UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Hardo Post

Sentiment Analysis of Forward-Looking Statements from Annual Reports Using Large Language Models

Bachelor's Thesis (9 ECTS)

Supervisor:
Fredrik Milani

Tartu 2025

Sentiment Analysis of Forward-Looking Statements from Annual Reports Using Large Language Models

Abstract:

This thesis presents a system for extracting and analyzing forward-looking statements from the annual reports of OMX Nasdaq Stockholm main list companies. The system uses large language models (LLMs), particularly Google Gemini 2.5 Pro, to identify forward-looking statements, classify them by type and theme, and assign sentiment labels. These statements are aggregated into concise company and sector summaries, enabling sentiment-based rankings and natural language querying via a chatbot.

A prototype was developed that combines report scraping, statement extraction, vector storage, and a web interface. Validation was conducted both manually and using a second LLM, confirming relevance and metadata accuracy. While the results were promising, challenges such as occasional misclassification, report retrieval issues, and the absence of a gold-standard dataset for testing, were noted. Still, the project demonstrates the viability of using LLMs for financial text analysis and highlights future development directions, including continuous data collection and analysis, improved model evaluation, and expanded chatbot functionality.

Keywords: natural language processing, large language models, sentiment analysis, annual reports, text extraction

CERCS: P176 Artificial intelligence

Lühikokkuvõte:

Käesolev lõputöö tutvustab rakendust, mis tuvastab, eraldab ja analüüsib tulevikku suunatud väiteid OMX Nasdaq Stockholmi põhinimekirja ettevõtete aastaaruannetest. Süsteem kasutab selleks suuri keelemudeleid, eelkõige Google Gemini 2.5 Pro mudelit, et määrata väidetele tüüp, ettevõtte või sektoripõhine kategooria ja sentiment. Väidetest koostatakse lühikesed ettevõtete- ja sektorite kokkuvõtted, millest tehakse ka nende sisu põhjal paremusjärjestus ning samuti kasutatakse neid väited ka juturobotile lisa konteksti andmiseks.

Arendati prototüüp, mis ühendab aruannete kogumise, väidete töötlemise, vektorpõhise andmesalvestuse ja veebiliidese. Väidete õigsust hinnati käsitsi ja teise keelemudeli abil. Ilmnes ka üksikuid probleeme kategooriate määramisel ja aruannete kogumisel. Kindlamaks testimiseks puudus märgendatud andmestik. Projekti tulemused näitavad, et suuri keelemudeleid on võimalik kasutada finantstekstide analüüsiks ning toovad esile ka võimaluse, et tulevikus võiks finantsandmeid jooksvalt koguda ja analüüsida ning juturoboti võimekust täiustada ja ühe suure keelemudeliga teist testida.

Võtmesõnad: loomuliku keele töötlus, suured keelemudelid, sentimendianalüüs, aastaaruanded, teksti ekstraheerimine

CERCS: P176 Tehisintellekt

Contents

1 Introduction

Anticipating future developments is a fundamental need across many domains. Whether preparing for weather conditions or planning major life decisions such as purchasing a home, individuals routinely rely on forecasts to guide their choices. In more complex environments such as finance and business, decision-makers face questions like whether to invest, hire, expand, or instead reduce operations and costs. These decisions require informed expectations about the future state of the economy.

One way to inform such expectations is by measuring forward-looking sentiment—that is, the collective outlook of people or organizations regarding future economic conditions. According to Frohm and Tillin (2015), gathering this sentiment directly from businesses and households through interviews and surveys can yield useful indicators. They emphasize that forward-looking sentiment data helps capture expectations and intended behavior before actual economic outcomes such as GDP are observed. Interviews, in particular, can reveal issues that structured surveys might miss. As the authors note:

"Asking questions about something you know nothing about is by its very nature impossible. Neither those who design the survey nor the interviewer know what they should ask about. [...] It makes it possible to discuss matters that the respondent thinks is important but that are not directly covered by the list of questions for the interviews" (Frohm & Tillin, 2015, p. 3).

In this context, forward-looking statements disclosed by companies in their annual reports—particularly in management discussion and analysis sections or CEO letters—can be viewed as naturally occurring equivalents to interviews. These statements are not limited by pre-formulated survey questions, and they allow managers to express their views on future prospects, both for the company and the broader sector or economy. As such, they form a valuable data source for analyzing business sentiment regarding future conditions.

However, manually identifying and interpreting such forward-looking statements across hundreds of documents is time-consuming and resource-intensive. Recent advancements in large language models (LLMs) open up new possibilities for automating this process. LLMs can potentially extract, categorize, and analyze forward-looking content from financial documents with little or no fine-tuning, making them a promising tool for scalable sentiment analysis.

This study focuses on public annual reports of 359 companies listed on the stock market of OMX Nasdaq Stockholm, in the main list. The analysis is limited to forward-looking statements, meaning that historical or present content is excluded. Reports from the financial year ending in 2024 are prioritized, although slight variations may occur depending on individual companies' reporting calendars.

The main objective of this thesis is to explore how a general-purpose large language model (LLM) can be used to extract, classify, and label forward-looking statements from company annual reports using prompt-based instructions, and to design a system that summarizes and visualizes this information, with an interactive chatbot interface that allows users to explore sentiment-based company and sector insights.

By demonstrating how LLMs can be used to analyze forward-looking sentiment at scale, this study contributes to both financial communication research and practical NLP applications. It explores how narrative disclosures—otherwise difficult to analyze—can be transformed into structured, searchable insights to support investors, analysts, and economic researchers.

The thesis is organized as follows. Chapter 2 outlines the theoretical background of sentiment analysis, financial disclosures, and large language models. Chapter 3 presents the methodology, including data collection, prompt engineering, model configuration, and validation approach. Chapter 4 describes the technical implementation of the system, covering data processing, storage, and user interface development. Chapter 5 presents the results, including extracted statements, validation outcomes, and cost analysis. Finally, Chapter 6 provides a discussion of the findings, limitations of the approach, and directions for future work.

OpenAI's ChatGPT (GPT-4, accessed via chat.openai.com, spring 2025) was used during the thesis writing process to assist with text formatting, phrasing, and editing for improved readability. The content, structure, and analytical interpretations are the author's own.

2 Theoretical Foundation

The purpose of this chapter is to provide a theoretical and methodological foundation for sentiment analysis in the context of financial reporting. First, it introduces the general concept of sentiment analysis and outlines its key challenges. It then focuses on how sentiment analysis is applied in financial domains, highlighting both the potential and the limitations of existing approaches like FinBERT.[1] Finally, the chapter discusses the importance of forward-looking statements in annual reports and motivates the use of large language models as a more flexible and context-aware tool for capturing sentiment in financial reports.

2.1 Overview of Sentiment Analysis

According to the dictionary, sentiment refers to a feeling or attitude toward something, and analysis means breaking down and examining a particular object, field, or situation (EKI Combined Dictionary, 2024). Based on this, sentiment analysis can be defined as the examination of an individual's or a group's feelings, attitudes, or opinions toward a particular event, situation, or domain. A challenging aspect of sentiment analysis lies in measuring human emotions and attitudes. In a simple case, a person may respond with a "yes" or "no," or "like" or "dislike." In more complex cases, it becomes necessary to assess varying levels and nuances of emotions, which makes the task more difficult.

Traditionally, sentiment has been assessed manually by humans, which is a time-consuming process and may also be subjective (Borromeo & Toyama, 2021). Due to these limitations, automated methods that can quickly process large volumes of data have become preferable. Automated sentiment analysis is part of the field of Natural Language Processing (NLP). It deals with detecting and classifying emotions, opinions, and attitudes in texts (Gavali & Shrigave, 2024). However, Borromeo and Toyama argue that automated analysis may still fall short of human-level accuracy. Gavali and Shrigave also found that the limitations of automated sentiment analysis are linked to its complexity. While humans can intuitively perceive emotions and attitudes in text, it is difficult for computers to interpret ambiguity, sarcasm, and context-dependent cues (Gavali & Shrigave, 2024). Despite these drawbacks, in certain fields, a fast but less accurate result may be more valuable than an accurate result that is available only later.

Sentiment analysis has applications across a wide range of domains, including medicine, politics, economics, and finance. In medicine, it can be used to analyze patient feedback, while in economics, it helps assess consumer attitudes and market trends (Medhat et al., 2014).

Finance is a domain where sentiment analysis can be used in different areas. In the stock market, investor sentiment may play a key role in determining market volatility (Narayanan et al., 2023). Thus, sentiment analysis can support strategic decision-making by providing insights that may not be captured by other analytical methods. When conducted automatically, it enables the quick analysis of large datasets.

[1] FinBERT https://onlinelibrary.wiley.com/doi/full/10.1111/1911-3846.12832

An example of sentiment analysis in the financial domain is the work by Pfeifer and Marohl (2023), who developed CentralBankRoBERTa, a fine-tuned version of the RoBERTa model, tailored for central bank communications. Using a manually labeled dataset of over 15,000 sentences from the Federal Reserve, European Central Bank, and Bank for International Settlements, the authors trained two classifiers: one to identify the economic agent being addressed (e.g., households, firms, or governments), and another to classify the sentiment expressed toward that agent. Their findings show that this approach significantly outperforms traditional bag-of-words models and older machine learning techniques, and even demonstrates superior performance over FinBERT on sentiment classification tasks. However, this performance comes at the cost of substantial manual annotation, which can be time-consuming and resource-intensive. While CentralBankRoBERTa is a contextual model with good results, it belongs to a generation of encoder-only transformer models and lacks the broader capabilities of modern generative large language models (LLMs) such as ChatGPT or DeepSeek.

The Bank of International Settlement study (Gambacorta et al., 2024) demonstrates that RoBERTa-based models, when adapted to the central banking domain, perform exceptionally well in sentence-level tasks such as classifying monetary policy stance. These domain-specific models consistently outperform general-purpose language models like ChatGPT or LLaMA when those are used without additional training. Their strength lies in their ability to produce accurate, fine-tuned classifications based on carefully labeled training data. However, when the task requires understanding longer, more complex texts or capturing the overarching message across multiple sentences or paragraphs — such as in full central bank speeches or policy news articles — large generative language models show a clear advantage. Thanks to their broader pretraining and extended context windows, models like ChatGPT-4 and LLaMA-3 70B can more effectively interpret nuanced meaning and dependencies across longer passages, even without fine-tuning. This distinction highlights a trade-off between task-specific precision and general contextual reasoning across different model types.

2.2 Technologies for Automated Sentiment Analysis

Various technologies and methods are used in automated sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning models. Lexicon-based methods rely on predefined dictionaries to identify, for example, positive and negative words (Gavali & Shirgave, 2024).

Machine learning models learn from training data how to accurately detect sentiment in text. Deep learning models, such as convolutional and recurrent neural networks, are capable of automatically detecting complex patterns in text (Gavali & Shirgave, 2024). According to Gavali and Shrigave, transformer-based generative models like BERT and GPT have introduced significant advancements in sentiment analysis, as they can interpret context and semantic relationships. These models are particularly successful at binary sentiment classification, but they struggle with identifying finer emotional nuances—such as how strongly or emphatically a word expresses a certain feeling or attitude. For example, large language models may not accurately assess how modifiers like “very” or “slightly” influence sentiment, as their training data may lack sufficient specificity. Gavali and Shrigave argue that

traditional neural networks, fine-tuned for specific datasets, can offer higher accuracy for fine-grained sentiment analysis. Their advantage lies in the ability to optimize the model for specific goals and domains (Gavali & Shirgave, 2024).

Although the above arguments favor the use of fine-tuned neural networks for narrowly focused and precision-requiring tasks, large language models using Retrieval-Augmented Generation (RAG) can offer significant advantages when the task is broader in scope. For example, when the analysis also involves numerical financial data in addition to text input.

2.3 Large Language Models in Sentiment Analysis

Large language models such as GPT-4 are particularly suitable for sentiment analysis due to their ability to process large and complex volumes of text and to interpret context-dependent relationships. For analyzing financial data and reports, it is crucial that these models can understand both unstructured and semi-structured data. For example, large language models supported by context-extending techniques like Retrieval-Augmented Generation can be provided with additional real-time information during analysis (Alharbi et al., 2024). Therefore, a large language model does not need to be pre-trained for a specific task; its broad training, combined with real-time contextual input, allows the model to deliver more accurate and up-to-date analyses.

In the financial sector—such as when analyzing reports from the Bank of Estonia or the European Central Bank—it is important to interpret both textual and numerical information. Large language models can combine the analysis of key sentiment-expressing terms like "positive growth" or "high inflation risk" with quantitative indicators such as GDP and inflation, thus offering a more comprehensive overview than purely text-based sentiment analysis. These models can also automatically compare a new report with previous ones, identifying similarities and differences in messaging and thus aiding decision-makers and analysts in their strategic decisions (Narayanan et al., 2023; Alharbi et al., 2024).

Using large language models is also effective for multilingual analysis, such as when evaluating reports that address the economic environment of the European Union. These models can operate in multiple languages and produce consolidated results, which is vital when assessing complex and international economic situations (Narayanan et al., 2023; Alharbi et al., 2024).

2.4 The Value of Forward-Looking Statements in Financial Reports

Forward-looking statements are textual disclosures within annual reports that provide insight into a company's expectations, strategies, and potential risks related to future performance. Unlike backward-looking financial figures, forward-looking statements are intended to inform stakeholders—such as investors, analysts, and regulators—about how firms perceive and plan for future developments in uncertain environments. These statements typically appear in sections such as the Management Discussion and Analysis or CEO Letters and may cover topics ranging from anticipated market trends and investment plans to strategic goals, earnings forecasts, and risk factors.

