Folder	Prospectus Year
18	219	Offerings 2023.md	13	RISK FACTORS	13.1	Risks Related to Our Market and Our Business	13.1.1	Our results of operations could be materially ...	Demand for our products and services is sensit...	NaN	as_expected	2023
19	219	Offerings 2023.md	13	RISK FACTORS	13.1	Risks Related to Our Market and Our Business	13.1.2	Our results of operations depend on economic d...	We operate primarily in the Swiss market and g...	NaN	as_expected	2023

Figure 4.2. *End-to-end procedure converting each PDF into hierarchical text blocks (sections, subsections), preserving context for LLM analysis.*

4.1.5 Advantages of a Unified Approach

- **Consistency:** Using a single script (`parse_prospectus.py`) for all sections eliminates the need for multiple parsers or manual tweaks.
- **Scalability:** New prospectuses can be added by placing their PDFs in the source folder. The same steps work no matter the formatting differences.
- **Robustness:** The script's hierarchical logic also captures risk indicators in sections like Management, Governance, or Capitalization, which can be as important as the "RISK FACTORS" section.

This unified parsing process creates a complete text dataset for LLM analysis. Including all major sections and keeping the document hierarchy gives us a full view of each company's risk factors, which is key for testing if LLMs can detect fundamental risks similarly to human analysts.

4.2 Fundamental Scores

While the Fundamental Score in its original form is a 1-to-5 rating, our workflow filters and reshapes that information to focus on the textual "risk parameter". We load the raw data from Capital Four's SQL repository and clean it by removing unneeded columns (see `clean_fundamental_score` in `data_processing.py`). The cleaned output keeps only the fields needed to identify specific risk exposures, along with the unique internal identifiers linking each row to the correct company.

A key step was to consolidate these text descriptors and match them to a standardized risk parameter (e.g., "Susceptible to technological advances"). In practice, working with financial analysts would refine these descriptors to match the intended risk signals. Since this thesis is a proof of concept, our parameter definitions remain broad; we focus on testing if the LLM can detect whether a risk is present or not.

4.2.1 Fundamental Score Labels as Targets

After we isolate the relevant parameters, we convert them into binary targets suitable for classification. Each risk parameter is given a label indicating its **presence (1)** or **absence (0)**. For example:

- *Technology Risk: Industry is susceptible to technological advances, Negative* $\longrightarrow$ Translated as `Technology_Risk_a = 1`.
- *Regulatory Risk: Significant exposure to changing legislation, Negative* $\longrightarrow$ Translated as `Regulatory_Risk_b = 1`.

Where no analyst note exists, or a parameter is irrelevant for a particular issuer, we assign a 0. We also turn these negative parameters into short, question-like statements to help the LLM understand. For example, the original descriptive label "Exposure to rapid technological change" is reformulated into, "Does the text indicate that the industry is susceptible to rapid technological advances or innovations?". Though these reformulations are intentionally simple, in a production setting, financial analysts would guide more nuanced phrasing to capture the exact conditions under which a risk is signaled. The full transformation is summarized in Figure 4.3, showing how raw Fundamental Scores and analyst notes become our final ground truth labels.

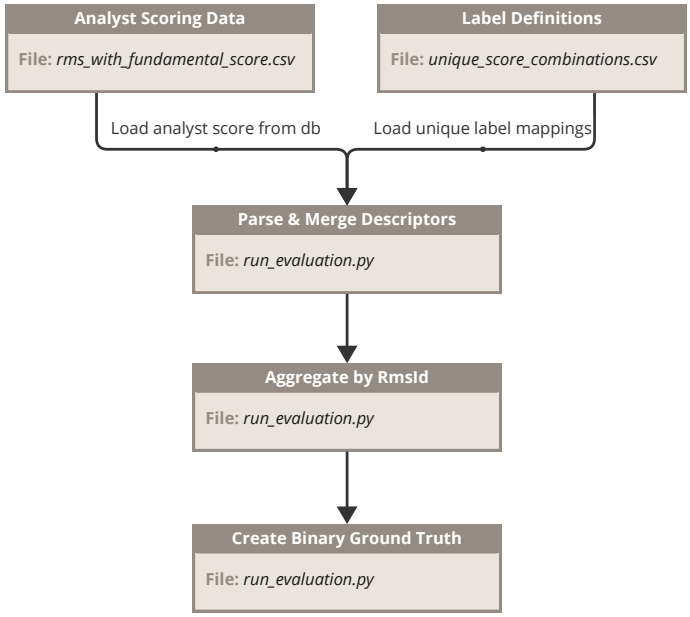


Figure 4.3. *Illustration of how raw analyst scores are transformed into binary labels (risk present or absent) for use in model evaluation.*

We note that each of the seven risk categories can include positive parameters (e.g., "Robust technological infrastructure"), but these are outside our current scope. We focus on negative parameters because they represent downside risk. Studying the positive aspects could be a natural extension for future research if one wished to evaluate how well LLMs can also identify and categorize a company's strengths.

4.3 Reference Cases

Within `data_processing.py`, we treat the reference cases similarly to how we manage Fundamental Scores, but we label them as example scenarios. Each reference case is manually linked to one of the seven risk categories and marked as either a negative example (risk present) or a positive example (risk mitigated or not applicable). These short narratives are stored in a structured Python docstrings for easy insertion in few-shot prompts using Python f-strings (see f-strings introduction).

Since some reference cases contain proprietary details (like a real issuer's name or specific strategies), we anonymized or slightly altered them to meet company privacy requirements. Each anonymized case keeps the key risk context (such as *market susceptibility to disruption*) without revealing sensitive details. In Figure 4.4 we illustrate how an analyst's reference case is turned into a few-shot prompt example for the LLM.

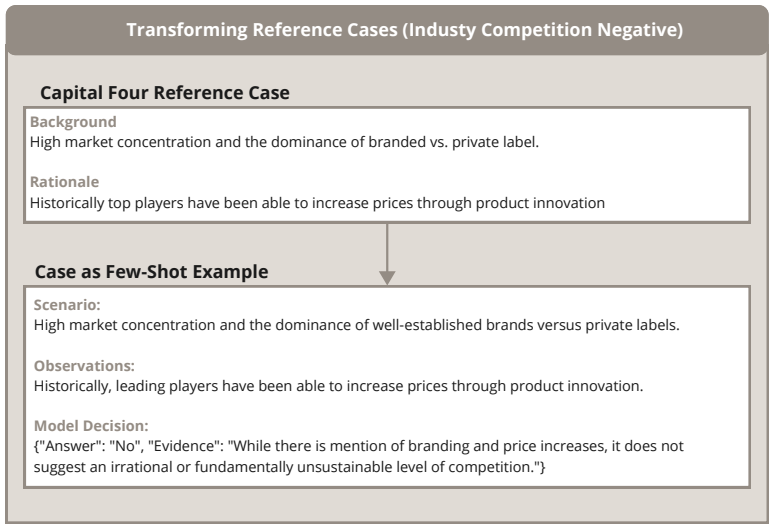


Figure 4.4. *An example showing how a full analyst reference case (top) is reformulated (bottom) for inclusion in the LLM's few-shot prompt.*

In the second part of this thesis, the LLM uses these reference cases as few-shot examples before scanning a new prospectus subsection. Comparing the model's performance with and without reference cases helps us see if examples improve risk detection.

4.4 Final Sampled Dataset

After merging the bond prospectus data with the Fundamental Score records and randomly selecting 50 companies, our final dataset includes:

- **Number of Companies:** 50 unique companies (from an initial pool of 228).
- **Time Span:** Documents range from 2012 to 2024, with 2021 appearing most frequently.
- **Geographical Distribution:** Prospectuses come from 13 countries, with the highest representation from the US (27), Germany (6), and the UK (4).
- **Industries:** The dataset covers 12 industries, the most frequent being Basic Industries (8), Technology & Electronics (6), and Capital Goods (6).
- **Risk Factor Entries:** On average, each prospectus contains about 73 risk-factor entries, totaling 12,483 entries across all documents.
- **(Sub)Subsection Entries:** Each prospectus averages 441 pages, with roughly 795 (sub)subsections per document, amounting to 135,945 entries in total.

With this structured text and linked risk labels, we now explain how we use the data in our LLM-based risk detection experiments.

Summary of Differences from the Dataset Assembly

In Section 3.5, we described our complete dataset that combined data from all available prospectuses and Fundamental Score records, covering 171 companies with a wider time period, more regions and industries. In contrast, the *Final Sampled Dataset* is a smaller, focused subset that randomly selects 50 companies. This selection maintains the overall diversity of the original dataset while making it easier to manage for our LLM risk detection experiments.

Part II

Experimental Methodology and Results

CHAPTER 5

Experimental Methodology

This chapter outlines how we tested Large Language Models (LLMs) in detecting fundamental risk factors within corporate bond prospectuses. We begin by explaining our rationale for using prospectuses as the primary data source, including why they align well with the firm's existing risk labels. We then describe how we segmented each document into subsections, grouped subsections by issuer, and compared model outputs against ground truth. Next, we detail our choice of model configurations, including an exhaustive (or explorative) approach to scanning every subsection of each prospectus for all risk categories. We also discuss why we tested intra-annotator reliability re-running the same pipeline multiple times with identical prompts to verify that the model's labels are stable rather than random. Finally, we present two approaches to risk detection: a single-pass "naive" prompt versus a two-pass method that includes few-shot reference examples to refine the model's initial classifications.

5.1 Rationale and Data Source

We chose the bond prospectus as our primary data source for two key reasons:

1. **Most Objective Available Document:** Among issuer disclosures we evaluated, the bond prospectus stands out for its regulatory rigor and structured format. It is a more objective document than, for example, internal memos or analyst notes, because it has to pass legal checks and is organized according to standardized guidelines (e.g., ESMA in the EU). In principle, this makes the prospectus an especially consistent and transparent source of risk-related information.
2. **Strict, Structured Layout:** Although bond prospectuses can be lengthy, their high-level sections (e.g., Risk Factors, Business, Management, Capitalization) generally appear in similar order and follow formal headings or subheadings. As noted in Section 4.1, this internal structure simplifies text parsing,

which in turn made it easier to convert entire prospectuses into a segmented dataset that LLMs could process.

It is important to note that our emphasis on prospectuses was not driven by analysts having previously linked their risk labels directly to these documents. Rather, we needed a reliable, systematic text for the LLM to scan, and the bond prospectus offered the best combination of completeness, objectivity, and consistent formatting.

5.2 Use of Fundamental Score Labels and Reference Cases

Capital Four's existing Fundamental Score labels provided the basis for our ground truth. Although these analyst-derived scores are stored in a simple 1-to-5 format, they also include free-text descriptors for various risk parameters (e.g., technology dependence, regulatory exposure). We used those parameters to derive binary labels: either a certain risk is present for the issuer (1) or absent (0).

We adopted these Fundamental Score labels for two main reasons:

1. **Best Available Risk Data:** The firm currently tracks only these 1-5 scores and risk descriptors, so they represent the most direct link between textual disclosures and analyst judgment. Even if the labeling schema is somewhat broad, it offered enough resolution to evaluate whether an LLM can identify a relevant risk.
2. **Reference Cases for Few-Shot Examples:** In addition to the negative risk labels, the company maintains reference cases that illustrate a risk being either present or mitigated in real-life scenarios. By embedding these illustrative snippets into prompts, we could guide the model more precisely. This we hoped would be especially useful for ambiguous text references, where we wanted the model to look to a few-shot example before deciding.