Forward-looking disclosures are widely viewed as important sources of insight into how management frames the company's future. These statements can help reduce information asymmetry, signal management's confidence, and influence investor expectations about cash flows, risk, and performance (Hassanein et al., 2019). However, their usefulness often depends on how specific, updated, and verifiable they are. If forward-looking content remains unchanged year to year, or if it uses overly general or "boilerplate" language, its value to investors may be limited. On the other hand, updates in such disclosures—especially in response to performance shifts—can signal meaningful changes in strategic direction or financial outlook (Hassanein & Hussainey, 2015).

Empirical research on UK companies shows that forward-looking statements are especially informative for low-performing firms. Managers of such firms are more likely to provide extensive forward-looking commentary, likely to mitigate concerns and signal corrective strategies. In contrast, high-performing firms tend to rely more on their current financial performance and may offer fewer forward-looking disclosures, which investors may already perceive as credible. Moreover, the effect of forward-looking statements on investor valuation appears stronger when the firm is audited by one of the Big Four auditing firms, suggesting that perceived credibility plays a key role in how such statements are received (Hassanein et al., 2019).

Because forward-looking statements are unaudited and often qualitative, they pose interpretive challenges. Their tone and content may reflect not only genuine expectations but also impression management strategies (Hassanein & Hussainey, 2015). This complexity makes them particularly interesting for sentiment analysis, especially when using large language models that can detect nuance and evaluate meaning across longer and more subtle stretches of text. In this thesis, forward-looking statements are treated as a valuable input for sentiment analysis, offering a lens through which to understand how companies describe future prospects—and how those descriptions might differ based on context, performance, and strategic intent.

3 Methodology

This chapter presents the methodology used to develop and test a system for extracting and analyzing forward-looking statements from company annual reports using a general-purpose large language model (LLM). It outlines the implementation pipeline, including data collection, PDF processing, prompt engineering, LLM-based classification, and vector storage. It also describes how the extracted information is visualized and made accessible through a chatbot interface. The final section addresses how the system's output was evaluated.

3.1 Research Objective and Design

The main objective of this thesis is to explore how a general-purpose large language model (LLM) can be used to extract, classify, and label forward-looking statements from company annual reports using prompt-based instructions, and to design a system that summarizes and visualizes this information, with an interactive chatbot interface that allows users to explore sentiment-based company and sector insights.

To achieve this, a system-oriented, exploratory design is adopted. The methodology involves building an end-to-end data processing pipeline that integrates LLM-based extraction, prompt iteration, embedding generation, and structured storage using a vector-compatible database. The extracted statements are categorized by type (company or sector), assigned to predefined thematic categories, and labeled by sentiment. These outputs are then used to generate company- and sector-level summaries and rankings, which are presented to users through a web-based interface and an LLM-supported chatbot. The chatbot interface is designed to enable user-friendly querying and interpretation of the extracted content. This supports the objective of transforming the data into an accessible, interactive format for end users

While the system is not benchmarked against a gold-standard labeled dataset, a small sample of outputs is manually reviewed and evaluated using another LLM to estimate the quality of the extraction, classification, and sentiment labeling. This validation step complements the practical aim of designing an accessible system that supports user interaction and exploration of company and sector sentiment.

3.2 Data description and availability

This study uses publicly available annual reports from companies listed on the OMX Nasdaq Stockholm[2] main list as its data source. These documents serve as the basis for extracting forward-looking statements that reflect how company leadership views future risks, opportunities, and economic conditions. However, the available reports differ significantly in structure, length, and scope, which introduces important considerations for the data extraction process and large language model (LLM) performance.

The most common report types encountered are:

- Annual Reports: These typically focus on financial performance, management commentary, and strategic planning. They contain essential sections such as the CEO letter

[2] OMX Nasdaq Stockholm

and Management Discussion and Analysis (MD&A), where forward-looking statements are frequently found.

- Annual and Sustainability Reports: In addition to standard financial information, these reports include extensive sections on environmental, social, and governance (ESG) topics. While these sections can provide valuable context, only those ESG-related discussions that are expected to affect financial outcomes are considered relevant in this study. Notably, some companies no longer publish traditional annual reports and instead provide only combined sustainability reports, which can exceed 200 pages. These longer reports pose challenges for LLM processing due to their size relative to the model's context window.

- Q4 and Year-End Reports: These vary widely in length, ranging from approximately 10 to 80 pages, and often provide a preliminary summary of the year's performance. In some cases, companies publish these as brief reports before releasing a full annual or sustainability report, but for others this remains as the full year report. The presence and depth of forward-looking statements in these reports can vary considerably.

This variation in report formats means that not all companies offer the same volume or richness of forward-looking content. Reports that lack CEO letters or detailed MD&A sections may provide limited material for sentiment analysis.

Another complexity arises from differing fiscal year structures. While many companies report based on the calendar year ending December 31, others operate on financial years that end in March, June, or September. As a result, not all reports available at the end of 2024 reflect a full calendar year. Additionally, even for companies with calendar-year fiscal periods, annual reports are typically published in the first quarter of the following year—often between February and April—but not all companies release them at the same time. In some cases, the most recent report at the time of analysis may be a shorter interim or year-end report with limited forward-looking content. These temporal inconsistencies affect the comparability and completeness of data across companies.

The most valuable forward-looking content tends to appear in management commentary, CEO letters, and the MD&A section, where companies discuss anticipated risks, sector developments, investment plans, and broader macroeconomic conditions. These sections are therefore prioritized in the extraction process, as they provide the most relevant material for sentiment analysis at both the company and sector level.

The annual reports that will be used in this study are publicly available on each company's investor relations website. However, the structure and layout of these websites vary significantly, which makes it impractical to build a single web scraper capable of reliably accessing reports from all individual sources. To address this, a more efficient approach is to find and utilize centralized financial news and disclosure platforms that aggregate company announcements and reports in a consistent format. Developing a scraper for just this type of platforms seems significantly more manageable and ensures coverage of a substantial portion of the target company set. To find and retrieve relevant reports from these platforms, a clean and complete list of companies from the OMX Nasdaq Stockholm main list is first required.

3.3 LLM selection

A number of large language models (LLMs) are considered for this study, including Google Gemini 2.5 Pro, OpenAI's GPT-4o and o3-mini, as well as DeepSeek R1. Model selection is guided by factors such as availability, usability, context window capacity, and cost.

Initial exploration indicates that DeepSeek R1 may not be reliable for sustained use. At the time of evaluation, the platform required for API access is frequently unavailable, and purchasing credits is sometimes restricted due to limited server capacity. Since the study depends on stable model access throughout the processing phase, such reliability concerns lead to the decision to exclude DeepSeek R1 to avoid potential interruptions.

Usability is another key factor, particularly the model's ability to handle long financial documents—ranging from 10 to 200 A4 pages. This places emphasis on the context window. As explained by Bergman (2024), this is the amount of text a model can process in a single interaction. However, context windows are measured in tokens, not characters or words. One token typically represents about four characters in English (OpenAI, n.d.). For example, the 2024 Volvo Group Annual Report contains approximately 793,204 characters, which corresponds to around 204,260 tokens.

Prompt tokens must also be considered in the total token count, and sufficient room must remain for the model to return a complete response. If the input text and prompt exceed the model's limit, the model may truncate its output or fail to account for later parts of the input (Bergman, 2024). These considerations are particularly relevant in tasks involving structured extraction from long texts.

OpenAI's GPT-4o and GPT-o3-mini offer input limits of 128,000 and 200,000 tokens, respectively. While documents exceeding these limits can be split and submitted in batches, this approach may lead to a loss of contextual continuity. Additionally, OpenAI's usage-based pricing presents a cost consideration. GPT-4o is priced at $2.50 for input and $10 for output per million tokens. The o3-mini model is more affordable, but it shows weaker performance on long-context tasks, as shown in OpenAI's model specifications (OpenAI, 2025).

By contrast, according to Google Cloud (2025), Google Gemini 2.5 Pro (gemini-2.5-pro-exp-03-25) provides an input context window of up to 1,000,000 tokens, which may offer practical advantages for processing lengthy documents such as full annual and sustainability reports. The model has been evaluated using the Multiround Co-reference Resolution (MRCR) benchmark, designed to assess a language model's ability to retrieve specific outputs from long, multi-topic inputs (Gemini Team, 2024). Benchmark results reported by Kavukcuoglu (2025) indicate that Gemini 2.5 Pro achieves an average recall of 94.5% at 128,000 tokens and 83.1% at 1 million tokens, compared to 64.0% for GPT-4.5 and 61.4% for o3-mini at the 128,000 level. These findings suggest that Gemini 2.5 Pro is capable of maintaining contextual coherence over extended input sequences, a characteristic that is relevant for tasks involving the extraction of forward-looking statements from complex financial texts.

An additional benefit of Gemini 2.5 Pro is that it is currently free to use. Although the model is marked as experimental, no cost or usage restrictions are encountered during initial tests. As

confirmed by Google's developer documentation, the Gemini Pro and Gemini 2.5 Pro models are available at no cost for non-enterprise use at the time of writing (Google Cloud, 2025). Given its large context window, strong benchmark results, reliable availability, and lack of associated costs, Gemini 2.5 Pro is selected as the most suitable model for this study.

3.4 Prompt strategy

The initial prompt instructs the language model to extract all forward-looking statements from the given input text. For each extracted statement, the model is required to determine whether it is of type: company or type: sector. Statements that describe future expectations specifically related to the company—such as revenue growth, profitability, or product development—are classified as company-level. In contrast, statements referring to broader trends—such as consumer behavior, demand shifts, or regulatory risks that may affect multiple companies in the same industry—are classified as sector-level. Based on this classification, the model must assign each statement to one of the predefined thematic categories associated with either company or sector types. The company-related categories reflect internal performance indicators, while the sector-related categories reflect wider macroeconomic or industry-level trends.

In addition, the model is instructed to identify the primary sector in which the company operates, choosing from a list of eleven predefined sectors. While companies may be active in more than one sector, the model must select the dominant sector based on contextual cues in the report. Finally, the model is expected to assign a sentiment label—positive, neutral, or negative—to each forward-looking statement, and to extract both the company name and the reporting period year.

To help the language model understand the structure of the task, few-shot prompting is used. A small number of examples are included in the prompt to demonstrate how statements should be extracted, labeled, and categorized. These examples illustrate the intended format and provide concrete demonstrations of how to distinguish between company- and sector-level statements, assign categories, and apply sentiment labels. Although it is commonly assumed that such examples improve model performance by providing correct labels, research by Min et al. (2022) suggests that performance gains from few-shot prompting do not primarily depend on label correctness. Their findings show that even when example labels are randomly assigned, model performance decreases only marginally. This implies that few-shot prompting does not "train" the model in the traditional sense. Instead, it helps the model understand the nature of the task—such as sentiment classification or categorization—by exposing it to examples that resemble the input and output structure expected. While it is not essential for example labels to be perfectly matched to the content, it is important that they are drawn from the appropriate domain (e.g., economics or finance) and follow a recognizable and consistent format. This enables the model to produce responses that are structurally aligned with the task, even if it is not learning from the examples in a conventional supervised manner.

The initial version of the prompt will be iteratively refined based on qualitative feedback. First, the model will be tested on short input texts, such as individual paragraphs or excerpts from CEO letters. The output will be manually reviewed to assess the accuracy and relevance of the

extracted statements, classifications, and sentiment labels. Based on the results, adjustments to the prompt will be made. This process will then be repeated with longer inputs—such as full annual reports—and eventually with multiple reports. When working with multiple reports, a random sample of extracted statements will be reviewed to evaluate consistency and generalization of the prompt across different companies and report structures.

Once the extraction phase is complete, additional prompts will be developed for summarizing and ranking company and sector-level information. These prompts will follow the same design principles outlined earlier, with the key difference being that the input now consists of structured data retrieved from the database, rather than unstructured report text.

3.5 Data storage

The extracted forward-looking statements need to be stored in a database to enable their retrieval for various purposes, including presenting results in the front-end application and generating summaries of company and sector outlooks. In addition to traditional structured storage, there is also a need to store these statements in vector format. This is necessary for enabling interaction with the data through a large language model (LLM) integrated into a chatbot interface. Unlike conventional databases, which retrieve information through exact matches or structured queries, LLMs operate on natural language input that is unstructured and open-ended. To support this, the system must use similarity search, where user queries are converted into vector representations and compared to stored vectors to find the most semantically relevant content (Belcic, 2024).

3.6 Planned Frontend and User Interaction Design

The frontend of the application will be developed using a modern JavaScript framework such as Vue.js, React, or Angular, with the final selection depending on ease of integration, charting capabilities, and development considerations at the time of implementation. The frontend will prioritize clarity, responsiveness, and usability—particularly for presenting sentiment data and enabling interactive exploration of company and sector-level insights.

A global chat window is planned to be integrated into the interface as a slide-in panel from the right. When active, the main content area will dynamically shift to maintain visibility of charts and summaries. The chatbot will allow users to ask questions in natural language and receive grounded responses supported by similarity search over vectorized data. This interaction is intended to enhance the accessibility of insights without requiring technical queries or filters.

The main application views are expected to include:

- Overview: A dashboard summarizing sentiment across predefined company and sector categories.
- Sectors: A section for sector-specific sentiment summaries, performance comparisons, and rankings.
- Companies: A view displaying company-specific outlook summaries, extracted statements, and sentiment breakdowns.

This interface is designed to support intuitive navigation and meaningful engagement with the processed data, regardless of the final choice of frontend framework.

3.7 Data validation

To evaluate the quality of the extracted forward-looking statements and their associated metadata, this study implements a validation process using a second large language model. The purpose of the validation is to assess whether the extracted statements are correctly identified and appropriately labeled with respect to type, category, sentiment, and sector.

A random sample of ten annual reports was selected for validation. For each report, the original full-text content and the extracted statements were provided to a second LLM. The model was asked to determine whether each statement:

- Accurately appears in the source report,
- Is labeled with the correct type (company or sector),
- Has an appropriate thematic category,
- Reflects the correct sentiment,
- Is assigned to the correct sector.