5.3 Exhaustive Approach and Experimental Setup

5.3.1 Exhaustive (Explorative) Risk Detection

We intentionally took an *exhaustive* approach to searching each prospectus for each risk factor. Instead of only scanning the Risk Factors section, our pipeline processed

every single section from introductory legal notices to financial details trying to detect references to each of the four risk categories tested.

Although this strategy is more computationally expensive (e.g., we had to prompt the model once per subsection per risk factor), it allowed us to discover *where* in the document risk signals might appear. Without prior knowledge on which sections might matter most (since no comparable automated detection research preceded ours), we opted to test widely. This approach did increase costs in terms of API tokens and runtime, but it was necessary to ensure we did not exclude unexpected sources of risk information.

5.3.2 Subsection-Level Analysis and Company-Level Aggregation

After converting each PDF prospectus into a structured dataset (see Chapter 4), we divided the text into subsections (and, in some cases, sub-subsections) so the LLM could focus on moderate text chunks. The model then produced a "Yes" or "No" label for each risk parameter in each subsection. Finally, we **aggregated** these labels to the issuer (company) level: if *any* subsection for a given prospectus was labeled "Yes" for a risk, the issuer was deemed to have that risk. We compared these final issuer-level classifications with the ground truth from analyst scores.

5.4 Model Selection, Configuration, and Reliability

5.4.1 Model Choice

We experimented with several local (on-premise) LLMs as well as small-scale and mid-scale GPT variants, noting that data privacy considerations made fully public APIs less desirable at first. However, the smaller local models struggled to maintain strict JSON output formatting, even after multiple prompt iterations.

Ultimately, we adopted **ChatGPT-4o-mini** as our primary model. While it requires a managed API environment, it offered a favorable balance of cost, output reliability, and domain-specific interpretive power. Unlike smaller models that frequently dropped the required JSON schema, ChatGPT-4o-mini adhered to the prompt specifications in a single pass. We used default generation settings (temperature, top-k sampling) to keep the outputs stable and systematically comparable. Details on these parameters appear in Appendix A.3.

5.4.2 Intra-Annotator Reliability (Repeated Runs)

Since LLMs sometimes produce inconsistent answers when re-prompted, we wanted to ensure that ChatGPT-4o-mini was not making random guesses. To check for stability, we ran the *same* set of prospectuses and risk prompts multiple times without changing any model parameters and measured how often the labels differed. This *intra-annotator* agreement test is essential for a credit risk workflow: a model that often flips between "risk" and "no risk" on consecutive runs is not yet ready for real-world use. As shown later in Chapter 7, we quantify this consistency using Fleiss' Kappa.

5.5 LLM-Based Risk Detection Workflows

We implemented two main approaches. Both approaches involve prompting the LLM with a short question (e.g., "Does the text indicate the industry is susceptible to rapid technological advances?") and specifying a strict JSON output format. However, the second approach adds a *few-shot evaluation step* to filter out borderline cases.

5.5.1 Approach 1: Naive Single-Pass Detection

1. **Subsection-Level Prompting:** For each subsection of a prospectus, we ask whether a particular risk is present, instructing the LLM to return a "Yes" or "No" and brief evidence.
2. **Company-Level Aggregation:** If *any* subsection returns "Yes," the company is labeled with that risk factor. This label is then compared to the corresponding ground truth.

Although this single-pass approach is straightforward, it often produces false positives. For instance, the text might mention "rapid technological innovation" as an industry background note rather than a real risk factor. The model might still label it as a risk.

5.5.2 Approach 2: Two-Pass Detection with Few-Shot Evaluation

In the second approach, we try to refine those borderline "Yes" flags. Specifically:

1. **Detection Step (Same as Above):** We begin just like the naive approach, classifying each subsection as "Yes" or "No."
2. **Evaluation Step with Reference Cases:** Whenever the model labels a subsection as "Yes," we re-prompt it with a short list of reference scenarios. These examples illustrate a clear risk presence versus a non-risk mention. The model then has to confirm or override its own initial decision in light of these reference cases.

The idea is that by showing the model how analysts distinguish a real technological risk (for example) from just a mention of new technology, it might discard false positives. We also record whether the model changes its classification, which helps us measure how effective the reference examples are in refining final outcomes. To see the reference cases passed to the LLM, we refer to Appendix A.2.

5.6 Token Usage Analysis

In addition to evaluating the risk detection performance of our LLM-based pipeline, we conducted an analysis of token usage across the processed prospectus sections. Given that our approach involves feeding substantial amounts of text into the language model, understanding the distribution of text and by extension, the associated token consumption is crucial for both cost management and computational efficiency. In this thesis, we approximate token usage by measuring the word count in each subsection, under three key metrics:

- **Total Text Length (words):** Aggregates the word count from all text chunks within a given section, serving as a proxy for the total number of tokens that would be processed.
- **Avg Text Length (words):** Divides the total word count by the number of unique prospectuses contributing to the section. This metric provides an average measure of text volume per document.
- **Text Percentage (%):** The ratio of the section's total text length to the overall text length across all sections. It indicates the proportional contribution of each section to the entire dataset.

Although we ultimately process *every* subsection of the prospectus to avoid missing any risk signals, these word-count metrics shed light on where the greatest token consumption occurs. For instance, sections with extensive legal details or disclaimers may contain large volumes of text that do not necessarily yield more risk signals, thus inflating total token costs. In Chapter 7, we present the breakdown of token usage and discuss how it interacts with our risk detection workflow. We also explore

whether bond prospectuses follow the Pareto principle that is, whether about 80% of the risks are concentrated in roughly 20% of the text.

5.7 Summary of Methodology

In summary, our methodological choices aimed to thoroughly explore whether an LLM-based system can reliably spot important risk factors within bond prospectuses. We selected the prospectus as our data source because it is the most objective, consistently structured document available, and because we lacked prior research pinpointing exactly which sections hide relevant risk clues. We therefore scanned *every* section for multiple risk categories, guided by the best data we had from the company: a set of negative risk parameters within each issuer's 1-5 Fundamental Score, plus reference cases to illustrate each risk. Finally, to verify we were not seeing arbitrary or random outputs, we re-ran each experiment and calculated intra-annotator reliability.

Next, Chapter 6 provides an overview of the implementation details that constitute a significant part of our contributions. Following that, in Chapter 7, we present our findings, including precision, recall, and agreement metrics for both the single-pass "naive" approach and the two-pass method that incorporates reference examples. By comparing these outputs to the analyst-derived ground truth, we gain insight into how effectively an LLM can prioritize real risks while minimizing false alarms.

CHAPTER 6

Implementation and Usage Details

A key contribution of this work is an implementation that automates the following:

1. Converting bond prospectuses into a structured, analyzable data source.
2. Using Large Language Models (LLMs) to detect and categorize risk factors in these documents.
3. Evaluating the LLM's outputs against ground truth labels and measuring intra-annotator agreement.

This chapter provides a high-level overview of how these components are put together and used. For more in-depth explanations of the underlying scripts, data structures, and code logic, please refer to Appendix C.1 ("Extended Implementation Details").

6.1 Repository Organization

The project is hosted in a version-controlled repository that separates functionality into clear modules:

- **Data Collection**: Scripts to scrape prospectuses from SharePoint and ESMA.
- **Data Processing**: PDF-parsing that convert documents into structured text.
- **LLM Analysis**: Code that prompts an LLM to detect risks in each subsection.
- **Evaluation**: Tools for comparing LLM predictions against analyst-based ground truth and measuring consistency (Fleiss' Kappa).

Within these modules, each script follows a naming pattern (e.g., `scrape_*.py`, `parse_*.py`, `analysis_*.py`) that indicates its purpose. Detailed descriptions and code snippets appear in Appendix C.1.

6.2 Typical Usage Flow

1. **Collect Prospectuses**
   Use the data-collection scripts to gather PDFs from either:
   - **SharePoint** (internal portal at Capital Four), or
   - **ESMA** (publicly available listings).
2. **Process Documents**
   Run the data-processing script on the downloaded PDFs. This script:
   a) Converts each PDF into Markdown (preserving headings).
   b) Builds a hierarchical structure (sections, subsections, etc.).
   c) Outputs a CSV file containing all text blocks plus metadata.
3. **LLM-Based Risk Detection**
   Invoke the LLM analysis script to label each subsection according to risk factor presence (e.g., technology risk, market dynamics):
   - **Detection Step**: Classifies each text block as `Yes`/`No` for a given risk.
   - **Evaluation Step**: Reevaluates all decisions using a few-shot prompt that includes analyst reference cases.
4. **Compare with Ground Truth**
   The final step uses the evaluation module to:
   a) Aggregate subsection-level predictions into issuer-level risk labels.
   b) Compare predictions to the analyst-derived ground truth.
   c) Compute confusion matrix, precision/recall, and intra-annotator agreement.

For scripts, command-line parameters, and environment details, see Appendix C.1.

6.3 Key Practical Considerations

6.3.1 Managing LLM Usage Cost

Each subsection–risk query to the LLM incurs a token cost (when using external APIs).

- **Cost Tracking**: Scripts can log the cumulative tokens spent.
- **Limiting Scope**: Users can choose to scan only the *Risk Factors* section rather than all sections, significantly reducing queries.

6.3.2 Maintaining Reproducibility

- **Version Control**: All scripts are tracked in Git.
- **Docker**: Multi-stage builds provide a consistent environment for parsing and model inference.

For a more extensive breakdown such as the exact parameters used for local versus cloud-based LLMs see Appendix C.1.

6.4 Summary of Value

By unifying PDF parsing, LLM-based annotation, and automated evaluation, this implementation streamlines a previously manual and time-intensive credit-risk workflow. The repository:

1. **Enables** large-scale text analysis of bond prospectuses.
2. **Facilitates** structured experiments on risk detection.
3. **Offers** a reproducible pipeline for comparing results with analyst ground truth.

Technical details that would interrupt the thesis flow are gathered in Appendix C.1, where readers can learn precisely how each module operates and how to tailor it to new datasets or risk factors.

CHAPTER 7
Experimental Results

This chapter presents the results of our LLM experiments on detecting specific risk factors in a subsample of bond prospectuses. We conducted three independent runs to assess the model's reliability and the impact of a second, "evaluation" pass that reduces false positives. Below, we summarize the main observations from each run's confusion matrices and the associated metrics (precision, recall, F1, and accuracy).

7.0.1 Overall Detection and Evaluation Metrics

Tables 7.1–7.3 show the combined (i.e., all-label) confusion matrices and high-level metrics for the three runs. Each run consists of:

- A **Detection Step**, in which the LLM scans the text of each prospectus subsection for a given risk label (e.g., "Technology Risk") and decides "Yes" (risk present) or "No" (risk absent).

- An **Evaluation Step**, in which the LLM's initial "Yes" labels are reevaluated against short reference examples. This second pass can overturn some questionable "Yes" classifications, thereby reducing false positives.