The model returned a structured Java object as response for each statement, including five boolean fields (one for each criterion), the matched original statement text if found, and a brief comment explaining any errors or ambiguities.

All validation results were stored in a dedicated database table and later analyzed using SQL queries to identify systematic issues and overall accuracy levels. This automated validation approach allowed for consistent, scalable quality control across a complex dataset while preserving traceability and explanation for each decision.

4 Implementation

This chapter presents the practical implementation of the system described in the methodology. While Chapter 3 outlined the planned approach, the following sections detail the final solution as it was built. The system consists of a backend for processing and storing extracted statements, a frontend for visualizing results and enabling user interaction, and an LLM-based interface for answering queries using retrieval-augmented generation (RAG). This chapter begins by defining the functional and non-functional requirements derived from the research objectives, then describes the chosen system architecture and technology stack, and finally demonstrates the system's functionality through screenshots and a walkthrough of key features.

4.1 Requirements

The system is designed to address the research objective:

> *To explore how a general-purpose large language model (LLM) can be used to extract, classify, and label forward-looking statements from company annual reports using prompt-based instructions, and to design a system that summarizes and visualizes this information, with an interactive chatbot interface that allows users to explore sentiment-based company and sector insights.*

From this objective, the following system requirements are derived:

Functional Requirements:

R1. The system must automatically retrieve annual reports and extract text content to prepare inputs for LLM processing.

R2. The system must extract all relevant forward-looking statements from the annual report text using a large language model (LLM).

R3. Each extracted statement must be classified as either company-level or sector-level.

R4. The system must assign each statement a thematic category based on predefined labels.

> *These labels include separate sets of categories for company-level and sector-level statements. The classification process applies these categories automatically, and the results will be examined further in Chapter 5.*

R5. The system must label the sentiment of each statement as positive, neutral, or negative.

R6. The system must store each extracted forward-looking statement in the database, including metadata such as the company name, reporting period, type (company or sector), assigned sector (from a predefined list of 11), and a reference to the report from which the statement was extracted.

R7. The system must generate embedding vectors for each extracted statement, including key metadata (company name, report period, type, sector, report ID), to support similarity-based search and retrieval-augmented generation (RAG).

R8. The system must store metadata about each processed report in the database, including the company name, download link, report type, and processing status.

R9. The system must log token usage information for each processed report, including input tokens, output tokens, and total token count, in order to support cost estimation and efficiency analysis.

R10. The system must generate and store a textual summary of each company's outlook in the database. In addition to structured storage, each summary must also be embedded into vector format for use in semantic search and chatbot interaction.

R11. The system must generate sector-level outlook summaries by aggregating sector-type statements across multiple reports. Each summary must include associated ranking information (e.g., sentiment ranking or position within the sector list) and be stored in the database.

R12. The system must generate company and sector sentiment rankings based on extracted data and summaries.

R13. The system must support user interaction through a chatbot that answers natural language queries based on stored structured and vectorized data.

Non-Functional Requirements

R14. The system should respond within a reasonable time when processing user queries.

R15. The system should maintain usability and clarity in its interface, with compatibility for screen resolutions starting from 1378 × 768.

R16. The system should store data in both structured and vector formats to support similarity-based search and RAG.

R17. The system should be designed with cost-efficiency in mind, particularly in terms of LLM usage and token consumption.

R18. The system should rely solely on prompt-based interaction with the language model, allowing prompt instructions or classification schemes to be updated without requiring fine-tuning or retraining.

R19. The system should provide an intuitive and responsive user interface that supports interactive exploration of sentiment data without obstructing key visualizations.

R20. The chatbot interface should be globally accessible across all views and implemented in a way that allows users to interact with sentiment data without disrupting the surrounding content.

4.2 System Architecture and Technology Stack

This section describes the overall architecture and implementation of the developed system. It is divided into two parts. Section 4.2.1 outlines the selected technologies and tools used across the backend and frontend, along with justifications for each choice. Section 4.2.2 provides a walkthrough of the main system components and their interactions, following the logical flow from report retrieval and language model processing to data storage, embedding, and user-facing features. Together, these sections explain both the rationale behind the system design and how the system was built in practice.

4.2.1 Technology Stack

While Python has traditionally been the most widely used language for data science and machine learning—primarily due to its rich ecosystem of libraries such as NumPy, pandas, TensorFlow, and scikit-learn—this project takes a different approach. The core tasks of this thesis involve data extraction, classification, and sentiment labeling, which in earlier workflows would often be implemented in Python using custom models. However, with the rise of general-purpose large language models (LLMs), many of these tasks can now be delegated to external APIs via prompt-based interaction, reducing the need for domain-specific libraries.

For this reason, the system is implemented in Java using the Spring Boot framework, which offers robust support for web services and integration in production environments. The decision to use Java is also motivated by the availability of Spring AI, a framework that provides an abstraction layer over multiple LLM providers (e.g., OpenAI, Anthropic, Google). Spring AI enables developers to switch between models with minimal code changes by using a consistent syntax and configuration model. This flexibility is valuable for experimentation and future-proofing the system (Shopen, 2024). The system also uses Java Persistence API (JPA) for most database operations. JPA is a standard for object-relational mapping in Java applications, allowing developers to interact with relational databases using Java classes rather than raw SQL queries. It integrates naturally with Spring Boot and simplifies entity management and data access throughout the application.

For LLM, the uses Google Gemini 2.5 Pro as the primary LLM, integrated through the Spring AI framework, which provides a uniform interface for working with external LLM providers. The rationale for selecting Gemini 2.5 Pro—based on context window, cost, and reliability—is discussed in Section 3.3.

In order to support semantically meaningful interactions between the user and the language model, the system implements a retrieval-augmented generation (RAG) approach. RAG enables large language models to generate more accurate and grounded responses by retrieving relevant content from an external knowledge base—such as a vector database—at the time of the query. According to Bergman (2024), this architecture reduces the risk of hallucination and improves transparency by tying the model's answers to verifiable source content. In this system, RAG is used to augment user queries with relevant forward-looking statements or

summaries stored in vector format, enabling the chatbot to respond with domain-specific and contextually accurate information.

To enable this retrieval process, the stored content must be represented in a way that allows for semantic similarity comparison. This requires that textual inputs be converted into numerical vectors, where meaning is encoded in geometric proximity. This transformation is handled by an embedding model, which converts text into high-dimensional vectors that preserve semantic relationships. According to Google (2024), embeddings are dense vector representations of language that encode contextual meaning, allowing similar concepts to be located near each other in vector space. These vectors make it possible to search for meaning-based rather than keyword-based matches.

The initial implementation used Gemini's experimental model text-embedding-large-exp-03-07, which supports vector dimensions of 768, 1536, and up to 3072 (Google Developers Blog, 2024). While this model showed promising performance, it only allowed a single text input per request. As a result, the system quickly hit daily usage limits and could not scale effectively. The final implementation used Gemini's production model text-embedding-005, which supports up to 250 inputs per request and uses 768-dimensional vectors. This model provided the necessary balance of performance, throughput, and reliability. It was used to embed both individual forward-looking statements and company-level summaries.

To store and query these embeddings, the system uses PGVector, an open-source extension for PostgreSQL. PGVector adds native support for vector types and similarity search using cosine distance, inner product, or Euclidean distance. This enables efficient semantic retrieval in response to user queries—an essential component of the system's LLM-based chatbot interface. One of the key advantages of PGVector is that it allows vector-based search to be integrated directly within a relational database, eliminating the need for a separate vector database engine and reducing architectural complexity (PGVector, n.d.). PostgreSQL was selected as the base database system due to its maturity, strong performance, and widespread adoption. According to the 2024 Stack Overflow Developer Survey, it is the most commonly used database technology, with 48.7% of developers reporting its use in their projects (Stack Overflow, 2024). This makes PostgreSQL a stable and well-supported platform for storing both structured metadata and semantic vector embeddings.

For local development and testing, the PostgreSQL 16 database with the PGVector extension was hosted using Docker Desktop. This setup simplifies environment management and ensures the system can be easily transferred or redeployed on other machines. Using a Docker container also allowed for fast and reproducible setup, with consistent behavior across development sessions.

The frontend of the application is built using Vue.js, Vite, and TypeScript, with Tailwind CSS used for styling. Vue.js was selected due to its relatively simple syntax, fast runtime performance, and suitability for small- to medium-scale applications. It enables the development of responsive, component-based interfaces and is supported in the frontend development community (Navale, 2025). The application uses Vite as its build tool, which offers efficient module bundling and quick development feedback through optimized build

processes. The integration of TypeScript improves code maintainability and reliability by enabling static type checking during development. As Patel (2025) notes, combining Vue with Vite and TypeScript results in a modular, performant, and scalable frontend architecture that supports long-term project maintainability. For styling, the application uses Tailwind CSS, a CSS framework designed to support responsive and maintainable user interfaces. Tailwind provides low-level utility classes that allow developers to build custom designs without writing traditional CSS from scratch. This approach improves development speed and encourages consistency across components. As Tailwind Labs (n.d.) notes, the framework promotes scalability and adaptability in modern frontend applications while reducing the need for context switching between HTML and CSS files.

4.2.2 System Components

The data collection process began with compiling a list of companies from the OMX Nasdaq Stockholm main list. Company names were obtained using filtering tools available on the Nasdaq Nordic website. Since the manually copied data also included extraneous information, it was cleaned and standardized using Python and saved to a CSV file (omx_stockholm_companies.csv). The list included company names, sector names, and capitalization categories (small, mid, large), and was then imported into the application database using a dedicated Java service (NameImportService.java).

To retrieve annual reports, two centralized financial disclosure platforms were targeted:

- https://www.annualreports.com/
- https://mfn.se/all/s/nordic

Both platforms offer access to annual and financial reports across a wide range of OMX-listed companies and support filtering by country, industry, and report type.

Two Java-based scraper classes were developed for this purpose:

- AnnualReportScraper.java – using the Jsoup library to parse the HTML DOM and extract download links based on company names.
- MfnReportScraper.java – using Selenium WebDriver with ChromeDriver to simulate browser interaction and submit search queries for company names.

This scraper-based approach proved more efficient and scalable than scraping individual investor relations pages, which are highly inconsistent across companies. The download links are stored in a dedicated database table (report) and used to associate extracted statements with their source documents.

Each report entry in the system includes:

- Company name
- Sector
- Report download link
- Capitalization category (small, mid, large)
- Fiscal year

Once the report links were collected, the application used ReportDownloadService.java to download the PDF files to local storage. Although some LLMs support direct PDF input, this study relies on Google Gemini 2.5 Pro, which imposes a maximum upload limit of 50 MB per file. While this is sufficient for most documents, certain reports—particularly combined Annual and Sustainability Reports containing embedded graphics and design elements—may exceed this limit.

To ensure reliable and consistent processing, the application extracts only the textual content from each PDF before submitting it to the LLM. This task is handled by the PDFExtractorService.java class, which uses the Apache PDFBox library (org.apache.pdfbox) to parse and extract text. This preprocessing step reduces file size, removes unnecessary visual data, and ensures that the model receives only text as input.

After text extraction, the next step is processing the input through the large language model (LLM) and embedding model. To use these models with Java Spring Boot, configuration is required. For OpenAI models, only an API key and model name are needed. However, using Google Gemini models via the API requires additional setup through Google Cloud's Vertex AI. The developer must have an active Google Cloud account, install the Google Cloud CLI, authenticate via CLI, and configure project permissions. In the Spring Boot application, these settings—such as project-id, model name, and location—are defined in the application.yml configuration file.

The core logic for text processing and data storage is implemented in the processing package. The main class, StatementProcessingService.java, coordinates the workflow by invoking several helper methods. Before calling the LLM, the model's temperature parameter is explicitly set to 0.0. According to Gemini API documentation, lower temperature values (e.g., 0.0–0.3) yield more deterministic responses, making them suitable for classification tasks, while higher values (e.g., 0.8–1.0) are better suited for creative tasks such as brainstorming.

Reports are processed sequentially. After the text is extracted, it is combined with a structured prompt and submitted to the language model. The LLM response is expected in a predefined a Java record object.

Spring AI's ChatClient [3]enables this behavior by appending response formatting instructions to the prompt and handling the deserialization of the JSON response into Java objects. Each response also includes metadata about token usage—specifically, the number of input tokens, output tokens, and the total used in the interaction.

This process of extracting forward-looking statements began with smaller test iterations to manually validate the model's output and refine the prompt based on feedback. For example, consider the constructed statement:

"Inflation has been rising. It will be an impediment to our growth in 2025."

[3] https://docs.spring.io/spring-ai/reference/1.0/api/chatclient.html

In this case, the language model correctly extracted the forward-looking part—"It will be an impediment to our growth in 2025"—but the resulting sentence lacked clarity. The pronoun "it" was ambiguous, making the statement unsuitable for sentiment analysis because it did not specify the cause of the anticipated growth impediment. To address this, the prompt was updated to instruct the model to incorporate surrounding context when rewriting forward-looking statements. The goal was to ensure that extracted statements are self-contained and include the relevant subject or object, for example:

"Inflation will be an impediment to Company AS's growth in 2025."

This improvement in prompt clarity also influenced how company-specific language was handled. In one real-world case from Avanza Bank Holding AB's 2024 annual and sustainability report, the following statement appeared:

"We believe that young people today, who are used to doing all their banking digitally, will not want traditional Private Banking with an advisor in the future" (Avanza Bank Holding AB, 2024, p. 23).

Due to the prompt instructions to rewrite ambiguous references, the model extracted and reformulated the sentence as:

"Avanza believes that young people who are used to digital banking will not want traditional Private Banking with an advisor in the future."