Metric	Detection Step	Evaluation Step
TP	39	39
FP	160	157
FN	0	0
TN	1	4
Precision	0.196	0.199
Recall	1.000	1.000
F1 Score	0.328	0.332
Accuracy	0.200	0.215

Table 7.1. Confusion matrix metrics for detection and evaluation steps in Run 1.

Metric	Detection Step	Evaluation Step
TP	37	37
FP	159	140
FN	0	0
TN	0	16
Precision	0.189	0.206
Recall	1.000	1.000
F1 Score	0.318	0.341
Accuracy	0.189	0.270

Table 7.2. Confusion matrix metrics for detection and evaluation steps in Run 2.

Metric	Detection Step	Evaluation Step
TP	37	37
FP	159	143
FN	0	0
TN	0	16
Precision	0.189	0.206
Recall	1.000	1.000
F1 Score	0.318	0.341
Accuracy	0.189	0.270

Table 7.3. Confusion matrix metrics for detection and evaluation steps in Run 3.

7.0.2 Key Observations

Perfect Recall but Many False Positives. Across all three runs, the LLM consistently shows a recall of 1.00, meaning it never fails to detect an actual risk mentioned in the analyst labels (no missed detections). However, this comes at the cost of a high false-positive rate, which drives down precision to roughly 0.19–0.20 in the Detection Step. In other words, while the model achieves perfect recall of genuine risks, it often overestimates by labeling more subsections as risky than analysts do.

Partial Correction by the Evaluation Step. In each run, the Evaluation Step reduces the number of false positives. The additional prompt providing short "reference cases" or examples of true versus spurious risk helps the model filter out borderline mentions. Although this increased precision somewhat (e.g., from 0.196 to 0.199 in Run 1, and from 0.189 to 0.206 in Runs 2 and 3), the improvement remains modest. Accuracy also improves accordingly (e.g., 0.20 to 0.215 in Run 1, and 0.189 to approximately 0.27 in Runs 2–3).

Consistency Across Repeated Runs. We ran the same pipeline three times, prompting the model with identical data and prompt instructions. The overall results

show minimal variation, suggesting the system reliably flags the same subsections as risky or non-risky. While slight differences in false positives appear across runs (e.g., 159 vs. 160 in some cases), the pattern remains consistent: recall stays at 100%, precision hovers in the high-teens or low-twenties, and a second pass slightly improves results.

7.0.3 Per-Label Confusion Matrices

Tables 7.4 and 7.5 illustrate how each individual risk category fared in Run 1 and Run 2, respectively. The patterns in Run 3 closely match Run 2 and are omitted here for brevity. Notably, certain labels (such as *Technology Risk* and *Regulatory Framework*) tend to have higher precision than *Intra-Industry Competition* or *Market Dynamics*, likely due to more specific language cues.

Label	TP	FP	FN	TN	Precision	Recall	F1	Acc.
Intra-Industry Competition	5	44	0	1	0.10	1.00	0.19	0.12
Market Dynamics	5	45	0	0	0.10	1.00	0.18	0.10
Regulatory Framework	13	37	0	0	0.26	1.00	0.41	0.26
Technology Risk	16	34	0	0	0.32	1.00	0.48	0.32

Table 7.4. Confusion matrix metrics by each risk label in Run 1's detection step.

Label	TP	FP	FN	TN	Precision	Recall	F1	Acc.
Intra-Industry Competition	5	44	0	0	0.10	1.00	0.19	0.10
Market Dynamics	5	44	0	0	0.10	1.00	0.19	0.10
Regulatory Framework	12	37	0	0	0.24	1.00	0.39	0.24
Technology Risk	15	34	0	0	0.31	1.00	0.47	0.31

Table 7.5. Confusion matrix metrics by each risk label in Run 2's detection step.

Examining label-level trends clarifies where the model's domain understanding is stronger (e.g., technology-related language) versus more general competitive or market dynamics. These patterns mirror real-world observations: phrases about "emerging tech threats" or "rapid innovation cycles" may appear more distinctly than, say, subtle references to shifts in a competitive landscape.

7.0.4 Intra-Annotator Agreement Analysis

We treated each run as an annotator and calculated Fleiss' Kappa to measure agreement. The LLM is highly consistent in the first pass (detection) across repeated runs, with only slight variations. The second pass (evaluation) introduces more variance for certain borderline sections, especially in the "Intra-Industry Competition" category,

suggesting that more specialized reference examples or tighter model parameters may be needed for consistent re-checks.

Table 7.6 summarizes the aggregated Fleiss' Kappa values at two stages:

1. **Detection-Level**: Immediately after the model's first pass classifies each subsection (and thereby the company as a whole).
2. **Evaluation-Level**: After the second pass (the "evaluation" step) potentially overturns some of the initial "Yes" labels based on few-shot reference examples.

Risk Label	Detection Kappa	Evaluation Kappa
Market Dynamics	0.85	0.80
Intra-Industry Competition	0.84	0.44
Technology Risk	0.90	0.87
Regulatory Framework	0.93	0.80

Table 7.6. Aggregated intra-annotator agreement (Fleiss' Kappa) across repeated runs, shown for both detection and evaluation steps by risk label. Values above 0.80 typically indicate near-perfect agreement among the three runs, while 0.40–0.60 is considered moderate agreement [31].

7.0.5 Interpretation and Future Directions

These intra-annotator (i.e., intra-LLM) agreement results indicate that *at a high level*, ChatGPT-4o-mini is reproducible in its initial ("detection-level") risk classifications across repeated queries. However, the second pass introduces a dimension of variability for certain risk categories, especially *Intra-Industry Competition*. In practical terms, this suggests that:

1. Researchers and practitioners should be aware that *even with the same model and prompt settings*, iterative steps or re-prompts can yield different classifications for borderline cases.
2. Adjusting model parameters such as `temperature` could further stabilize these marginal decisions, a worthwhile avenue for future research and experimentation.

7.0.6 Token Usage Analysis

In line with the methodology described in Section 5.6, we examined the distribution of text and by proxy, token consumption across the prospectus sections processed by

our pipeline. As a reminder, we approximate token usage by measuring the word count, using the following key metrics:

- **Total Text Length (words):** Aggregates the word count from all text chunks within a given section, serving as a proxy for the total tokens processed.
- **Avg Text Length (words):** The total word count divided by the number of unique prospectuses contributing to the section, providing an average text volume per document.
- **Text Percentage (%):** The ratio of a section's total text length to the overall text length across all sections, indicating its proportional contribution.
- **Accumulated Text Percentage (%):** The running total of text percentages from the top of the report down to the current section, showing the cumulative share of text.
- **Overall Risk Percentage (%):** The percentage of overall risk counts (TP + FP) for the section relative to the grand total risk count across all sections.
- **Accumulated Overall Risk Percentage (%):** The running total of overall risk percentages from the top of the report down to the current section, representing the cumulative contribution to overall risk.

Although our dataset spans 745 distinct section titles, Table 7.7 reproduces the first 75 rows of the script output for illustration. This excerpt details both risk detection counts (true positives and false positives across various risk categories) and the corresponding text-length metrics.

Implications for Token Usage. Since GPT-based systems price API calls by token count, the *Total Text Length (words)* and *Avg Text Length (words)* per section provide a rough estimate of the computational cost incurred. In our explorative approach where each subsection's text is processed alongside prompts for multiple risk categories sections with large word counts (such as `RISK FACTORS` and `DESCRIPTION OF THE NOTES`) are particularly token-intensive. While this comprehensive scanning ensures that no potential risk signal is overlooked, it also increases both runtime and cost.

Notably, from the displayed first 75 rows (out of 745 rows) of Section Titles sorted by overall total identified risk (TP + FP), we see that to catch 83% of the identified risks, we have to process 71% of the total text. This finding indicates that risks are distributed across many sections rather than being concentrated in just a few, which contradicts the 80–20 rule.

Section Title	Market Dynamics - a TP	Market Dynamics - a FP	Intra-Industry Competition - a TP	Intra-Industry Competition - a FP	Technology Risk - a TP	Technology Risk - a FP	Regulatory Framework - a TP	Regulatory Framework - a FP	Overall TP	Overall FP	Overall Total	Total Text Length (words)	Avg Text Length (words)	Prospectus Count	Text Percentage (%)	Overall Risk Percentage (%)	Accumulated Text Percentage (%)	Accumulated Overall Risk Percentage (%)
RISK FACTORS	54	255	25	108	124	249	232	711	435	1,323	1,758	1,145,090	23,369	49	10.3%	25.3%	10.3%	25.3%
MANAGEMENT'S DISCUSSION AND ANALYS...	28	140	2	5	53	149	43	96	126	390	516	493,669	20,569	24	4.4%	7.5%	14.7%	33.2%
SUMMARY	14	27	0	1	80	136	13	66	107	230	337	348,283	8,930	39	3.1%	4.2%	17.8%	38.1%
PLAN OF DISTRIBUTION	0	0	2	22	0	0	85	189	87	211	298	158,136	3,765	42	1.4%	4.4%	19.2%	42.5%
DESCRIPTION OF THE NOTES	0	0	0	0	1	4	68	206	69	210	279	1,163,368	72,710	16	10.4%	4.1%	29.6%	46.6%
INDUSTRY	1	30	0	0	22	76	1	49	24	155	179	124,410	17,772	7	1.1%	2.6%	30.7%	49.1%
BUSINESS	7	8	0	0	11	51	0	41	18	100	118	80,982	16,196	5	0.7%	1.7%	31.5%	50.9%
DESCRIPTION OF OTHER INDEBTEDNESS	0	3	0	0	0	1	31	55	31	59	90	558,058	55,805	10	5.0%	1.3%	36.5%	52.2%
CERTAIN ERISA CONSIDERATIONS	0	0	0	0	0	0	14	73	14	73	87	88,118	4,005	22	0.8%	1.3%	37.3%	53.5%
REGULATION	0	3	0	1	0	5	16	56	16	65	81	124,711	31,177	4	1.1%	1.2%	38.4%	54.7%
LIMITATIONS ON VALIDITY AND ENFORC...	0	0	0	0	0	1	0	76	0	77	77	88,059	88,059	1	0.8%	1.1%	39.2%	55.8%
DESCRIPTION OF CERTAIN FINANCING A...	0	0	0	0	0	0	4	68	4	68	72	404,134	36,739	11	3.6%	1.1%	42.8%	56.8%
CAPITALIZATION	0	2	0	0	15	1	33	19	48	22	70	261,741	5,816	45	2.3%	1.0%	45.1%	57.9%
INDUSTRY AND MARKET DATA	10	1	0	4	1	21	2	22	13	48	61	23,982	1,498	16	0.2%	0.9%	45.3%	58.8%
DESCRIPTION OF NOTES	0	0	0	0	1	0	25	29	26	29	55	413,305	45,922	9	3.7%	0.8%	49.0%	59.6%
CERTAIN LIMITATIONS ON THE VALID...	1	0	0	0	0	1	0	52	1	53	54	40,486	40,486	1	0.4%	0.8%	49.4%	60.3%
TRANSFER RESTRICTIONS	0	0	0	1	0	0	7	44	7	45	52	88,043	2,667	33	0.8%	0.8%	50.2%	61.1%
OUR BUSINESS	0	2	0	0	25	0	22	0	47	2	49	22,617	22,617	1	0.2%	0.7%	50.4%	61.8%
LIMITATIONS ON VALIDITY AND ENFO...	0	0	0	0	0	0	0	47	0	47	47	66,732	66,732	1	0.6%	0.7%	51.0%	62.5%
WHERE YOU CAN FIND MORE INFORMAT...	5	1	0	2	0	11	2	25	7	39	46	206,957	12,934	16	1.9%	0.7%	52.9%	63.2%
CERTAIN TAX CONSIDERATIONS	0	0	0	0	0	0	7	38	7	38	45	67,294	11,215	6	0.6%	0.7%	53.5%	63.8%
FORWARD-LOOKING STATEMENTS	2	11	0	0	4	8	5	13	11	32	43	23,909	1,406	17	0.2%	0.6%	53.7%	64.5%
STABILIZATION	0	0	1	5	0	0	0	36	1	41	42	16,472	2,353	7	0.2%	0.6%	53.8%	65.1%
LIMITATIONS ON VALIDITY AND ENFO...	0	0	0	0	0	1	0	41	0	42	42	94,250	94,250	1	0.8%	0.6%	54.7%	65.7%
REGULATORY ENVIRONMENT	0	3	0	1	5	6	0	26	5	36	41	14,846	7,423	2	0.1%	0.6%	54.8%	66.3%
STABLE UTILIZATION RATES OF DACH...	8	0	0	0	0	14	0	19	8	33	41	56,195	56,195	1	0.5%	0.6%	55.3%	66.9%
IMPORTANT NOTICE	0	0	0	0	0	0	11	27	11	27	38	23,593	2,949	8	0.2%	0.6%	55.5%	67.4%
REGULATORY MATTERS	0	0	1	0	0	2	0	34	1	36	37	8,424	8,424	1	0.1%	0.5%	55.6%	68.0%
WHERE PROSPECTIVE INVESTORS CAN F...	0	3	0	0	0	3	31	0	31	6	37	126,632	126,632	1	1.1%	0.5%	56.7%	68.5%
CERTAIN U.S. FEDERAL INCOME TAX C...	0	0	0	0	0	0	11	25	11	25	36	43,554	3,111	14	0.4%	0.5%	57.1%	69.0%
TAXATION	0	0	0	0	0	0	17	18	17	18	35	19,019	4,754	4	0.2%	0.5%	57.3%	69.6%
LITIGATION	0	0	2	0	1	0	32	0	35	0	35	14,448	14,448	1	0.1%	0.5%	57.4%	70.1%
IMPORTANT INFORMATION ABOUT THIS ...	0	0	0	1	0	0	20	13	20	14	34	26,742	5,348	5	0.2%	0.5%	57.6%	70.6%
MANAGEMENT'S DISCUSSION AND ANALY...	8	0	0	0	0	17	0	7	8	24	32	16,845	16,845	1	0.2%	0.5%	57.8%	71.0%
DESCRIPTION OF CERTAIN OTHER INDE...	0	0	0	0	0	1	0	29	0	30	30	170,488	28,414	6	1.5%	0.4%	59.3%	71.5%
INDEPENDENT AUDITORS	0	2	2	0	0	3	0	21	2	26	28	40,049	1,820	22	0.4%	0.4%	59.7%	71.9%
CERTAIN UNITED STATES FEDERAL INC...	0	0	0	0	0	0	3	25	3	25	28	20,718	3,453	6	0.2%	0.4%	59.9%	72.3%
3.** **CAPITAL MANAGEMENT	0	5	0	0	0	8	0	15	0	28	28	43,602	43,602	1	0.4%	0.4%	60.3%	72.7%
CERTAIN INSOLVENCY LAW CONSIDERAT...	0	0	0	2	0	0	0	25	0	27	27	21,080	21,080	1	0.2%	0.4%	60.4%	73.1%
CERTAIN INSOLVENCY LAW CONSIDERAT...	1	0	0	0	0	1	0	25	1	26	27	51,158	51,158	1	0.5%	0.4%	60.9%	73.5%
MORE FLEXIBILITY, TIME SAVING,AND...	0	0	0	0	0	15	12	0	12	15	27	20,709	20,709	1	0.2%	0.4%	61.1%	73.9%
THE OFFERING	0	4	0	0	0	3	5	14	5	21	26	100,235	3,456	29	0.9%	0.4%	62.0%	74.3%
SUMMARY HISTORICAL FINANCIAL INFO...	0	0	1	0	9	0	0	16	10	16	26	12,384	6,192	2	0.1%	0.4%	62.1%	74.6%
COVID-19	0	2	0	0	0	17	0	6	0	25	25	21,562	21,562	1	0.2%	0.4%	62.3%	75.0%
NOTICE TO INVESTORS	0	0	1	2	0	0	7	14	8	16	24	40,193	1,826	22	0.4%	0.4%	62.7%	75.4%
LTIP 2019	0	2	1	0	0	6	0	15	1	23	24	26,919	26,919	1	0.2%	0.4%	62.9%	75.7%
SUBSCRIPTION AND SALE	0	0	0	0	0	0	17	7	17	7	24	9,250	3,083	3	0.1%	0.4%	63.0%	76.1%
OFFERING CIRCULAR SUMMARY	0	2	0	0	0	18	0	4	0	24	24	12,794	6,397	2	0.1%	0.4%	63.1%	76.4%
LIMITATIONS ON VALIDITY AND ENFOR...	0	0	0	0	0	0	24	0	24	0	24	38,140	38,140	1	0.3%	0.4%	63.4%	76.8%
BOOK ENTRY; DELIVERY AND FORM	0	0	0	0	0	3	7	12	7	15	22	17,672	3,534.40	5	0.2%	0.3%	63.6%	77.1%
LTIP 2020	0	2	1	0	0	4	0	15	1	21	22	27,654	27,654.00	1	0.3%	0.3%	63.8%	77.4%
LIMITATIONS ON VALIDITY AND ENFOR...	0	0	0	0	0	0	22	0	22	0	22	22,709	22,709.00	1	0.2%	0.3%	64.0%	77.7%
DESCRIPTION OF THE SENIOR SECURED...	0	0	0	0	0	0	0	22	0	22	22	167,502	167,502.00	1	1.5%	0.3%	65.5%	78.0%
OUR STRENGTHS	0	8	0	1	5	6	0	1	5	16	21	25,618	12,809.00	2	0.2%	0.3%	65.8%	78.3%
LIMITATIONS ON VALIDITY AND ENFOR...	0	0	0	0	0	0	21	0	21	0	21	21,127	21,127.00	1	0.2%	0.3%	66.0%	78.7%
BOOK ENTRY, DELIVERY AND FORM	0	0	0	0	4	2	5	9	9	11	20	22,192	3,698.67	6	0.2%	0.3%	66.2%	78.9%
(IN MILLIONS)	0	2	0	0	0	4	0	14	0	20	20	27,010	27,010.00	1	0.2%	0.3%	66.4%	79.2%
I-LOGIC TECHNOLOGIES BIDCO LIMITE...	0	3	0	0	6	0	0	10	6	13	19	11,835	11,835.00	1	0.1%	0.3%	66.5%	79.5%
KEY INCOME STATEMENT ITEMS	0	17	0	0	0	0	0	1	0	18	18	36,677	18,338.50	2	0.3%	0.3%	66.8%	79.8%
CERTAIN INSOLVENCY CONSIDERATIONS...	0	0	0	0	0	0	0	18	0	18	18	11,995	11,995.00	1	0.1%	0.3%	66.9%	80.0%
LIABILITIES AND STOCKHOLDERS' EQU...	0	6	0	1	2	0	0	8	2	15	17	32,576	32,576.00	1	0.3%	0.3%	67.2%	80.3%
KEY FACTORS AFFECTING OUR RESULTS...	0	9	0	0	1	1	0	6	1	16	17	17,748	8,874.00	2	0.2%	0.3%	67.4%	80.5%
CASH FLOWS FROM FINANCING ACTIVIT...	0	4	0	0	0	4	0	9	0	17	17	32,170	32,170.00	1	0.3%	0.3%	67.7%	80.8%
SERVICE OF PROCESS AND ENFORCEMEN...	1	0	0	1	0	0	0	14	1	15	16	79,044	19,761.00	4	0.7%	0.2%	68.4%	81.0%
IMPORTANT INFORMATION	0	0	0	1	0	0	0	15	0	16	16	15,928	3,185.60	5	0.1%	0.2%	68.5%	81.3%
LIABILITIES AND EQUITY	4	0	0	0	0	3	0	9	4	12	16	30,843	15,421.50	2	0.3%	0.2%	68.8%	81.5%
CERTAIN INSOLVENCY LAW CONSIDERAT...	0	0	0	1	0	0	0	15	0	16	16	19,733	19,733.00	1	0.2%	0.2%	69.0%	81.7%
NOTES TO CONSOLIDATED FINANCIAL S...	0	1	0	0	5	0	0	9	5	10	15	19,920	19,920.00	1	0.2%	0.2%	69.2%	81.9%
DESCRIPTION OF CERTAIN FINANCING ...	0	0	0	0	0	1	0	14	0	15	15	110,343	110,343.00	1	1.0%	0.2%	70.2%	82.2%
UNITED KINGDOM	0	0	0	0	0	0	0	15	0	15	15	14,650	7,325.00	2	0.1%	0.2%	70.3%	82.4%
PRINCIPAL STOCKHOLDERS	0	0	0	0	0	0	0	15	0	15	15	62,960	62,960.00	1	0.6%	0.2%	70.8%	82.6%
ACURIS INTERNATIONAL LIMITED COMP...	0	0	0	1	6	0	0	8	6	9	15	8,762	8,762.00	1	0.1%	0.2%	70.9%	82.8%
PRESENTATION OF INDUSTRY AND MARK...	1	4	1	2	0	6	0	0	2	12	14	9,598	1,919.60	5	0.1%	0.2%	71.0%	83.0%
IMPORTANT INFORMATION FOR INVESTO...	0	0	0	1	0	0	3	10	3	11	14	7,092	2,364.00	3	0.1%	0.2%	71.1%	83.2%

Table 7.7. Excerpt showing how text length metrics and risk detections (TP/FP) are aggregated by section, illustrating the distribution of token usage across prospectus sections. Long titles are truncated (with three dots "...").

7.0.7 Summary of Findings

Overall, these results suggest that LLM-based pipelines can *drastically reduce the chance of missing* an actual risk (recall of 100%), which is crucial in a high-stakes domain like credit assessment. However, the method's tendency to over-flag requires a supplementary step such as our evaluation pass or additional prompt engineering to be practical in a real production workflow. The partial reduction in false positives between the Detection and Evaluation Steps confirms the importance of rechecking flagged subsections with concrete examples or rules. Further refinement of these techniques might improve precision enough to make the approach a robust front-line assistant for credit analysts.

We see that a simple "80–20" approach is unworkable. If we limit our attention to only a small number of sections, we could miss a significant number of risks. Although our exhaustive scanning strategy ensures that no risk signals are overlooked, it also results in higher token costs. To build a robust, production-ready pipeline for credit risk analysis, additional steps such as evaluation passes or more advanced filtering and chunking strategies are needed to balance comprehensive coverage with computational efficiency.