While grammatically accurate and contextually appropriate, the language model labeled the statement as type: company, assumingly because the sentence explicitly included the company's name. However, the substance of the statement reflects a general expectation about industry-wide behavior. Such forecasts, if realized, would likely apply across the entire sector—not just to Avanza Bank. To correct this misclassification, the prompt was further revised to include explicit guidance on how to handle statements that contain company names but describe sector-level trends. Additional examples were provided to help the model distinguish more effectively between company-specific and sector-wide expectations. This example illustrates how prompt design evolved through iterative testing and analysis of edge cases.

In addition to extracting and classifying forward-looking statements, the model is also instructed to generate a company outlook summary during the same prompt. Early tests showed that total input token counts ranged from 50,000 to 200,000 and outputs ranged from 4,000 to 15,000 tokens—well within Gemini 2.5 Pro's input (1 million tokens) and output (64,000 tokens) limits. Generating summaries in the same request avoided the need for a second LLM call and reduced both execution time and potential cost.

Token usage tracking was implemented initially to monitor proximity to the model's token limits and to ensure that content was not being truncated. This monitoring later enabled cost estimation, simulating what the expense might have been if a commercial LLM were used in production. Token usage was stored to database.

Once processed, each report yields two types of data to store: statements and a company summary. These are saved in separate database tables (statement and report) using corresponding Java entities. Although the report table originally tracked report metadata, it was reused to store the company-level summaries, which align logically with the concept of a company-level report.

Each text entry also required embedding. The application uses the Gemini text-embedding-005 model, which accepts up to 250 inputs per request. An issue arose during embedding creation when a batch of 285 statements exceeded this limit. This was resolved by batching the embeddings into groups of 200.

The actual storage process is handled by the StorageService.java class. Because Java JPA (Java Persistence API) does not support storing float arrays or vectors natively, the application uses a JDBC client for database insertion. Embedding vectors, each 768 dimensions long, are first converted to string format, then re-cast as vectors during insertion via the JDBC layer.

Sector summaries and sentiment-based rankings are generated using a similar approach. These processes are implemented in the same processing package but are handled through separate prompts and do not involve vector embeddings. For each sector, a summary was generated by providing the LLM with all forward-looking statements categorized as type: sector and relevant to that particular sector. The model was instructed to generate a 300–400 word outlook summary, assign a sentiment tone (very positive, positive, neutral, negative, or very negative), and provide a 150–200 word justification explaining the assigned tone. These summaries were stored in the sector table in the database without embeddings.

Once the sector summaries were stored, the LLM was then prompted with all sector summaries and their associated sentiment justifications. Its task was to rank the sectors relative to one another based on their overall outlook and provide a rationale of one to three sentences explaining the ranking. Finally, within each sector, all company summaries were compared against the corresponding sector-level outlook. The LLM was tasked with identifying the top three companies in each sector that demonstrated the highest potential to outperform others in terms of future economic performance.

The frontend of the application was implemented using Vue 3 in combination with Vite to support efficient development and optimized builds. Styling was handled using Tailwind CSS, which enabled a responsive, utility-first design across various screen sizes. Interactive sentiment visualizations are rendered using ECharts, integrated via the vue-echarts wrapper. These charts dynamically display categorized sentiment data retrieved from the backend. Communication between frontend and backend is managed through a centralized api.ts module, using Axios to send and receive REST API requests.

The chatbot in the frontend of the applicaton is backed by a retrieval-augmented generation (RAG) workflow. When a user submits a query, such as *"How is Swedbank's outlook in 2025?"*, the message is first vectorized and used to perform a similarity search in the PGVector-extended PostgreSQL database. The most relevant matching statements or summaries are retrieved and

appended as context before the prompt is sent to the language model. This allows the LLM to respond with grounded, context-aware answers based on actual extracted data.

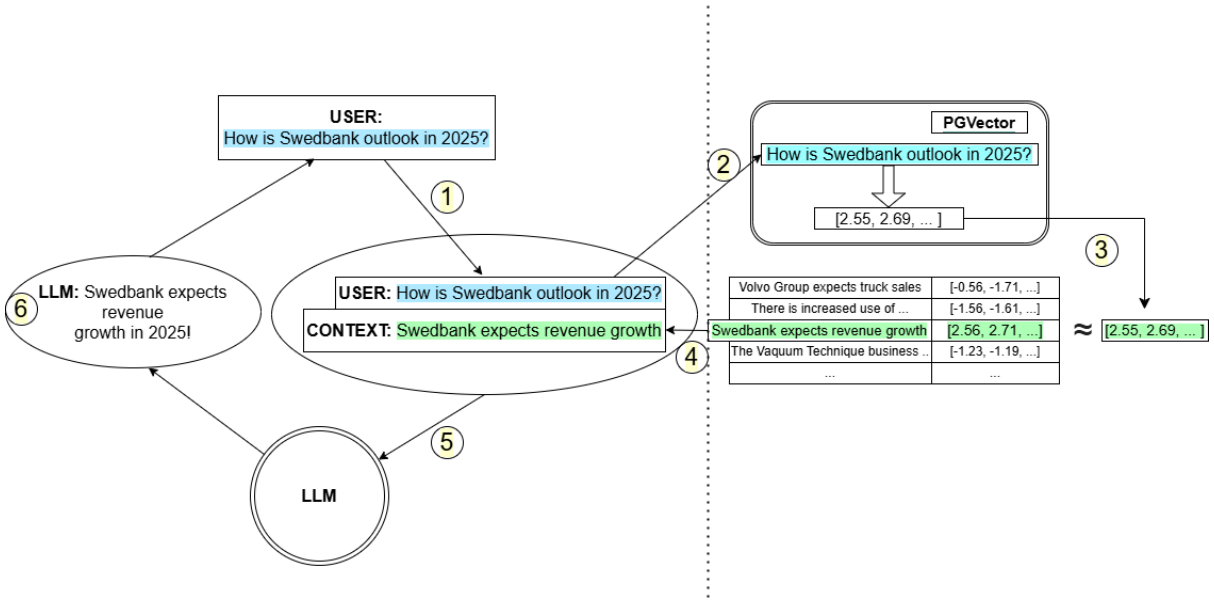


Figure 1. RAG flow

The RAG flow is illustrated in Figure 1, adapted and extended from IBM's conceptual model of RAG architecture (Belcic, 2024). The diagram illustrates how user input is converted into a vector, matched against stored embeddings, and then used to construct a context-enriched prompt for the model.

4.3 Demonstration

The application employs a global layout structure consisting of a persistent top navigation bar and a collapsible chat sidebar. The sidebar serves as the user interface for interacting with the system's LLM-based assistant, which supports natural language queries related to company and sector outlooks.

The default landing page of the application—referred to as the *Overview* view (see Figure 2)—displays two horizontal bar charts. Each bar represents a distinct thematic category of forward-looking statements. The top chart visualizes categories associated with company-level statements, while the bottom chart displays categories related to sector-level statements. These categories are described in detail in Chapter 5 (Results).

The length of each bar indicates the total number of forward-looking statements extracted for that category across all analyzed annual reports. Within each bar, sentiment distribution is color-coded: green represents the number of positive statements, grey indicates neutral statements, and red shows negative statements. When a user hovers over a bar, a tooltip appears displaying the exact count of statements by sentiment type.

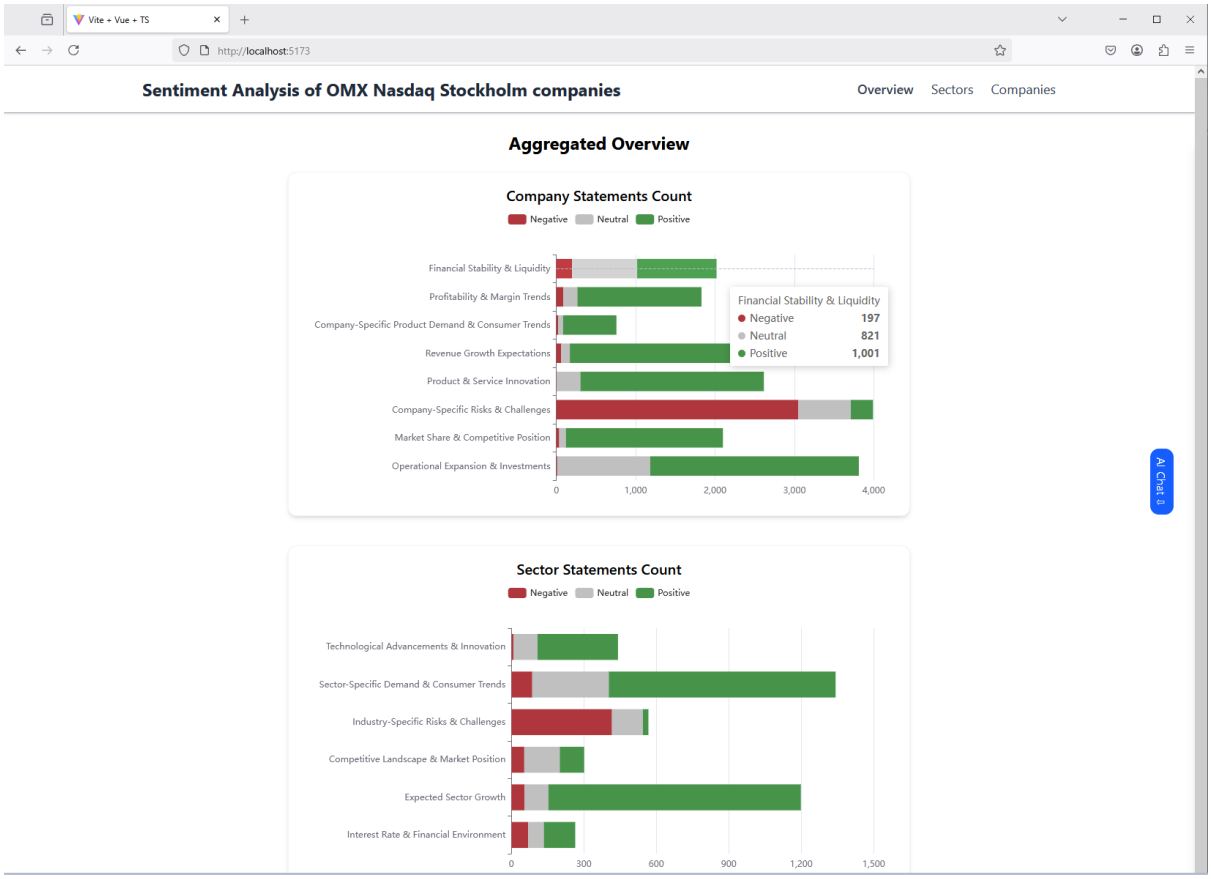


Figure 2. Opening view of the application

Clicking the *Sectors* navigation link in the upper-right corner of the top navigation bar (see Figure 3) opens the Sectors view. In this view, a series of charts is presented—each corresponding to one of the eleven predefined sectors. The charts are displayed vertically in the center of the screen. Each chart uses the same category structure as introduced in Figure 2

but filters the data to show only the statement counts relevant to the specific sector. The number shown in the upper-left corner of each sector chart indicates its overall ranking, where a lower number signifies a more favorable outlook for that sector. The sentiment label—positive, neutral, or negative—displayed in parentheses next to the ranking reflects the tone of the sector's outlook summary.

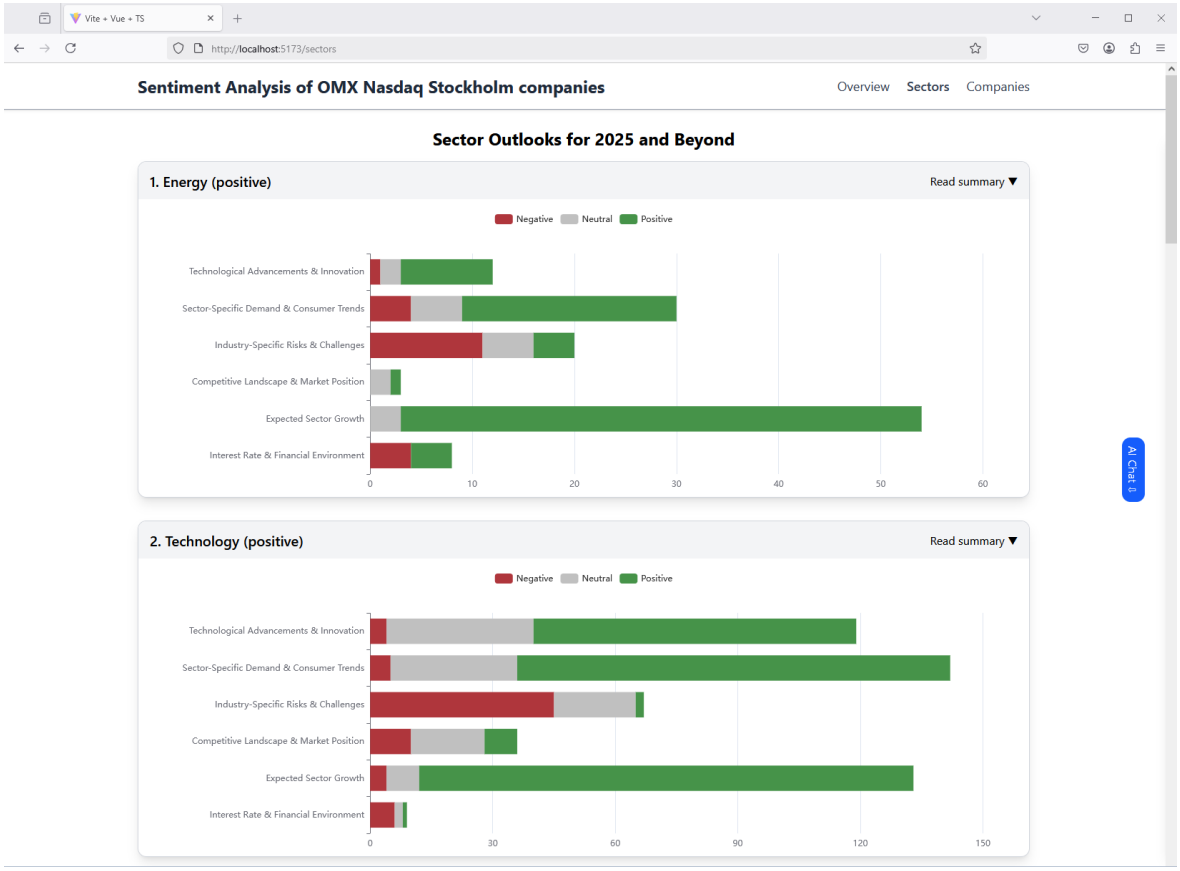


Figure 3. Sectors view

Clicking the *Read summary* link, indicated by a downward triangle in the upper-right corner of a chart expands the section downward to display the sector's written summary (see Figure 4).

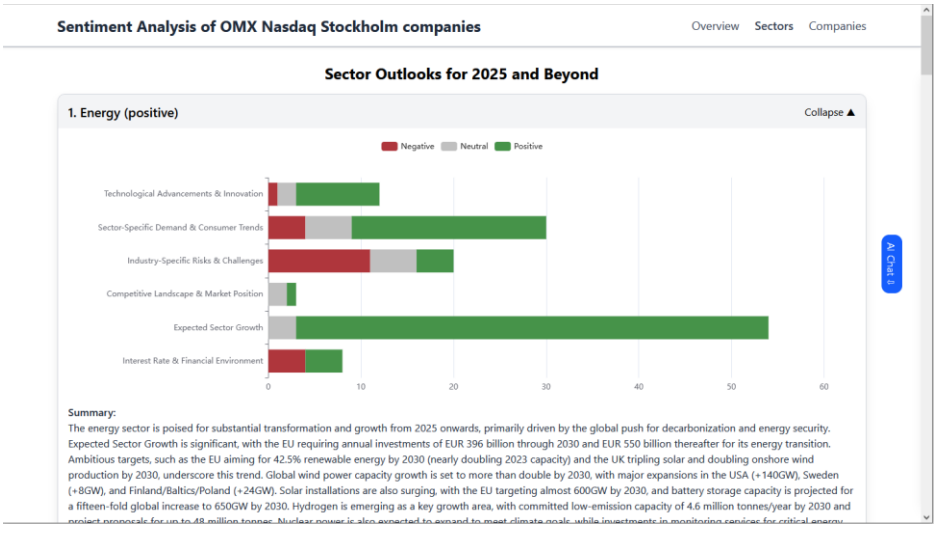


Figure 4. Sector chart expanded to show summary

The sector summary includes both a sentiment rationale, explaining the reasoning behind the assigned sentiment, and a ranking rationale, justifying the sector's position relative to others.

Navigating to the Companies view via the *Companies* navigation link in the top navigation bar opens a page dedicated to the top three companies in each sector (see Figure 5). Each section begins with an underlined sector heading, followed by individual charts representing the top three companies within that sector. The number preceding each company name in the upper-left corner of the chart indicates its ranking within the sector—the lower the number, the stronger the company's forward-looking outlook. The market capitalization category (Small Cap, Mid Cap, or Large Cap) appears in parentheses after the company name and is based on the company's stock market valuation.

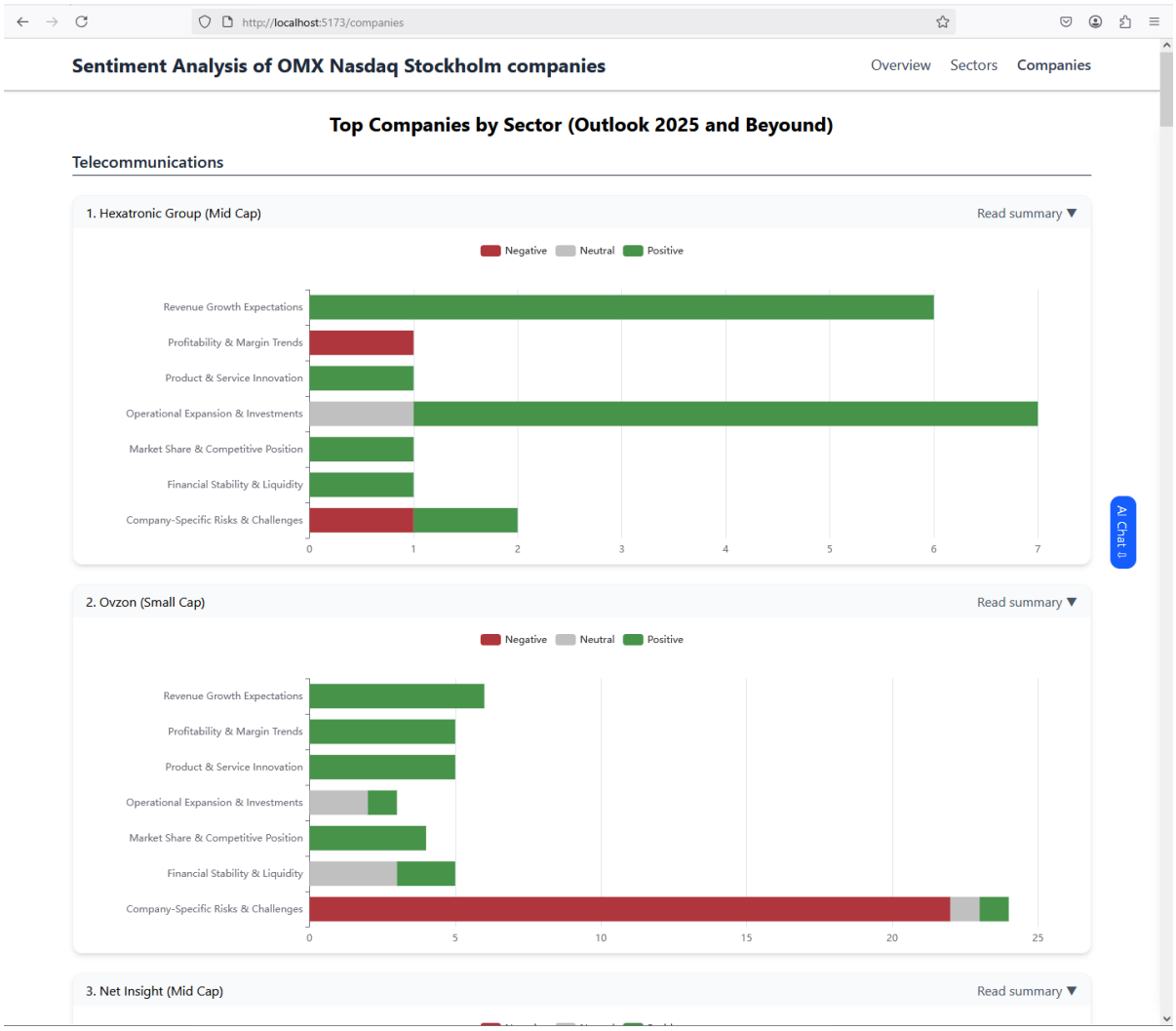


Figure 5. Companies view, top 3 companies in each sector

Each company chart visualizes the count of forward-looking statements across predefined company-level categories, using the same sentiment-based color coding as in the overview.

As in the *Sectors* view, clicking the *Read summary* link in the upper-right corner of a company chart expands the section downward to reveal the company's outlook summary and the rationale behind its ranking (see Figure 6). In all views, users can access the chatbot by clicking

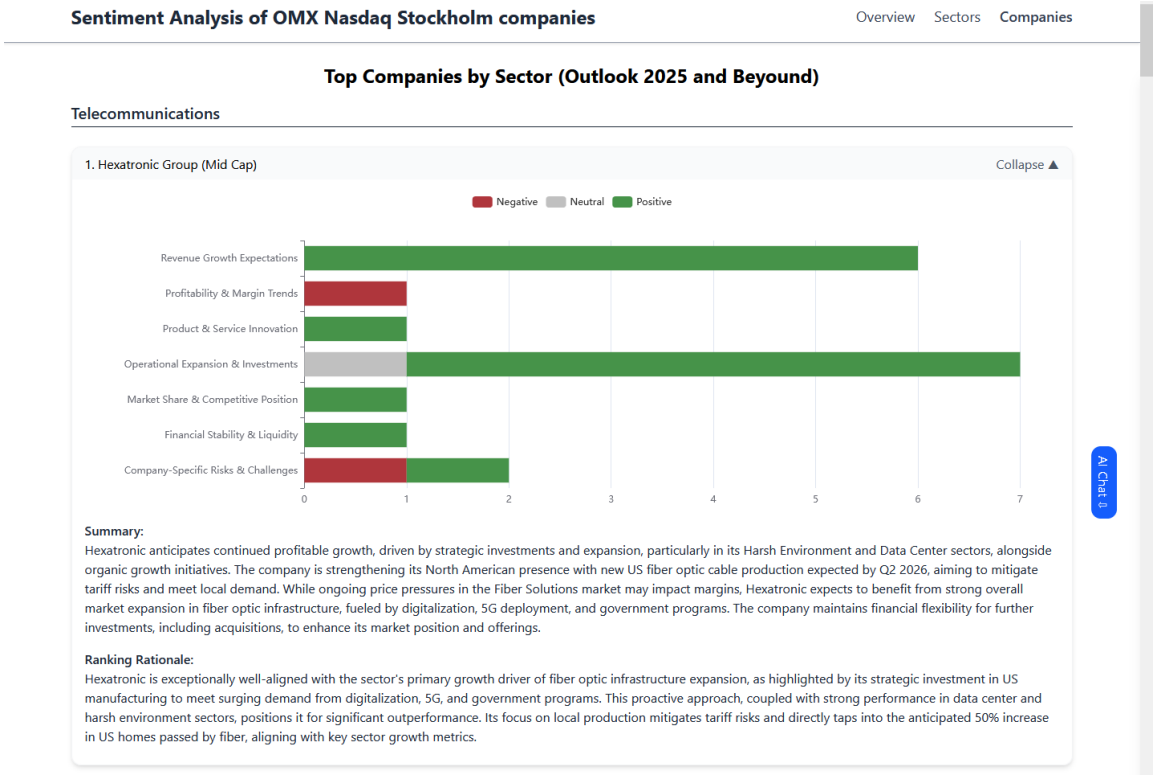


Figure 6. Company chart expanded down to show summary and ranking.

the blue AI Chat tab located on the right-hand side of the screen (see Figure 7). When activated, the chat window slides in from the right and shifts the main content area to the left.

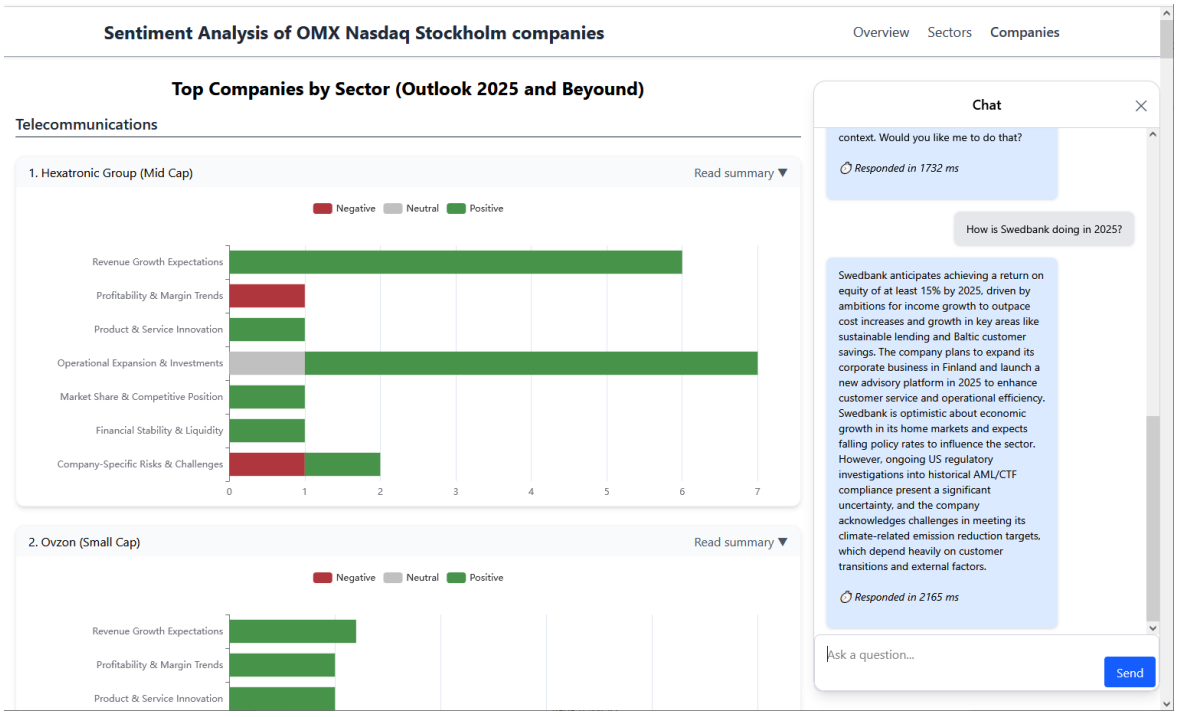


Figure 7. Chat window

Figure 7 illustrates the chat window that enables interactive exploration of company and sector insights using natural language queries. By combining structured sentiment data with retrieval-augmented generation, the system supports both visual and conversational access to financial outlook information.

This concludes the demonstration of the application's core functionalities. The following chapter presents the results of applying the system to forward-looking statements extracted from annual reports, including quantitative insights and sentiment trends across companies and sectors. The complete source code of the application, including backend processing logic and frontend components, is available in the project's public repository https://github.com/hardopost/sentiment-analysis, enabling further review, testing, or replication of the system.