CHAPTER 8

Discussion

The results of our experiments underline both the promise and limitations of using Large Language Models (LLMs) to detect fundamental risks in bond prospectuses. On one hand, the pipeline consistently achieved perfect recall across multiple runs, indicating it did not overlook any risk parameters flagged by human analysts. This outcome is compelling from a credit-risk standpoint: it is generally more harmful to miss a potential threat than to flag extra ones. On the other hand, the model often interpreted standard disclaimers or routine mentions of factors like technology or regulatory concerns as true risks, leading to considerable over-flagging. These false positives lowered precision to around 20%, which would create a substantial workload for analysts in a production setting.

8.1 High Recall and Over-Flagging

8.1.1 Advantages of Perfect Recall

A central benefit of the pipeline is its ability to detect all analyst-identified risks, ensuring that no genuine red flags go unnoticed. In high-stakes domains like credit evaluation where missing a key threat can be far more damaging than dealing with surplus alerts this characteristic may be desirable. Even incremental improvements in recall, relative to existing processes, could help reduce the chance of overlooking critical disclosures.

8.1.2 Drawbacks of Excessive False Positives

However, the abundance of irrelevant alerts poses a practical issue. When four out of five flagged items turn out to be false alarms, analysts can experience “alarm fatigue,” causing them to spend considerable time checking routine sections that pose no genuine threat. If this pattern persists, practitioners might come to distrust

or simply ignore the model's recommendations. Over-flagging also obscures which text segments truly warrant deeper attention, undermining one of the major goals of automation namely, improving efficiency.

8.2 Two-Pass Evaluation and Modest Precision Gains

8.2.1 Impact of Reference Examples

To address false positives, we introduced a second pass in which the model revisits its "Yes" decisions using brief examples that illustrate what a real risk versus a routine mention might look like. This additional step did reduce the false-positive rate somewhat, offering a slight but measurable boost to precision. The improvement, though modest, confirms that targeted prompts or reference cases can nudge the model toward closer alignment with analyst judgments.

Nevertheless, the second pass did not solve the problem entirely. For certain risk categories particularly those tied to ambiguous or broadly worded statements about competition the final classification varied more than expected. While the model grew more conservative about labeling some text passages as risks, it remained inconsistent for borderline snippets that mention industry-level shifts or generic competitive factors.

8.2.2 Reasons for Persistent Over-Flagging

The persistence of false positives likely stems from several factors. First, prospectuses commonly include catch-all legal language to satisfy disclosure requirements, which can read as genuine risk indicators if not filtered carefully. Second, broad prompts (for example, "Does this text indicate a vulnerability to technological disruption?") may encourage the model to err on the side of caution. Finally, the model was not specifically fine-tuned on sector- or issuer-level data; as a result, it lacks the more specialized judgment that seasoned financial analysts acquire by reading countless prospectuses.

8.3 Consistency and Practical Integration

8.3.1 Stable Initial Detection, Variable Re-Evaluation

Another notable finding is that the model's first-pass results were highly stable across multiple runs, as shown by high Fleiss' Kappa values. This consistency implies that the pipeline would likely flag the same paragraphs each time it processes a new document. However, once reference examples or a second prompt were introduced, the system demonstrated more volatility for certain categories, particularly Intra-Industry Competition. In real-world use, teams might accept the first-pass consistency for some areas (e.g., well-defined technology disclosures) but would need more nuanced prompt engineering where language is especially vague.

8.3.2 Role of Human Analysts

Taken as a whole, these outcomes suggest that analysts still need to oversee the model's flags. While LLMs clearly help prevent oversights thanks to their exhaustive text coverage they may also require analysts to filter out extensive boilerplate. A possible workflow would have the LLM generate an initial set of flags, then run a follow-up pass with domain-specific examples or additional filtering logic, and ultimately present the reduced set of alerts to an analyst for final review. This hybrid approach might increase overall efficiency by ensuring that the few genuinely concerning points stand out more clearly.

8.4 Future Directions

Several refinements could help balance perfect recall with acceptable precision. First, prompt tuning especially for risk categories prone to misclassification may steer the model away from labeling broad generic text. Second, more tailored or detailed reference examples could show the model a wider variety of "false alarm" scenarios, making it less likely to flag them. Third, domain-specific fine-tuning could be valuable, especially if analysts can compile a richer collection of prospectus examples that clearly separate standard disclosures from genuine red flags. Lastly, multi-step reasoning or chain-of-thought approaches might allow the model to parse whether a risk mention is just boilerplate or truly specific to the issuer's situation.

Additionally, we note that we have identified 745 distinct section titles across our subsampled dataset of 50 prospectuses. A more selective approach could involve narrowing the processing to focus on only those section titles of clear relevance. Implementing this requires a reliable method of determining which category each version of a section title belongs to. For instance, one could perform a clustering analysis to group titles like "Our Group," "The Group's Business," "Our Business," and "The Company's Business" into a single category, then focus exclusively on those that genuinely pertain to risks.

It is important to emphasize that it does not make total sense to narrow down the risk identification to only a select number of Section Titles in the prospectus. As demonstrated by our token usage analysis, to catch 83% of the identified risks, we must process 71% of the total text. In other words, the risks are spread over many sections, rather than concentrated in a small handful. This observation underscores why a purely "80–20" approach is unworkable: limiting risk detection to just a few sections could miss a significant number of risks. A more holistic look across most if not all sections is crucial when building a robust, production-ready pipeline for credit risk analysis.

In addition, scaling our current approach to the full range of risks used by a firm (e.g., 23 risk factors) calls for a more efficient prompting workflow. Right now, we process each text chunk separately for each risk factor. In a real-world environment with 23 risk factors, this would be computationally expensive. One potential solution is an orchestrator workflow, as highlighted in an Anthropic blog post on agentic systems [32]. In such a setup, a central "orchestrator" model or routing logic would first identify the most likely risk(s) in a given text chunk and then activate *only the relevant risk identification LLM* instead of prompting for every risk 23 times for each chunk. We would still process all text chunks across the prospectus to avoid missing risks, but for each chunk, we would only call the specific risk identification model(s) that the orchestrator deems relevant. This approach could substantially reduce token usage and costs while maintaining comprehensive coverage, offering a more scalable solution for applying LLM-based risk identification to large volumes of text.

CHAPTER 9
Conclusion & Future Work

In this thesis, we examined whether Large Language Models (LLMs) can reliably spot important risk factors in corporate bond prospectuses. These documents are a core component of credit risk analysis, but they are also long and detailed. Our work centered on building a pipeline that reads entire prospectuses (not just the "Risk Factors" section) and then compares the LLM's detected risks to the analyst-based "fundamental score" framework used at Capital Four.

9.1 Contributions and Key Findings

A central achievement of this work is the end-to-end system we developed, which cleans and parses real prospectuses, runs them through an LLM for risk detection, and evaluates the results against ground-truth labels. We found:

- **LLMs show very high recall** In our experiments, the model did not miss any risks that analysts had flagged. This means it is good at catching red flags whenever they appear in the text.
- **False positives remain common** While the model rarely missed an actual risk, it also picked up many non-relevant risks, which caused a low precision of about 20%.
- **Reference examples help but only to a degree.** Providing the LLM with short "real vs. not real" risk scenarios did reduce false alarms slightly, which indicates that adding carefully chosen examples can make the model more cautious.
- **Consistency is strong on the first pass** Across several runs with the same prompts, the model mostly labeled the same subsections in the same way. However, once we asked it to revisit its decisions with extra prompts and examples, it sometimes changed its mind on borderline cases.

9.2 Practical Implications

Even with its tendency to over-flag, the LLM-based approach has value. In credit analysis, failing to detect a genuine threat can be far worse than issuing a few extra alerts. From that angle, a pipeline that consistently highlights all known dangers just with some false positives can still be a solid first step. Yet if four out of five flags turn out to be unnecessary, analysts may face "alert fatigue."

To address that, our results show that letting the model "double-check" its own flags can bring down the false-positive rate a bit. However, further fine-tuning or domain-specific prompts might be needed before this becomes a fully streamlined tool for daily use. The broad takeaway is that LLMs can serve as a helpful front-line scanner, but human reviewers must remain part of the loop to identify genuine issues from routine text.

9.3 Limitations

Although this thesis advances our understanding of how LLMs might automate parts of risk analysis, it has a few clear limitations:

- **Data Constraints** We focused on a sample of real-world prospectuses tied to specific risk categories. Prospectuses can vary in structure or terminology, so results may differ with other bond offerings or in different regulatory settings.
- **Generic LLM** We relied on standard LLMs that were not specialized in finance or law. Models more finely tuned on legal or credit-specific texts might handle ambiguous wording more accurately.
- **Cost and Token Usage** Processing entire prospectuses for multiple risks can be expensive in terms of tokens and runtime. More targeted scanning or chunking strategies might be necessary for large-scale deployment.
- **Limited Prompt Tuning.** We tested two-pass evaluation and a handful of examples, but more advanced prompt engineering (like step-by-step reasoning, or deeper financial domain prompts) was not fully explored.

9.4 Future Work

Several improvements could boost precision and further reduce unnecessary alerts:

- **Refined Prompt Engineering** Providing the model with richer references especially ones distinguishing typical boilerplate language from genuine issuer-specific threats could help cut back on false alarms.
- **Selective Section Scanning.** Although we showed that risks can appear in many sections, some portion of the text may be far less likely to contain critical red flags. An "orchestrator" model could first decide which sections look relevant, then pass only those along for closer analysis.
- **Domain-Specific Fine-Tuning.** Training or fine-tuning an LLM on credit and legal documents, including prospectuses, might help it interpret nuanced regulatory or industry-specific references more accurately.
- **Integration with Other Data Sources.** It may help to pair textual analysis with numerical credit metrics, ESG data, or rating-agency insights. That extra information could make the model more context-aware when deciding if a mention is an actual risk.
- **Application to Further Risk Parameters.** Extending this work to detect the full set of 23 risk factors (instead of the subset of 4 tested in our work) would confirm if the approach remains practical for real production setups.
- **Application to Corresponding Risk Strenghts** While we focused on negative or downside signals, LLMs could also evaluate the corresponding positive factors of Capital Four's fundamental scoring framework, expanding their use from risk detection alone to more holistic credit analysis.

9.5 Concluding Statement

Taken together, these findings suggest that LLMs are well-suited to assist credit analysts in flagging potential risks and reducing the chance of missing major red flags. At the same time, the quantity of false positives needs to be lowered before the system can meaningfully reduce analysts' workload. With further refinement such as more precise prompt engineering or domain-specific fine-tuning LLMs can ultimately serve as a practical tool in credit risk assessment.

Additionally, it would be wise to re-examine the definition of the risk factors themselves in collaboration with the credit analysts, making each risk parameter more specific. This approach tailors the scoring framework to the strengths of LLMs rather than trying to bend the models into an existing human-centered system, which helps reduce ambiguity and improve precision in automated risk detection. It would imply a shift towards a scoring framework that is natively built around LLM strengths, potentially resulting in more precise and objective evaluations of credit risk. Overall, with the ongoing evolution of LLMs, combined with solid data preparation and analyst-driven scoring frameworks, LLM-powered tools can become a valuable resource in

managing growing data demands, supporting analysts in delivering more consistent and thorough risk evaluations.

Appendices

APPENDIX A

Prompts & LLM Configurations

A.1 Prompt Templates

This section contains the two prompt templates used in our experiments: one for the initial detection step and another for the secondary evaluation step that incorporates reference examples.

A.1.1 Detection Prompt (Single LLM)

In the detection prompt (see Figure A.1), the model is asked to examine each text chunk and decide whether it indicates the presence of a specific risk. The instructions direct the model to answer "Yes" or "No" and to provide concise evidence in a structured JSON format.

DETECTION_PROMPT

```
DETECTION_PROMPT = """
Question: {question}
Title: {subsection_title}
Text: {subsection_text}

Provide your answer in JSON with no extra commentary, using this format:
{{
 "Answer": "Yes" or "No",
  "Evidence": "Exact sentences or phrases from the text that support the answer (blank if No)"
}}
"""
```

Figure A.1. *A visual representation of our detection prompt. The model is prompted to label each text chunk as indicating a specific risk (Yes/No), returning structured output.*

A.1.2 Evaluation Prompt (Dual LLM)

In the evaluation prompt (see Figure B.1), the model re-examines any subsection that it labeled "Yes" in the detection step. This second pass includes short reference examples of what genuinely constitutes a risk versus a routine mention. The goal is to refine the model's classification, reduce false positives, and produce a final label in the same structured JSON format.

EVALUATION_PROMPT

```
EVALUATION_PROMPT = """
You are a risk assessment evaluator. Your task is to verify whether the following risk factor is truly
present in the given subsection of a bond prospectus.

Risk Factor Question: {question}

Text Under Review:
Title: {subsection_title}
Text: {subsection_text}

Previous Evaluation Output:
Answer: {answer}
Evidence: {evidence}

Reference Examples:
(1) Negative Example (Risk Absent): {negative_case}

(2) Positive Example (Risk Present): {positive_case}

Instructions:
1. Analyze the provided Evidence in the context of the Text Under Review.
2. Compare the Evidence with the Reference Examples:
 - If the Evidence clearly follows the pattern of the Positive Example (indicating the risk factor is present),
then the correct answer should be "Yes".
   - If the Evidence is weak, ambiguous, or aligns more with the Negative Example (indicating the risk
factor is absent), then the correct answer should be "No".
3. Decide whether to confirm the previous evaluation or override it:
   - Confirm "Yes" if the evidence strongly supports the presence of the risk factor.
   - Override to "No" if the evidence does not convincingly indicate the risk factor.
4. Provide a concise explanation for your decision.

Output your final evaluation in JSON with the following structure:
{{
  "Answer": "Yes" or "No",
  "Reasoning": "A brief explanation of your decision."
}}
"""
```

Figure A.2. *A visual representation of our evaluation prompt. The model reviews flagged text again with real vs. not-real risk examples, aiming to filter out false positives.*

A.2 Reference Cases

Below are the reference cases used as few-shot examples in the evaluation prompt. Each “Negative Case” shows a scenario where the risk is absent, and each “Positive Case” illustrates where the risk is genuinely present. These paired examples guide the model in distinguishing between standard disclosures or minor mentions versus genuine risk exposures.

A.2.1 Cyclical Product Risk

Negative Case
Scenario: Submetering is an infrastructure-like business with long-term contractual agreements and non-discretionary demand.

Observations: 80% of revenue is recurring; stable demand even during recessions.

Model Decision: {”Answer”: ”No”, ”Evidence”: ”This business does not show cyclical product risk due to its long-term, stable nature.”}

Positive Case
Scenario: Construction equipment rented day-to-day.

Rationale: 57% of revenue from construction equipment; weak macroeconomic conditions historically had a high impact on business (-25% EBITDA in 2009).

Model Decision: {”Answer”: ”Yes”, ”Evidence”: ”The reliance on construction equipment and historical downturn in EBITDA indicates exposure to cyclical product risks.”}

A.2.2 Intra-Industry Competition

Negative Case
Scenario: High market concentration and the dominance of well-established brands versus private labels.

Observations: Historically, leading players have been able to increase prices through product innovation.

Model Decision: {”Answer”: ”No”, ”Evidence”: ”While there is mention of branding and price increases, it does not suggest an irrational or fundamentally unsustainable level of competition.”}

Positive Case
Scenario: A highly fragmented product offering dominated by non-branded or generic products.

Observations: The pricing environment has been highly competitive in recent years with competitors driving down prices. As a result, the company's market share declined when prices were kept unchanged to protect profitability.

Model Decision: {"Answer": "Yes", "Evidence": "A highly fragmented market with downward price pressure indicates intra-industry competition that is not based on underlying fundamentals."}

A.2.3 Technology Risk

Negative Case
Scenario: A leading cloud computing provider offering scalable storage and computing solutions to businesses worldwide.

Observations: Continuous investment in cutting-edge technology, strong R&D capabilities, and a robust infrastructure that supports high availability and security standards.

Model Decision: {"Answer": "No", "Evidence": "The company's proactive investment in technology and strong infrastructure mitigate technology risk, positioning it as a resilient leader in the cloud computing industry."}

Positive Case
Scenario: Traditional brick-and-mortar retail chain specializing in apparel with minimal online presence.

Observations: Limited adoption of e-commerce platforms, outdated inventory management systems, and slow response to digital marketing trends.

Model Decision: {"Answer": "Yes", "Evidence": "The company's lack of technological advancement and slow adaptation to digital retail trends expose it to significant technology risk, making it vulnerable to competition from more technologically adept retailers."}

A.2.4 Regulatory Framework

Negative Case
Scenario: Ongoing planning for elderly care is regulated at a national level with a high degree of reliance on the governmental level.

Observations: Elderly care is considered political goodwill, but the French state directs individuals to the most economical solutions in terms of public funding (i.e., homecare for the least dependent and medicalized nursing homes for others). The company offers both services and is therefore shielded from this strategic government focus.

Model Decision: {"Answer": "No", "Evidence": "The company's diversified service offerings protect it from targeted regulatory pressures, indicating a low regulatory framework risk."}

Positive Case
Scenario: Collections are regulated by various authorities and according to various statutes in each European country, with all countries aiming for customers to be treated "fairly".

Observations: The company profits from the most vulnerable group and is under political scrutiny. There is a trend in laws, rules, and regulations requiring increased availability of historic information about receivables for collection purposes, along with a higher degree of consumer protection.

Model Decision: {"Answer": "Yes", "Evidence": "The company operates under stringent and evolving regulations aimed at protecting vulnerable consumers, increasing its exposure to regulatory framework risks."}

A.3 LLM Configuration Files and Parameters

Our experiments used the `ChatGPT-4o-mini` model via the `LangChain` ChatOpenAI wrapper. Below is the snippet showing how we set up the model and handled environment variables for the API key:

```
if model_type == "openai":
    from langchain_openai import ChatOpenAI

    load_dotenv()
    OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")
    if not OPENAI_API_KEY:
        print("OPENAI_API_KEY not found. Make sure it's set in your .env file.")
        sys.exit(1)

    return ChatOpenAI(
        model_name="gpt-4o-mini",
```

```
        openai_api_key=OPENAI_API_KEY,
        max_tokens=256,
    )
```

The key configuration parameters for ChatGPT-4o-mini are listed in Table **??**. Note that while parameters such as `temperature` and `top_p` were not explicitly set in the initialization snippet, they default to values that are consistent with other OpenAI models: The default configuration for ChatGPT-4o-mini includes the following parameter settings:

- **Temperature**: 1.0 (controls the randomness of the output)
- **Top_p**: 1.0 (limits token selection based on cumulative probability)
- **Frequency Penalty**: 0.0 (discourages repetitive output)
- **Presence Penalty**: 0.0 (encourages topic variation)
- **Stop**: None (no explicit stop sequence)

These settings, particularly the default values for `temperature` and `top_p`, play a critical role in balancing creativity and determinism in the model's responses. Adjustments to these parameters can influence experimental outcomes significantly, making it important to document and justify these choices in the context of your research.

APPENDIX B

Extended Experimental Results

The second appendix provides additional details on the experimental results, including extended tables and figures not fully detailed in the main text.

B.1 Confusion Matrices and Metrics

We include the extended tables, charts, and figures that break down the model's performance for each risk category, showing precision, recall, and F1 scores at a finer level of detail.

B.2 Intra-Annotator Agreement

B.2.1 Fleiss' Kappa Interpretation

κ **Range**	**Interpretation**
< 0	Poor agreement
0.0–0.20	Slight agreement
0.21–0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–1.0	Almost perfect agreement

Table B.1. Interpretation of Fleiss' Kappa Values [31]

B.2.2 Full Intra-Annotator Agreement Results

Below are the complete intra-annotator agreement results, including the detailed distributions of `Yes` and `No` responses for both the Detection and Evaluation steps, as well as the aggregated Fleiss' Kappa values for each risk category.

B.2.2.1 Detection Answer Distribution per Annotator

```
Annotator 1:
  Column: Market Dynamics - a -> Yes: 998, No: 23556 (Total valid answers: 24554)
  Column: Intra-Industry Competition - a -> Yes: 442, No: 24110 (Total valid answers: 24552)
  Column: Technology Risk - a -> Yes: 1577, No: 22979 (Total valid answers: 24556)
  Column: Regulatory Framework - a -> Yes: 5032, No: 19502 (Total valid answers: 24534)

Annotator 2:
  Column: Market Dynamics - a -> Yes: 944, No: 24632 (Total valid answers: 25576)
  Column: Intra-Industry Competition - a -> Yes: 438, No: 25134 (Total valid answers: 25572)
  Column: Technology Risk - a -> Yes: 1649, No: 23928 (Total valid answers: 25577)
  Column: Regulatory Framework - a -> Yes: 5063, No: 20499 (Total valid answers: 25562)

Annotator 3:
  Column: Market Dynamics - a -> Yes: 937, No: 22028 (Total valid answers: 22965)
  Column: Intra-Industry Competition - a -> Yes: 433, No: 22533 (Total valid answers: 22966)
  Column: Technology Risk - a -> Yes: 1575, No: 21392 (Total valid answers: 22967)
  Column: Regulatory Framework - a -> Yes: 4687, No: 18263 (Total valid answers: 22950)
```

B.2.2.2 Evaluation Answer Distribution per Annotator

```
Annotator 1:
  Column: Market Dynamics - a -> Yes: 873, No: 23681 (Total valid answers: 24554)
  Column: Intra-Industry Competition - a -> Yes: 228, No: 24324 (Total valid answers: 24552)
  Column: Technology Risk - a -> Yes: 1415, No: 23141 (Total valid answers: 24556)
  Column: Regulatory Framework - a -> Yes: 4332, No: 20202 (Total valid answers: 24534)

Annotator 2:
  Column: Market Dynamics - a -> Yes: 703, No: 24873 (Total valid answers: 25576)
  Column: Intra-Industry Competition - a -> Yes: 74, No: 25498 (Total valid answers: 25572)
  Column: Technology Risk - a -> Yes: 1291, No: 24286 (Total valid answers: 25577)
  Column: Regulatory Framework - a -> Yes: 3047, No: 22515 (Total valid answers: 25562)

Annotator 3:
  Column: Market Dynamics - a -> Yes: 693, No: 22272 (Total valid answers: 22965)
  Column: Intra-Industry Competition - a -> Yes: 79, No: 22887 (Total valid answers: 22966)
  Column: Technology Risk - a -> Yes: 1253, No: 21714 (Total valid answers: 22967)
  Column: Regulatory Framework - a -> Yes: 2875, No: 20075 (Total valid answers: 22950)
```

B.2.2.3 Aggregated Intra-Annotator Agreement (Fleiss' Kappa)

```
Column: Market Dynamics - a
  Aggregated Detection-level Fleiss' Kappa:  0.8460
```

```
  Aggregated Evaluation-level Fleiss' Kappa: 0.8035

Column: Intra-Industry Competition - a
  Aggregated Detection-level Fleiss' Kappa:  0.8389
  Aggregated Evaluation-level Fleiss' Kappa: 0.4406

Column: Technology Risk - a
  Aggregated Detection-level Fleiss' Kappa:  0.8991
  Aggregated Evaluation-level Fleiss' Kappa: 0.8671

Column: Regulatory Framework - a
  Aggregated Detection-level Fleiss' Kappa:  0.9291
  Aggregated Evaluation-level Fleiss' Kappa: 0.7990
```

B.3 Fundamental Score Label Distribution

Below you'll find the number of analyst identified risk factors by label. First for the full dataset, and then for the sampled dataset. In both tables, we've marked in bold the risk labels that we put our focus on in this work.