5 Results

This chapter presents the results of applying the implemented system to forward-looking statements extracted from the annual reports of OMX Nasdaq Stockholm companies. The results are structured to reflect key stages of the data collection, extraction, and validation workflow. First, an overview is given of the report retrieval process and the volume of data collected, followed by statement extraction statistics, including categorization by type, sentiment, and thematic focus. The accuracy and consistency of the extracted data are then assessed through both manual inspection and a secondary language model validation process. Finally, token usage and simulated processing costs are analyzed to evaluate the system's efficiency and scalability for large-scale applications.

5.1 Extracted Statements

The full list of 359 companies on the OMX Nasdaq Stockholm main list was used as the target dataset for report retrieval. A scraper-based approach was used to collect annual and sustainability reports from centralized disclosure platforms, primarily AnnualReports.com and MFN.se. This automated method successfully retrieved reports for approximately 250 companies. The remaining ~100 reports were downloaded manually due to missing entries, inaccessible formats, or inconsistencies in company naming across sources.

Once the reports were collected, each document was processed through the system pipeline described in Chapter 4. This resulted in a total of 23,642 forward-looking statements extracted from the dataset. To enable meaningful aggregation, comparison, and sentiment analysis, each extracted statement was classified along two dimensions: its type (either company or sector) and its thematic category. The type designation was based on the scope and focus of the statement—whether it referred to a company's internal expectations or to broader industry-level developments. Within each type, statements were further categorized into one of several predefined thematic areas. These categories are outlined below. Statements labeled as **type: company** reflect the company's own future-oriented expectations, plans, risks, or strategies. These include projections about internal growth, innovation, investment decisions, financial stability, or other company-specific considerations. Even when a statement references sectoral trends, if it includes a concrete expectation specific to the company itself, it is classified as type: company.

The following categories are used for company-level statements:

1. Revenue Growth Expectations – Projections or goals related to future revenue increases.
2. Profitability & Margin Trends – Statements about expected changes in profit margins, cost structure, or operational efficiency.
3. Operational Expansion & Investments – Plans for expanding operations, entering new markets, or investing in facilities, technologies, or assets.
4. Market Share & Competitive Position – Expectations about gaining or defending market position or competitive advantages.

5. Company-Specific Product Demand & Consumer Trends – Anticipated shifts in demand for the company's own products or services, informed by consumer behavior.
6. Product & Service Innovation – Intentions to launch new offerings, improve existing products, or invest in R&D.
7. Financial Stability & Liquidity – Outlooks on cash flow, debt levels, solvency, or general financial health.
8. Company-Specific Risks & Challenges – Identified threats or uncertainties specific to the company's context, operations, or strategy.

Statements labeled as **type: sector** refer to general expectations about the broader industry, economic environment, or market dynamics. These statements are not limited to a single company but apply to all players operating in the same sector. They often concern demand trends, regulation, competition, or external risk factors.

The following categories are used for sector-level statements:

1. Expected Sector Growth – Projections about overall industry growth rates or expansion.
2. Sector-Specific Demand & Consumer Trends – Insights into consumer behavior, needs, or preferences expected to shape the sector as a whole.
3. Technological Advancements & Innovation – Anticipated innovations or disruptive technologies influencing the sector.
4. Competitive Landscape & Market Position – Broader shifts in competition, market consolidation, or global positioning.
5. Interest Rate & Financial Environment – Sector-relevant impacts of macroeconomic factors like inflation, interest rates, or currency fluctuations.
6. Industry-Specific Risks & Challenges – Regulatory, geopolitical, environmental, or structural risks that could affect the entire sector.

This classification enables the comparison of sentiment and outlook both within companies and across industries, allowing for nuanced insights into future expectations at multiple levels of analysis.

In total, 23,642 forward-looking statements were extracted from the analyzed reports. Of these, 19,529 statements (approximately 83%) were categorized as type: company, while 4,113 statements (approximately 17%) were categorized as type: sector. The following figures illustrate how these statements are distributed across the predefined thematic categories. Each horizontal bar represents the total number of statements assigned to a category, segmented by sentiment: green for positive, grey for neutral, and red for negative. Figure 8 presents the

distribution of company-level statements across the eight company-specific categories.

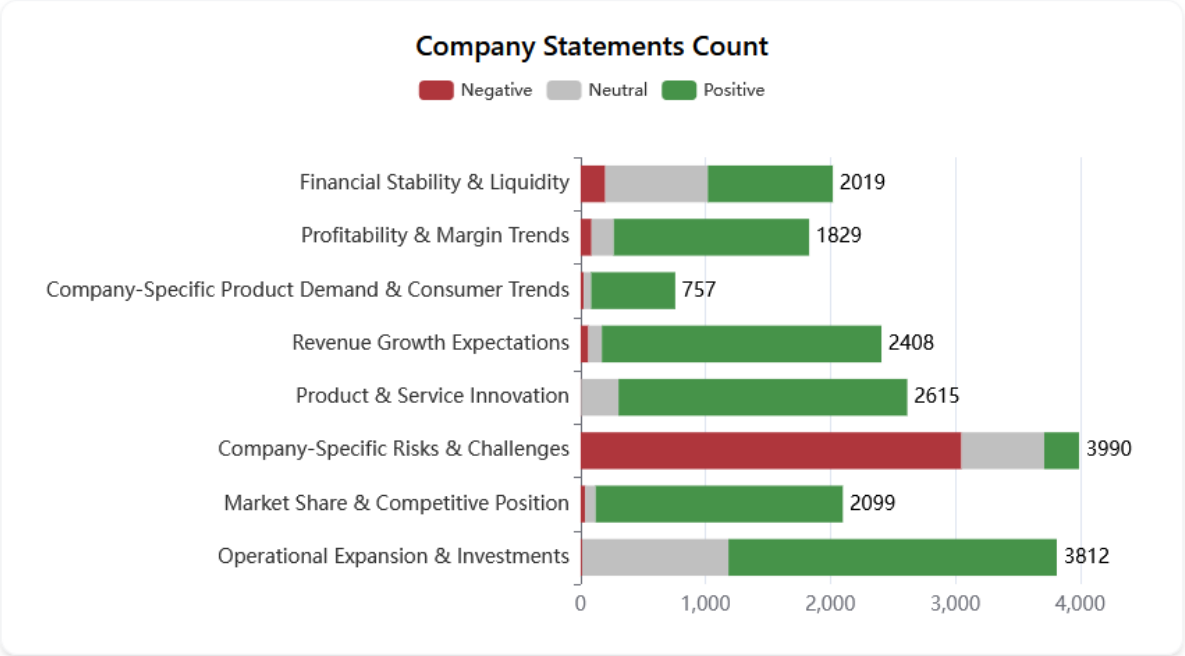


Figure 8. Type: Company statements per category

Figure 9 shows the corresponding distribution for sector-level statements. As with the company-level chart, sentiment is visualized using color segmentation, and each bar represents a distinct thematic category assigned to sector-level statements.

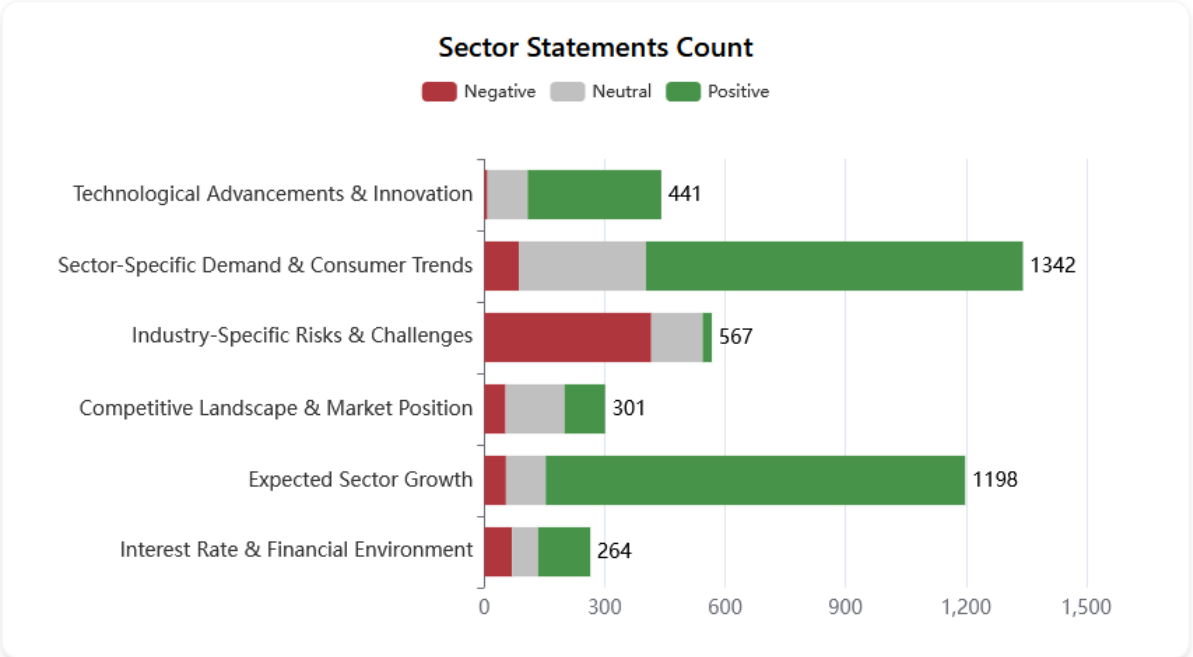


Figure 9. Type: Sector statements per category

Although sector classifications for companies were initially predefined using data from the Nasdaq website, the language model was independently instructed to assign a primary sector to each statement based on contextual information in the report. This allowed the model to determine which sector each statement most closely related to, regardless of the predefined classification. Figure 10 below presents the total number of forward-looking statements attributed to each sector, combining both type: company and type: sector statements. This illustrates how frequently different sectors were referenced in forward-looking disclosures across the dataset.

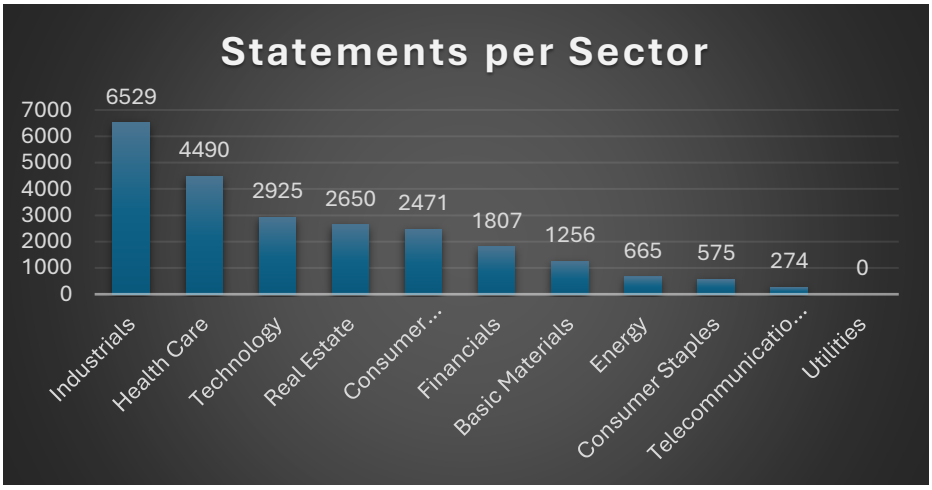


Figure 10. Statements in each sector

To provide additional context, Figure 11 presents a side-by-side comparison between the number of companies per sector based on official Nasdaq classifications and the number of companies assigned to each sector by the language model during statement extraction.

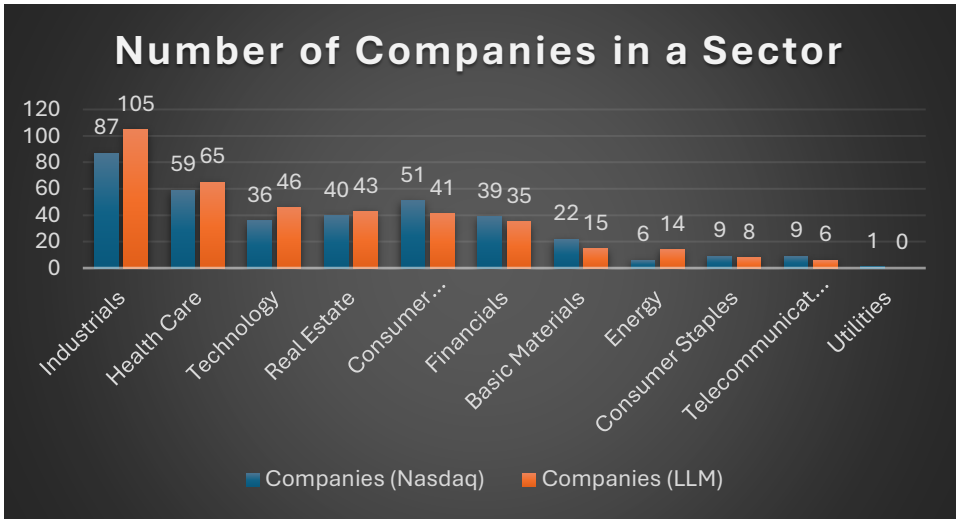


Figure 11. Companies official Nasdaq classification and LLM classification

While there it can be observed comparing Figure 10 and 11 that sectors with more companies also tend to have more statements—the language model's sector assignments are not identical to those provided by Nasdaq.

5.2 Sector and Company-Level Outputs

While the extracted forward-looking statements presented in the previous section represent the most granular layer of analysis, they serve primarily as building blocks for generating higher-level outputs. The goal of this process is to transform individual statements into concise, structured summaries that capture each company's or sector's future outlook in a way that is useful for decision-makers, analysts, and other stakeholders.