Label	Count
Business Model.a	45
Business Model.b	11
Business Model.c	15
Business Model.d	8
Business Model.e	72
Competitive Positioning.a	44
Competitive Positioning.b	19
Competitive Positioning.c	21
Intra-Industry Competition.a	**49**
Intra-Industry Competition.b	75
Intra-Industry Competition.c	39
Management & Ownership.a	29
Management & Ownership.b	67
Management & Ownership.c	61
Management & Ownership.d	43
Management & Ownership.e	44
Market Dynamics.a	**46**
Market Dynamics.b	46
Market Dynamics.c	22
Regulatory Framework.a	**49**
Regulatory Framework.b	20
Technology Risk.a	**70**
Technology Risk.b	11

((a)) Full Dataset

Label	Count
Business Model.a	16
Business Model.b	0
Business Model.c	2
Business Model.d	1
Business Model.e	18
Competitive Positioning.a	8
Competitive Positioning.b	2
Competitive Positioning.c	5
Intra-Industry Competition.a	**6**
Intra-Industry Competition.b	20
Intra-Industry Competition.c	8
Management & Ownership.a	4
Management & Ownership.b	20
Management & Ownership.c	13
Management & Ownership.d	13
Management & Ownership.e	12
Market Dynamics.a	**7**
Market Dynamics.b	13
Market Dynamics.c	4
Regulatory Framework.a	**17**
Regulatory Framework.b	8
Technology Risk.a	**26**
Technology Risk.b	3

((b)) Sampled Dataset

Table B.2. Fundamental Score Distribution (Full and Sampled Datasets)

B.4 Risk Detections (TP/FP) Aggregated by Section Title

Find the full version of Table 7.7 in the `./_explorative_data_analysis` folder: `/risk_by_title_tp_fp_output.csv`. The table illustrates how text length and risk detections (true positives and false positives) are aggregated by section. This view helps identify which sections consume the most tokens and where the LLM is most likely to detect or over-detect risks.

B.5 Local LLM Results and Shortfalls

We also ran tests with smaller local models (e.g., `Llama 3.2 3B`). While they could identify some relevant risks, they often struggled to maintain strict output formats (such as JSON) and frequently required multiple retries to produce valid responses. Below is a snapshot of confusion matrix metrics, along with the percentage of successful JSON outputs after up to three retries:

```
=== Parsing Success Statistics ===
                    column_name  total_valid_rows  parse_errors  parse_successes  success_rate
0            Market Dynamics - a              5217          3180             2037      0.390454
1  Intra-Industry Competition - a              5216          3089             2127      0.407784
2       Regulatory Framework - a              5217          3050             2167      0.415373
3            Technology Risk - a              5216          2851             2365      0.453413

=== Per-Label Confusion Matrix with Metrics ===
                      Label  TP  FP  FN  TN  Precision    Recall  F1 Score  Accuracy
0  Intra-Industry Competition.a  13  73   2  12   0.151163  0.866667  0.257426      0.25
1            Market Dynamics.a  28  60   1  11   0.318182  0.965517  0.478632      0.39
2       Regulatory Framework.a  18  71   6   5   0.202247  0.750000  0.318584      0.23
3            Technology Risk.a  19  72   1   8   0.208791  0.950000  0.342342      0.27
```

Figure B.1. *Output from experiments with `Llama 3.2 3B`, showing confusion matrix metrics and a roughly 40% success rate in adhering to the requested JSON schema after three retries.*

Local inference on GPUs can reduce reliance on external APIs, but it may require fine-tuning or advanced prompt-engineering strategies to match the reliability of larger, hosted LLMs.

For demonstration, we provide a modeling script and a shell script for submitting jobs to DTU's HPC cluster (`./_local_llm_modelling`). These resources show how GPU acceleration can speed up local LLM evaluations, although careful tuning is essential to ensure robust output formatting.

APPENDIX C

Extended Implementation Details

This appendix provides a deeper look at the repository, covering the main scripts, data structures, and operational logic. It is intended for readers who want to replicate, adapt, or extend the system in practice.

The project is hosted at `github.com/pierrehogenhaug/mester`.

C.1 Repository

- **`data/`**: Contains both raw and processed documents, as well as CSVs (e.g., parsed prospectus files, fundamental score data).
- **`scripts/`**: Holds entry-point scripts for each stage of the pipeline. For example:
  - `data_collection`: Scripts for scraping SharePoint and ESMA data, and for database extraction.
  - `data_processing`: Scripts for parsing prospectuses and merging SQL-based ground truth.
  - `analysis`: Scripts for running the two-step LLM analysis (detection and evaluation).
  - `evaluation`: Scripts for aggregating LLM outputs and comparing against analyst ground truth, including intra-annotator agreement.
- **`src/`**: Python modules encapsulating the core logic (e.g., PDF parsing, SharePoint/ESMA scraping, LLM prompt handling, evaluation metrics).
- **`Dockerfile`**: A multi-stage build file for reproducible environments.
- **`_explorative_data_analysis/`**: Scripts used for exploratory data analysis.
- **`_local_llm_modelling/`**: Demonstration scripts for local LLM inference (via llama.cpp and LangChain wrappers).

C.1.1 Data Collection

C.1.2 Database Extraction

- **Script**: `scripts/data_collection/run_database_utils.py`
- **Description**: Connects to SQL databases using the `capfourpy.databases` module, retrieves "Fundamental Score" data and RMS issuer information, and then merges them via functions defined in `src/data_processing/data_processing.py`.
- **Usage Example**:

```
python scripts/data_collection/run_database_utils.py \
    --rms_id <RmsId> [--no_csv]
```

- **Key Points**:
  - Supports optional filtering by a specific RMS ID.
  - Can output the merged data as a CSV file.

C.1.3 SharePoint Scraper

- **Script**: `scripts/data_collection/run_scrape_sharepoint.py`
- **Description**: Authenticates with the internal Capital Four SharePoint portal (using a custom subclass of `SharePoint` from `capfourpy`), then downloads PDFs based on metadata fields (e.g., RMS ID).
- **Usage**:

```
python scripts/data_collection/run_scrape_sharepoint.py
```

- **Key Points**:
  - Relies on the `capfourpy` library for secure authentication.
  - Uses metadata fields to correctly map downloaded PDFs to company IDs.

C.1.4 ESMA Scraper

- **Module**: `src/data_collection/scrape_esma.py`

- **Description**: Automates a headless browser session (using Selenium) to download prospectus PDFs from ESMA by querying with ISIN numbers. This module acts as a fallback when documents are not found via internal sources.
- **Key Points**:
  - Includes logic to handle multiple downloads and file naming conflicts.
  - Can be integrated into the pipeline via a custom run script or called directly.

C.2 Data Processing

C.2.1 Prospectus Parsing

- **Script**: `scripts/data_processing/run_parse_prospectus.py`
- **Core Functionality**:
  1. **PDF-to-Markdown Conversion**: Uses `pymupdf4llm` (and `PyMuPDF`) to convert PDFs into Markdown, preserving formatting cues (such as bold and italic).
  2. **Hierarchy Building**: Processes the Markdown text to detect top-level sections (e.g., `RISK FACTORS`), subsections, and sub-subsections.
  3. **Storage**: Saves each parsed text block with associated metadata (e.g., prospectus ID, original filename, heading levels) as CSV files.
- **Error Handling**: Logs and marks files where parsing fails or when key sections (such as risk factors) are missing.

C.2.2 Fundamental Scores and Ground Truth

- **Description**: Merges SQL-based fundamental score data (from `run_database_utils.py`) with the parsed subsections.
- **Implementation**:
  - Cleans and processes both datasets using functions in `src/data_processing/data_processing.py`.
  - Produces a final table where each row corresponds to a prospectus subsection along with the analyst-assigned risk labels.

C.3 LLM Analysis and Evaluation

C.3.1 LLM Analysis (Detection and Evaluation)

- **Script**: `scripts/analysis/run_analysis_two_step_llm.py`
- **Overview**:
  1. **Detection Step**: Module in `src/analysis/analysis_two_step_llm.py`) formulates a prompt combining a risk-related question with subsection text. The LLM is then invoked to return a JSON object indicating whether a risk is present (answer "Yes" or "No") and includes evidence.
  2. **Evaluation Step**: If detection returns "Yes" (or for further confirmation), a second prompt is issued using reference cases. This step may override the initial answer based on a detailed comparison.
- **Response Handling**:
  - JSON responses are parsed and validated using `Pydantic` models.
  - Retry logic is implemented to handle transient parsing errors.
- **Usage Example**:

```
python scripts/analysis/run_analysis_two_step_llm.py \
    --model_type openai --sample 3
```

C.3.2 Evaluation Against Ground Truth

- **Scripts**:
  - `scripts/evaluation/run_evaluate_llm.py`: Aggregates LLM outputs from one or more processed CSV files and computes confusion matrices, precision, recall, and other performance metrics.
  - `scripts/evaluation/run_evaluate_llm_intra_annotator.py`: Computes intra-annotator agreement using Fleiss' Kappa across CSV triple groups.
- **Key Points**:
  - **Aggregation**: Subsection-level labels are aggregated to the company level.
  - **Comparison**: The system compares LLM-assigned labels against analyst-provided ground truth.
  - **Agreement**: Supports repeated runs to measure intra-annotator consistency (Fleiss' Kappa).

C.4 Practical Notes on Usage and Configuration

1. **Command-Line Arguments**: Many scripts accept parameters such as:
   - `--rms_id` (e.g., in `run_database_utils.py`) to filter data.
   - `--model_type` (e.g., in `run_analysis_two_step_llm.py`) to select between a remote (OpenAI) or local LLM.
   - `--sample` to limit the number of prospectuses processed.
2. **LLM Integration**:
   - Supports both local LLMs (via `LlamaCpp` and `llama.cpp`) and remote models (via ChatGPT-4o-mini).
   - Responses are expected as JSON and are validated using `Pydantic`, with retry logic built in.
3. **Environment Configuration**: Many scripts add the project root to `sys.path` and rely on configuration (e.g., API keys in a `.env` file) to function correctly.
4. **Docker Setup**:
   - The provided `Dockerfile` builds a reproducible environment containing all required dependencies for scraping, parsing, and LLM inference.

C.5 Example Commands

Below is an example workflow illustrating how a user might run the entire pipeline:

1. **Database Extraction**

```
python scripts/data_collection/run_database_utils.py --rms_id 12345
```

2. **SharePoint Scraping**

```
python scripts/data_collection/run_scrape_sharepoint.py
```

3. **Prospectus Parsing**

```
python scripts/data_processing/run_parse_prospectus.py \
    --file data/raw/esma/Issuer123.pdf \
    --output data/processed/Issuer123/
```

4. **LLM Analysis (Detection and Evaluation)**

```
python scripts/analysis/run_analysis_two_step_llm.py \
    --model_type openai --sample 3
```

5. **LLM Evaluation Against Ground Truth**

```
python scripts/evaluation/run_evaluate_llm.py \
    data/processed/Issuer123_parsed.csv
```

6. **Intra-Annotator Agreement (Fleiss' Kappa)**

```
python scripts/evaluation/run_evaluate_llm_intra_annotator.py \
    --processed_root data/processed
```

C.6 Conclusion and Extensibility

The repository is designed to be modular; each step (data collection, processing, LLM inference, and evaluation) can be updated independently. By following the instructions and sample commands in this appendix, practitioners can:

- Reproduce the proof-of-concept pipeline.
- Adapt the system to new risk parameters or document types.
- Integrate more advanced LLMs or modify prompt templates as new models become available.

C.7 Software and Tools

C.7.1 Overview of the Software/Hardware Stack

Component	Technologies
Programming Language	`Python`, `SQL`
Data Processing	`Pandas`, `JSON`, `Pickle`
PDF Parsing	`PyMuPDF` (with `pymupdf4llm`), custom markdown parsers
Web Scraping & Automation	`capfourpy` (for SharePoint access), `Selenium` (for ESMA)
Database Interaction	`capfourpy.databases` (for SQL querying)
Utilities	Standard libraries (`os`, `sys`, `platform`), `TQDM`
LLM Integration	`langchain` (`ChatOpenAI`, `LlamaCpp`), `Pydantic`
Concurrency	`concurrent.futures`

Table C.1. *Overview of the technology stack used in this work.*

Device	Specifications
MacBook Pro (M1)	10-core CPU, 16-core GPU, macOS
HP Z2 Tower G5 Workstation	Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz (8 cores, 16 logical processors), x64-based PC
HPC Cluster (`gpuv100` node)	Nvidia Tesla V100 GPU, exclusive GPU process mode, system memory allocated per job requirements

Table C.2. *Hardware used for scraping, parsing, evaluation, and (local) LLM inference*

C.7.2 LLM Integration

Component	Details
Open Source	Llama 3.2 1B, 3B, Llama 2 7B, and Llama 30B via Hugging Face Using `llama.cpp` and LangChain's `LlamaCpp` wrapper
Closed Source	ChatGPT-4o-mini accessed via API
Prompt Engineering	Custom prompt templates for detection and evaluation
Response Handling	Responses are parsed into JSON and validated using Pydantic models
Errors & Logging	Retry logic for parsing failures and progress logging via WandB

Table C.3. *Overview of the LLM integration components.*

Component	Details
SQL Databases	**Azure:** ODBC Driver 18 for SQL Server on Linux **On-Prem:** SQL Server / ODBC Driver 17 for SQL Server on Linux Data accessed via SQL queries
File Storage	CSV files and metadata (from SharePoint) stored on Capital Four's OneDrive
HPC Storage	Temporary scratch storage (100GB allocated) for storing LLM model weights for local LLM inference

Table C.4. *Overview of data storage and retrieval components*

C.7.3 Data Storage and Retrieval

C.8 Additional Implementation Notes

Structured Output and Pipeline Integration: Intermediate results (e.g., raw LLM responses and validated JSON) are saved to CSV files after each inference step. This design choice enhances traceability, facilitates re-runs (in case of transient failures), and allows for robust error handling.

Local and HPC LLM Inference: In addition to remote inference via ChatGPT-4o-mini, the repository includes scripts for local model inference using llama.cpp. These scripts (found in `_local_llm_modelling/` and integrated via `run_analysis_two_step_llm.py`) can be adapted for HPC cluster submissions.

Modularity and Extensibility: Each pipeline component (data collection, processing, LLM analysis, evaluation) is designed to be updated independently. Researchers can easily extend the system by replacing modules or adjusting prompt templates without modifying the entire codebase.

Bibliography

[1] A. W. Lo, M. Singh, *et al.*, "From eliza to chatgpt: The evolution of natural language processing and financial applications.," *Journal of Portfolio Management*, vol. 49, no. 7, 2023.

[2] K. Du, Y. Zhao, R. Mao, F. Xing, and E. Cambria, "Natural language processing in finance: A survey," *Information Fusion*, vol. 115, p. 102755, 2025.

[3] Z. T. Ke, B. T. Kelly, and D. Xiu, "Predicting returns with text data," tech. rep., National Bureau of Economic Research, 2019.

[4] J. Fang-Klingler, "Impact of readability on corporate bond market," *Journal of Risk and Financial Management*, vol. 12, no. 4, p. 184, 2019.

[5] F. Li, M. Yang, and T. Zhang, "Does prospectus readability matter for bond issuance pricing? evidence from china," *Pacific-Basin Finance Journal*, vol. 80, p. 102074, 2023.

[6] H. Jing, J. Qiu, Y. Yao, and L. Wei, "The influence of bond prospectus sentiment on credit risk premium," *Procedia Computer Science*, vol. 221, pp. 474–481, 2023.

[7] R. K. Sharma, G. Bharathy, F. Karimi, A. V. Mishra, and M. Prasad, "Thematic analysis of big data in financial institutions using nlp techniques with a cloud computing perspective: A systematic literature review," *Information*, vol. 14, no. 10, p. 577, 2023.

[8] S. Kaur, C. Smiley, A. Gupta, J. Sain, D. Wang, S. Siddagangappa, T. Aguda, and S. Shah, "Refind: Relation extraction financial dataset," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3054–3063, 2023.

[9] A. Zaremba and E. Demir, "Chatgpt: Unlocking the future of nlp in finance," *Modern Finance*, vol. 1, no. 1, pp. 93–98, 2023.

[10] M. . Company, "Embracing generative ai in credit risk." `https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/embracing-generative-ai-in-credit-risk?utm_source=chatgpt.com#/`, 2024. Blog post, July 1, 2024.

[11] D. Feng, Y. Dai, J. Huang, Y. Zhang, Q. Xie, W. Han, Z. Chen, A. Lopez-Lira, and H. Wang, "Empowering many, biasing a few: Generalist credit scoring through large language models," *arXiv preprint arXiv:2310.00566*, 2023.

[12] Y. Nie, Y. Kong, X. Dong, J. M. Mulvey, H. V. Poor, Q. Wen, and S. Zohren, "A survey of large language models for financial applications: Progress, prospects and challenges," *arXiv preprint arXiv:2406.11903*, 2024.

[13] J. Lee, N. Stevens, S. C. Han, and M. Song, "A survey of large language models in finance (finllms)," *arXiv preprint arXiv:2402.02315*, 2024.

[14] X. Li, S. Chan, X. Zhu, Y. Pei, Z. Ma, X. Liu, and S. Shah, "Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? a study on several typical tasks," *arXiv preprint arXiv:2305.05862*, 2023.

[15] G. Son, H. Jung, M. Hahm, K. Na, and S. Jin, "Beyond classification: Financial reasoning in state-of-the-art language models," *arXiv preprint arXiv:2305.01505*, 2023.

[16] S. Roychoudhury, S. Sunkle, N. Choudhary, D. Kholkar, and V. Kulkarni, "A case study on modeling and validating financial regulations using (semi-) automated compliance framework," in *The Practice of Enterprise Modeling: 11th IFIP WG 8.1. Working Conference, PoEM 2018, Vienna, Austria, October 31–November 2, 2018, Proceedings 11*, pp. 288–302, Springer, 2018.

[17] W. Sherchan, S. A. Chen, S. Harris, N. Alam, K.-N. Tran, and C. J. Butler, "Cognitive compliance: Assessing regulatory risk in financial advice documents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13636–13637, 2020.

[18] J. Zhang and N. M. El-Gohary, "Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, p. 04015014, 2016.

[19] C. Hänig, M. Schlösser, S. Hamotskyi, G. Zambaku, and J. Blankenburg, "Nlp-based decision support system for examination of eligibility criteria from securities prospectuses at the german central bank," *arXiv preprint arXiv:2302.04562*, 2023.

[20] A. Garimella, A. Sancheti, V. Aggarwal, A. Ganesh, N. Chhaya, and N. Kambhatla, "Text simplification for legal domain:{i} nsights and challenges," in *Proceedings of the Natural Legal Language Processing Workshop 2022*, pp. 296–304, 2022.

[21] P. Törnberg, "Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning," *arXiv preprint arXiv:2304.06588*, 2023.

[22] Y. Zhu, P. Zhang, E.-U. Haq, P. Hui, and G. Tyson, "Can chatgpt reproduce human-generated labels? a study of social computing tasks," *arXiv preprint arXiv:2304.10145*, 2023.

[23] M. V. Reiss, "Testing the reliability of chatgpt for text annotation and classification: A cautionary remark," *arXiv preprint arXiv:2304.11085*, 2023.

[24] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[25] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[26] G. Tichy, K. Lannoo, O. Ap Gwilym, R. Alsakka, D. Masciandaro, and B. Paudyn, "Credit rating agencies: Part of the solution or part of the problem?," *Intereconomics*, vol. 46, no. 5, pp. 232–262, 2011.

[27] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[28] R. J. Gallo, M. Baiocchi, T. R. Savage, and J. H. Chen, "Establishing best practices in large language model research: an application to repeat prompting," *Journal of the American Medical Informatics Association*, vol. 32, no. 2, pp. 386–390, 2025.

[29] J. Moore, T. Deshpande, and D. Yang, "Are large language models consistent over value-laden questions?," *arXiv preprint arXiv:2407.02996*, 2024.

[30] E. Kim, I. Isozaki, N. Sirkin, and M. Robson, "Generative artificial intelligence reproducibility and consensus," *arXiv. org*, 2024.

[31] J. R. Landis and G. G. Koch, "An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers," *Biometrics*, pp. 363–374, 1977.

[32] Anthropic, "Building effective agents." `https://www.anthropic.com/research/building-effective-agents`, 2024. Blog post, December 19, 2024; Accessed: 2024-11-01.

{2.7182818284}ϑφερτυθιοπσδφγηξκλ