For each company, a brief summary was generated by aggregating its forward-looking statements of type company. Similarly, sector-level summaries were produced by grouping all sector-type statements across companies operating in a given sector. Each summary includes an overall tone, a list of key positive and negative drivers, and a short textual overview. A sample sector summary (Appendix A.1) and a company summary (Appendix A.2) are included in Appendix A to illustrate the format, structure, and level of detail provided in these outputs. In addition, the application supports statement-level search and comparison through an interactive chatbot interface, allowing users to retrieve insights in natural language based on the underlying extracted data. A screenshot of the chatbot in use is shown in Appendix B.1.

While categorized forward-looking statements are essential building blocks for generating meaningful sector and company-level outputs, it is equally important to ensure that these statements are accurately extracted and correctly labeled. If the underlying data is misclassified or inaccurate, the summaries and rankings derived from them risk being misleading. The following chapter addresses this by presenting a validation procedure designed to assess the quality and reliability of the extracted statements.

5.3 Manual Validation and Observations

Although the extraction prompt enforced a clear separation between company-level and sector-level categories, manual inspection revealed that finally four cross-categorization errors still occurred. Some statements labeled as type: sector were incorrectly assigned company-specific categories such as *Revenue Growth Expectations* or *Profitability & Margin Trends*, which should only be used for type: company statements.
Because the category sets were predefined and mutually exclusive, these inconsistencies could be detected using a simple SQL query that returned all statements labeled as type: sector but assigned a company-specific category. This allowed identification of violations in type-category pairing.

The final results contained 4 wrong categorizations:

- Holmen: A statement about Europe's energy transition was misclassified as *Operational Expansion & Investments*. Since it referred to sector-wide developments, it should be categorized as *Expected Sector Growth*.

- Wihlborgs Fastigheter AB: A statement on Helsingborg's port relocation was labeled as sector-level, but if Wihlborgs is involved, it should be type: company, *Operational Expansion & Investments*. If not, the correct category is *Expected Sector Growth*.
- Humana: A description of industry-wide margin pressure was labeled *Profitability & Margin Trends*, but it reflected sectoral conditions. The appropriate label is type: sector, *Industry-Specific Risks & Challenges*.

- Samhällsbyggnadsbolaget i Norden AB: A statement on rental income trends was correctly typed as sector-level, but misclassified as *Profitability & Margin Trends*. The correct category is *Interest Rate & Financial Environment*.

In addition to cross-categorization issues, the manual review also revealed that 3 statements were assigned categories not included in the original predefined list. These included new or unintended labels such as *Strategy*, *Corporate Development*, and *Product Demand & Consumer Trends* (missing the prefix "Company-Specific").

For example:

- Camurus: A statement about out-licensing CAM2032 was labeled as *Corporate Development*, a category not formally defined. Depending on interpretation, it could be reassigned to *Product & Service Innovation* (if focused on commercialization strategy) or *Company-Specific Risks & Challenges* (if highlighting dependency on external partners).

- Ericsson: A forward-looking statement about strategic execution in Cloud Software and Services was assigned the label *Strategy*. More appropriate alternatives include *Operational Expansion & Investments* (for execution-related plans) or *Profitability & Margin Trends* (if emphasizing financial discipline).

- BioArctic: A statement regarding patient access to Leqembi treatment was categorized under *Product Demand & Consumer Trends*, but without the "Company-Specific" prefix. While the intent is clear, this label should have matched the formal category name *Company-Specific Product Demand & Consumer Trends*.

These cases demonstrate that even when label sets are defined, model outputs may drift slightly unless explicitly constrained.

Some companies had forward-looking statements that were assigned to different sectors, depending on the content and context of each statement. Rather than classifying a company under a single sector, the language model assigned sectors at the statement level. As a result, individual companies may appear in multiple sectors. Figure 12 highlights companies that were represented in more than one sector based on this statement-level classification.

Figure 12. Companies assigned to multiple sectors based on extracted statement

Manual review of such cases is challenging due to the volume and variability of the data. The next chapter introduces the use of a second language model to assist in validating statement-level classifications.

5.4 LLM-Assisted Validation

To complement the manual review, a second large language model was used to validate the extracted forward-looking statements. This reverse-validation process involved supplying the model with the full report text alongside the previously extracted statements. The model was instructed to verify whether each statement appeared in the original source and to assess the accuracy of its associated metadata—type, category, sentiment, and sector.

Each statement was evaluated based on the following five criteria:

1. Whether the statement content matched the source report,
2. Correctness of type classification (company or sector),
3. Accuracy of the assigned thematic category,
4. Appropriateness of the sentiment label,
5. Correct assignment to the relevant sector.

In addition, the model was asked to indicate where in the report the original statement appeared and to provide brief justifications for any incorrect labels.

The model returned a structured response for each statement, including boolean flags for each validation criterion and short explanatory comments. All validation results were stored in a dedicated database table for further analysis.

From a random sample of 10 reports selected out of the full dataset of 359, a total of 570 forward-looking statements were evaluated. As seen on Figure 13, a second Gemini 2.5 Pro model (gemini-2.5-pro-preview-05-06) instructed by the validation prompt, confirmed that all 570 statements were present in the original report texts. However, it identified 3 cases where the type was incorrectly assigned, 9 cases of incorrect category labeling, 5 sentiment misclassifications, and 6 sector assignment errors.

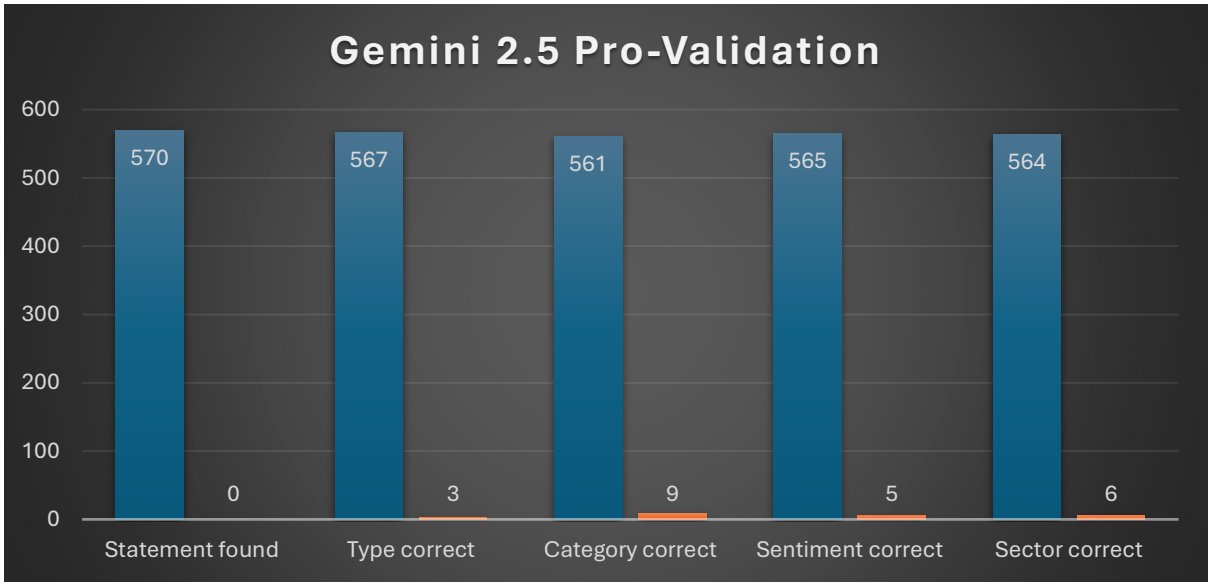


Figure 13 Gemini 2.5 Pro (gemini-2.5-pro-preview-05-06) validation results

5.5 Token Usage and Cost

Token usage was logged for each processed report, including both prompt (input) and completion (output) tokens. Cost estimates were calculated based on a tiered pricing model that differentiates between small and large prompt sizes.

The cost per million tokens depends on the size of the prompt (input):

- If the prompt is ≤ 200,000 tokens, the rate is $1.25 per million tokens for both prompt and completion.
- If the prompt exceeds 200,000 tokens, the rate increases to $2.50 per million tokens, applied to both prompt and completion tokens.

This means that even if the output is relatively small, a large prompt causes both parts of the transaction to be charged at the higher rate.

Table 1. Token Usage Summary

Prompt Size	Total Tokens	Cost per Million	Estimated Cost
Prompts ≤ 200,000	37,598,510	$1.25	$46.99
Prompts > 200,000	10,823,893	$2.50	$27.06

As shown in Table 1, most of the token usage fell within the lower-cost tier, but a significant portion of prompts exceeded the 200,000-token threshold, triggering the higher pricing rate. If the model had not been available under experimental access, the total estimated cost would have reflected standard commercial charges. These estimates provide a realistic picture of what similar large-scale processing would cost in a production environment. Pricing assumptions were based on the official rates published on the Google Cloud Vertex AI Pricing page for Gemini 2.5 Pro https://cloud.google.com/vertex-ai/generative-ai/pricing .

6 Discussion

This chapter reflects on the results presented in Chapter 5, examining the performance, reliability, and practical value of the system developed for extracting and analyzing forward-looking statements using large language models. The discussion is organized around the quality of the extracted data, the usefulness of the generated sector and company summaries, and the role of the interactive chatbot interface. It also considers observed limitations in the approach and outlines possible improvements and future directions. By interpreting both quantitative validation outcomes and qualitative system outputs, this chapter aims to assess how well the system fulfills its intended purpose and where further refinement is needed.

6.1 Interpretation of Results

The system demonstrated high accuracy in extracting forward-looking statements from annual reports. During manual database-level checks using SQL queries, only 7 misclassified statements were identified out of a total of 23,642. In a smaller sample of 570 statements from 10 randomly selected reports, the second Gemini 2.5 Pro model (gemini-2.5-pro-preview-05-06) identified only 23 issues across type, category, sentiment, or sector. These results suggest that the extraction process is reliable, though the absence of a gold-standard labeled dataset limits the ability to draw definitive accuracy conclusions.

One notable behavior was how the model assigned sectors. Although instructed to classify each statement based on the company's main sector, the model sometimes assigned different sectors to different statements from the same company. While this may reflect a failure to follow prompt instructions, it also suggests the model's ability to classify statements granularly based on their specific context—a potentially useful feature when companies operate in multiple domains.

The integrated chatbot interface successfully supports natural language querying by retrieving relevant company summaries through similarity search. While functional, its current implementation has limitations. For example, it does not use tool-based responses or function calling to execute specific tasks like comparing two companies directly or filtering summaries by time or theme. Additionally, the system uses a temperature setting of 0, which favors deterministic classification but limits conversational variety and explanatory depth.

Improving the chatbot experience could involve enabling dynamic tool use or adjusting the temperature for more descriptive responses. It may also be beneficial to incorporate domain-specific functions that allow users to directly compare sectors, rank companies, or extract specific metadata on demand.

6.2 Limitations

Regarding technical limitations, while the scraper-based was effective for initial data gathering it exhibited several limitations. In some cases, the collected PDF links pointed not to full reports but to announcement notices or summary documents. These cases required manual correction after inspection. Furthermore, because report retrieval was conducted iteratively and not all actions were logged in a fully automated tracking system, exact counts of manually corrected entries are not available. Nonetheless, a substantial portion of the dataset was assembled with partial automation. Due to these limitations, the current implementation is not yet suitable for deployment in a fully autonomous or production-grade environment. In a production environment, a more robust and consistent report retrieval solution should be considered—such

as integrating with a professional financial disclosures API or commercial data provider—to improve reliability and minimize manual intervention.

This study is based exclusively on 2024 annual reports. As time passes, the relevance of these reports will decline, since economic forecasts, regulatory developments, and market conditions evolve. The system does not currently account for changes in real-time data or apply time-sensitive filtering to summaries.

Furthermore, there is no gold-labeled dataset available to benchmark the extraction performance against human annotations. While LLM-based validation provides insight into label quality, its results cannot be fully equated with human judgment.

Finally, although Gemini 2.5 Pro was able to handle long documents due to its 1 million-token context window, alternative models (such as Anthropic Claude 3.5 Sonnet) were ruled out due to smaller input limits. This constrained comparative testing, especially for validation tasks requiring both the full report and extracted statement set as input.

6.3 Future directions

In future iterations, the system could be extended to continuously collect and process new forward-looking data from quarterly reports, press releases, earnings calls, and other corporate disclosures. In addition, continuously monitoring company-specific news could offer more timely insights into shifting expectations, especially between formal reporting periods. This would require a more reliable retrieval backend—such as a professional financial filings API, news aggregation service, or transcription feed for earnings calls—along with a fully automated ingestion and processing pipeline.

As companies release reports on a rolling basis (typically 1–3 months after fiscal year-end), the system will need to balance timeliness with token processing costs. Rather than generating summaries after each new report, summaries could be updated at regular intervals (e.g., monthly or quarterly) to avoid redundancy while maintaining relevance.

To further enhance adaptability, the system could implement a moving time window to determine which forward-looking statements are considered current. This approach would give more weight to recent data, phase out outdated content, and help maintain accurate sector and company outlooks in a constantly evolving market environment.

The underlying language model should also be periodically re-evaluated as newer, more capable models become available. The LLM landscape is evolving rapidly: in 2025 alone, models such as DeepSeek R1, Grok 3, Claude 3.7 Sonnet, GPT-4.5, and multiple versions of Google Gemini were released. Given this pace of development, research and practical applications may not immediately reflect the full capabilities or limitations of these models. Regular benchmarking against emerging alternatives could help improve accuracy, efficiency, and cost-effectiveness over time.

Conclusion

This thesis explored the use of large language models (LLMs) to extract and analyze forward-looking statements from annual financial reports of companies listed on the OMX Nasdaq Stockholm main list. The project combined natural language processing, classification, sentiment analysis, and summarization to transform unstructured disclosures into structured insights at both the company and sector level. A functioning prototype was built that supports statement extraction, aggregation, sector and company summary generation, and interactive natural language querying through a chatbot interface.

The results show that LLMs—particularly Gemini 2.5 Pro—are capable of reliably identifying forward-looking content and assigning relevant metadata with a high degree of accuracy. Both manual review and automated LLM-assisted validation confirmed strong performance in identifying relevant statements, although some misclassifications in category and sector labeling were observed. These limitations were manageable, and the outputs proved usable for constructing sector-level summaries, company outlooks, and sentiment-based rankings.

From a technical standpoint, the system demonstrated feasibility, but not full autonomy. Report collection required partial manual correction, and the system depends on specific model behavior and prompt quality. Moreover, the lack of a gold-standard labeled dataset makes formal accuracy benchmarking difficult. Still, the approach shows strong potential for scaling to include additional data sources such as quarterly reports, earnings calls, and real-time company news.

The chatbot functionality illustrates how domain-specific LLM outputs can be made accessible to non-technical users via natural language interfaces. While basic retrieval and similarity-based responses worked well, the experience could be improved through more interactive model capabilities, tool use, and dynamic function calling.

Overall, the project shows that LLMs can be leveraged not just to extract information, but to synthesize and deliver insights in a form useful for analysts, policymakers, and decision-makers. Future work should focus on automating data collection, updating summaries regularly, expanding to additional languages or markets, and evaluating newer LLMs as they emerge in a fast-moving field.

References

Alharbi, A., Alsaedi, N., & Qureshi, M. (2024). Enhancing sentiment analysis with retrieval-augmented generation for multilingual contexts. *Proceedings of the 15th International Conference on Natural Language Processing*.

Avanza Bank Holding AB. (2024). *Annual and sustainability report 2024* (p. 23).

Belcic, I. (2024). What is RAG (retrieval-augmented generation)? *IBM Think*. https://www.ibm.com/think/topics/retrieval-augmented-generation

Bergmann, D. (2024, November 7). What is a context window? *IBM*. https://www.ibm.com/think/topics/context-window

Borromeo, R., & Toyama, K. (2021). Automatic vs. crowdsourced sentiment analysis: Comparative accuracy and applicability. *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*.

Frohm, E., & Tillin, M. (2015). How can we measure firms' expectations? *Sveriges Riksbank Economic Commentaries*, (3). https://archive.riksbank.se/Documents/Rapporter/Ekonomiska_kommentarer/2015/rap_ek_kom_nr3_150422_eng.pdf

Gambacorta, L., Qureshi, S., & Chen, Z. (2024). *Large language models in central banking: Applications and implications* (BIS Working Papers No. 1215). Bank for International Settlements. https://www.bis.org/publ/work1215.htm

Gavali, R., & Shirgave, V. (2024). Refined word embeddings with intensity awareness for fine-level sentiment classification. *Journal of Computational Linguistics*.

Gemini Team. (2024). *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context* (arXiv:2403.05530v5 [cs.CL]). arXiv. https://arxiv.org/pdf/2403.05530

Google. (2024). *Embeddings overview*. Gemini API Documentation. https://ai.google.dev/gemini-api/docs/embeddings

Google Cloud. (2025). *Gemini 2.5 Pro model—Vertex AI documentation*. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro

Google Cloud. (2025). *Gemini API pricing*. https://ai.google.dev/gemini-api/docs/pricing

Google Developers Blog. (2024, March 27). Embed text with Gemini's new embedding model. https://developers.googleblog.com/en/gemini-embedding-text-model-now-available-gemini-api/

Hassanein, A., & Hussainey, K. (2015). Is forward-looking financial disclosure really informative? *The International Journal of Accounting, 50*(3), 303–326. https://doi.org/10.1016/j.intacc.2015.07.002

Hassanein, A., Hussainey, K., & Ibrahim, E. E. (2019). The impact of narrative forward-looking information in UK interim reports on investors' decisions. *Review of Quantitative Finance and Accounting, 52*(2), 631–653. https://doi.org/10.1007/s11156-018-0717-6

Kane, A. (n.d.). *pgvector: Open-source vector similarity search for PostgreSQL*. GitHub. https://github.com/pgvector/pgvector

Kavukcuoglu, K. (2025, March 25). Gemini 2.5: Our most intelligent AI model. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#enhanced-reasoning

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint* arXiv:2202.12837. https://arxiv.org/abs/2202.12837

Narayanan, S., Kumar, A., & Gupta, M. (2023). The role of sentiment analysis in economic forecasting: A comprehensive study. *International Journal of Economic Analytics, 12*(3), 45–62.

Navale, R. (2025, May). Angular vs React vs Vue: Choosing the right frontend framework for 2025. *Medium*. https://javascript.plainenglish.io/angular-vs-react-vs-vue-choosing-the-right-frontend-framework-for-2025-78bef7d260d6

OpenAI. (2025). *Model comparison: GPT-4o, GPT-4.5, GPT-o3mini*. https://platform.openai.com/docs/models/compare

OpenAI. (n.d.). *What are tokens and how to count them?* OpenAI Help Center. Retrieved April 4, 2025, from https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

Patel, M. (2025, February). Vite with Vue: Why combining them is a smart move. *Vocal*. https://vocal.media/writers/vite-with-vue-why-combining-them-is-a-smart-move

Shopen, O. (2024, November 19). Why Spring AI? *Spring Blog*. https://spring.io/blog/2024/11/19/why-spring-ai

Stack Overflow. (2024). *2024 Developer Survey: Most popular technologies*. https://survey.stackoverflow.co/2024/technology#most-popular-technologies-language

Appendices

Appendix A. Summary Output Examples

A.1 Technology Sector Summary

The technology sector is poised for significant and diverse growth from 2025 onwards, fueled by accelerating digital transformation and rapid technological advancements, particularly in Artificial Intelligence (AI), cloud computing, and 5G. Numerous sub-sectors anticipate strong expansion: the Nordic software industry expects a threefold revenue increase by 2030, "Everything as a Service" is projected to grow over 24% annually, and markets like digital customer communications, advanced authentication, and electronic shelf labels foresee double-digit CAGRs. AI-related investments are expected to claim over 15% of global IT budgets by 2025, according to Gartner.

Demand is robust, driven by businesses seeking efficiency, sustainability, and enhanced customer experiences through digitalization, automation, and AI integration. Key trends include the shift to hybrid cloud solutions, increasing adoption of biometrics for security, strong growth in e-commerce and digital marketing, and a rising need for cybersecurity. The demand for IT hardware is expected to be boosted by AI-capable computers and Windows 11 upgrades. While traditional IT services face volume decline due to cloud transformation (Tietoevry), the overall demand for tech solutions, especially those embedding AI and cloud-native capabilities, is high.

Innovation is central, with AI expected to be foundational for autonomous systems (Ericsson), streamline workflows, and create personalized experiences. Network APIs are anticipated to unlock 5G's value, and Rich Communication Services (RCS) will enhance messaging. Ultrasonic fingerprint technology and advancements in optical interconnects for data centers are also notable technological developments.

The sector faces a challenging competitive landscape characterized by price pressures (Tietoevry), the emergence of new competitors from adjacent markets or as national champions (Ericsson), and geopolitical impacts on market access. The financial environment is marked by caution due to inflation and high interest rates impacting investment appetite (Eniro, Vitec), though a gradual macroeconomic recovery is anticipated by some from late 2025 (Tietoevry).

Significant industry-specific risks persist. Cybersecurity threats, amplified by AI, are a primary concern, with the global cost of cybercrime projected to reach USD 12 trillion by 2025 (Yubico). Geopolitical instability, including trade restrictions and supply chain disruptions (Ericsson, Hexagon), poses ongoing threats. Evolving and complex regulations concerning data privacy (GDPR), AI (EU AI Act), and cybersecurity (NIS2, DORA) require significant compliance efforts. Other challenges include potential talent shortages (Knowit), the environmental impact of technology, and managing increasing data volumes.

Sentiment Rationale:

The overall sentiment for the technology sector outlook from 2025 onwards is positive. This is primarily driven by the overwhelming number of forward-looking statements expressing strong

confidence in growth across a wide array of sub-sectors, including AI, cloud services, digital marketing, cybersecurity, and various niche software markets. Companies frequently cite specific high double-digit growth forecasts and multi-billion dollar market projections (e.g., "Everything as a Service" +24.4% annually, digital customer communications market to $130B by 2029). Technological innovation, especially around AI, is consistently highlighted as a powerful engine for new opportunities and efficiencies (Knowit, Tietoevry, Ericsson).

While significant risks and challenges are extensively detailed—particularly concerning cybersecurity threats (Yubico, Proact), geopolitical instability (Ericsson), regulatory complexities, and economic headwinds (Eniro)—the sheer volume and conviction of statements forecasting market expansion and transformative technological impact outweigh these concerns. The negative outlooks on growth are often confined to specific segments like traditional telecom (Ericsson) or IT services (Tietoevry), whereas the broader technology landscape is depicted with considerable optimism for innovation-led growth. Thus, the outlook reflects more growth opportunities than overriding risks.

Ranking Rationale:

Accelerating digital transformation, particularly in AI, cloud, and 5G, drives significant and diverse growth across numerous sub-sectors. High innovation and robust demand create immense market expansion potential, despite notable cybersecurity and geopolitical risks.

A.2 Arla Plast AB Company Summary

Tone: positive

Summary: Arla Plast AB plans to enhance its operations through focused investments in its Spanish and Czech segments and expects continued positive diversification of its product and customer portfolio from the Nudec S.A.U. acquisition. The company's board has proposed a dividend, indicating financial confidence. While raw material prices are currently trending favorably downwards, their future predictability is low, presenting a potential market uncertainty.

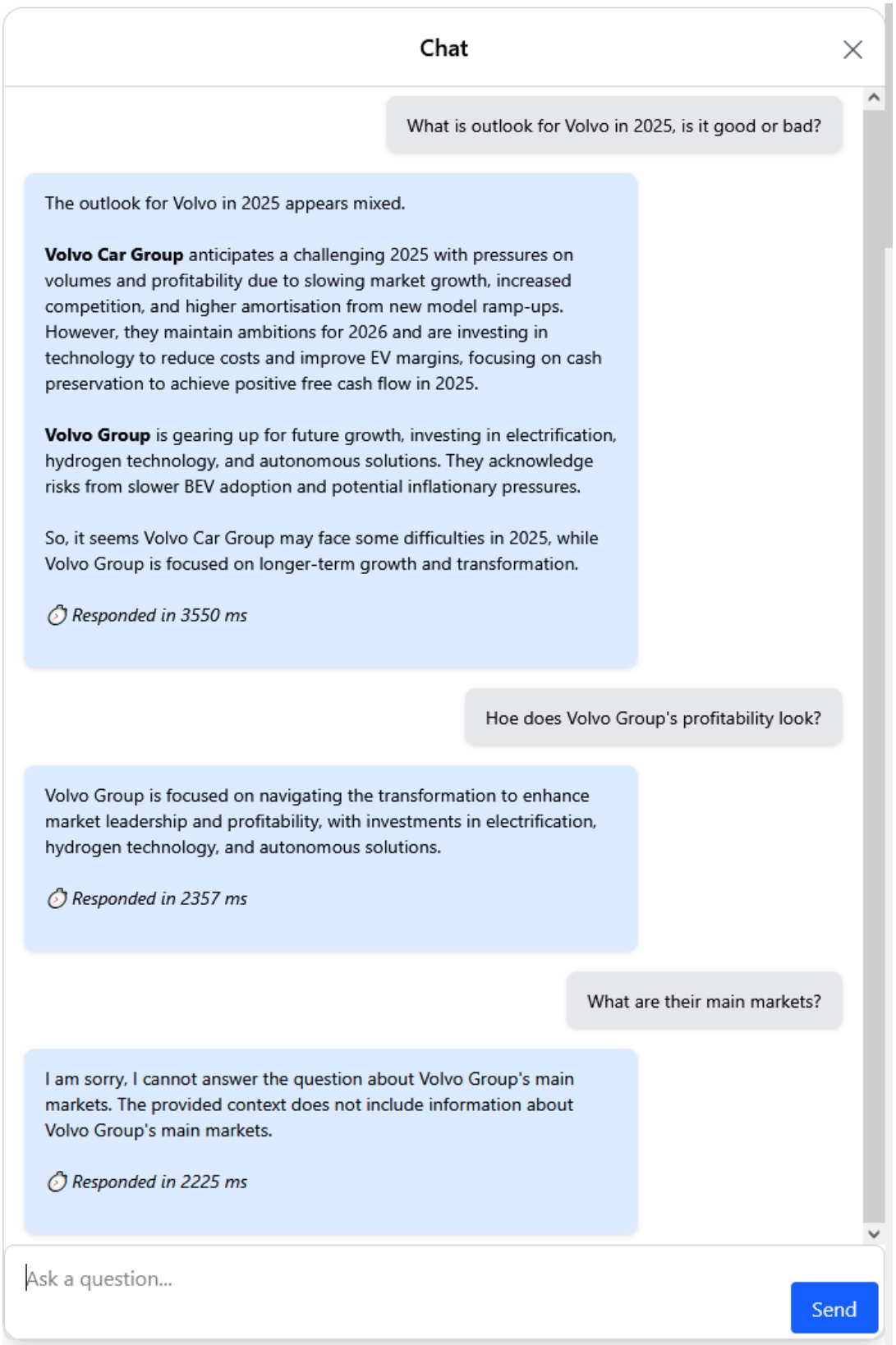
Report type: Year-End Report and Q4 Report,

Negative Drivers: Unpredictability of raw material prices.

Positive Drivers: Strategic investments and process improvements (Spain, Czech Republic), Portfolio and customer diversification (Nudec S.A.U. acquisition), Proposed dividend payout.

Appendix B. Chatbot Interface Example

Figure B.1 Chat interface showing a natural language query and response based on company-level summaries.



License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Hardo Post ,
(author´s name)

1. grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis

   Sentiment Analysis of Forward-Looking Statements from Annual Reports Using Large Language Models ,
   (title of thesis)

   supervised by Fredrik Milani ;
   (supervisor´s name)

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Hardo Post

15.05.2025