FinMAS: Financial Analysis using LLM Multi-Agent systems

Kevork Sulahian, Ivar Soares Urdalen, Suhas Parray

WorldQuant University
kevorkysulahian@gmail.com
ivar.urdalen@gmail.com
parraysuhas@gmail.com

Abstract

The focus of this practical project is to develop a Retrieval-Augmented Generation (RAG) financial analysis app that utilizes Multi-Agent systems to perform financial tasks such as extracting key insights from news articles and SEC filings, sentiment analysis, and decision-making. Our work demonstrates that an LLM-based multi-agent system is able to complete financial tasks without having the need to finetune them. The app is developed to provide transparency into the data sources and the parameters so that the result from the system can be interpreted in the context of the data that is fed to the system. The system is developed in a modular manner so that new components can be connected when there is a new model available or a different data source that needs to be connected. We show the results from case studies using the 4 multi-agent systems that are configured in the project. The system is successful at extracting key insights from the data and provide recommendations. The user needs to be aware that consistency is a challenge when using LLMs, and this project focuses on giving the system specific sets of data for analysis to reduce the variations in output between each analysis run.

Keywords: *FinTech, Financial LLMs, Multi-agent Systems, Financial Text Analysis, Retrieval-Augmented Generation*

1. Introduction

This paper presents the development of a web app that connects various financial data sources with Large Language Model (LLM) agents in a multi-agent system. The web app aims to explore the capabilities of the latest developments in LLMs for tasks in the financial domain, with a particular focus on financial text analysis, such as financial news articles and SEC filings. Reading and interpreting news articles and SEC filings are very time-consuming for financial analysts, and therefore, improving the efficiency of this task is one of the great benefits of using LLMs. We also explore other tasks, such as summarization. The app can be identified as a Retrieval-Augmented Generation (RAG) [1] app, as we are feeding the LLM agents with retrieved content from the data sources together with our queries.

In the space of AI, breakthroughs come by fast, and the cause of them is usually the increase of data availability, advancements in algorithms that simplify the architecture, and faster GPUs. The statistical learning theory suggests that AI models become better with more and better data, as well as with algorithms that efficiently learn the data. This pattern is seen in important AI milestones. from the start of using of GPUs for large-scale models in 2009 [2] to the development of Transformer architectures like Attention is All You Need in 2017 [3] and Large Language Models (LLMs) such as GPT-3 [4] and InstructGPT [5] in recent years. These advancements show the importance of data quantity and quality, computational resources, and algorithmic efficiency in driving AI capabilities.

Current solutions in the financial domain have challenges in effectively synthesizing information from both structured and unstructured sources. Ideally, teams should have

seamless access to up-to-date data such as earnings reports, regulatory filings, and social media sentiments. However, current solutions fall short of this. The current methods used to process data are slowed down due to a combination of bottlenecks in data collection, data annotation, and challenges in handling unstructured information. These inefficiencies not only result in potentially poor decision-making and open opportunities for errors that can negatively affect the accuracy of financial models but also limit the application of advanced AI techniques that could otherwise enhance financial analysis.

1.1. Financial Multi-agent System (FinMAS)

We are aiming to develop an open-source web app called FinMAS (Financial Multi-agent System) that will use LLM multi-agent system for performing financial analysis. LLMs have their main strength in the analysis of unstructured data, but we will also include some analysis of structured market data as well. A major goal of the app will be to unlock the usage of LLM models in a ready-to-use framework so that it is easy to perform financial analysis for a given ticker. Another essential goal for the app is to provide transparency in terms of the data that is fed to the system and how the system is configured. By enabling transparency in the system, it is possible to evaluate and have more confidence in the results from the system. This system intends to improve decision-making in financial institutions by providing more complex insights from various structured and unstructured data sources.

The following financial tasks we would aim to develop the multi-agent system to handle:

- Sentiment analysis and summarization of recent news articles from Benzinga News via Alpaca News API.
- Summarization of Management's Discussion & Analysis (MD&A) and Risk Factors sections from a target 10-K (annual) or 10-Q (quarterly) SEC filing.
- Question and answer from news or SEC Filing report.
- Summarization of the relevant market data for the given ticker
- Decision-making by advising for buy, sell, or hold based on a combination of the different information sources.

1.2. Use cases for FinMAS

Some examples of use cases where FinMAS would be appropriate are highlighted below:

1.1.1 Timely sentiment analysis: Financial analysis of unstructured data, such as sentiment analysis and company report analysis, is inherently time-consuming for an individual or group to perform. It also contains subjectivity for the analyst. Therefore, making LLM models easy to use by combining them in a structured framework can improve and reduce the time spent by analysts to produce analyst reports.

The daily information flow is simply too vast for a financial analyst to keep pace with news and multiple company filings. So there is a need for having an easily accessible framework that is able to perform textual analysis of different sources such as news articles and SEC filings.

1.1.2 Extract key insights of SEC filings: The app can be used to easily access SEC filings and the crucial information contained in them. The multi-agent system can provide a summary of key insights that can be useful in itself, compared to the manual process for the user for reading the filing. The app is configured to extract information from the MD&A and Risk Factors sections.

1.1.3 Educational app: This app can, together with visualization of relevant time series about a company, be a very educational experience. The LLM agents can answer different questions that the user may have about the particular ticker or the filings. Through a question-and-answer process, the user may increase their knowledge on what are the important factors for financial analysis. The transparency that the app provides also enables the user to get a better understanding of how the system works.

2. Literature Review

2.1. Introduction

This literature review will provide an overview of the state-of-the-art methods for applying LLMs to financial tasks, particularly financial text analysis. We will also provide an overview of multi-agent systems for LLMs and the recent trends in this field, with a focus on the financial domain. We will highlight the challenges of the existing solutions, as this will indicate what gaps exist in the field and where ourproject will have a contribution.

Large Language Models (LLM) have become increasingly popular ever since OpenAI released ChatGPT-3.5 in November 2022, and have also proved to be an invaluable tool in the financial domain. In recent years, we have witnessed a significant proliferation of research publications focused on the utilization of such models for financial applications. This can also be seen in the publication dates of the articles that we reference in this literature review. Recent comprehensive literature reviews also reveal that the field has experienced a rapid acceleration of academic interest and output rarely seen in any field of study [6], [7]. This underscores that there is a current and significant focus on applying these novel technologies to various financial tasks. Another aspect to highlight is that many of our references are scientific papers only published in pre-print on the arXiv.org server, and also with generally over five authors or more for many of the references. This is indicative that this is certainly a highly collaborative and fast-moving field.

2.2. Financial tasks for LLMs

LLMs are used for a wide range of tasks when applied to the financial domain. To give a brief overview of the different tasks, we provide a short description in the table below, and we use the same shorthand notation as the latest open-source benchmark dataset for LLMs called FinBen [8]. This list is not exhaustive but highlights some of the most common tasks that have been subject to application of LLMs in finance.

Table 1: Summary of relevant financial tasks applicable to LLMs

Task	Description
Textual Analysis (TA)	Sentiment analysis (SA), news headline classification, multi-class classification
Information Extraction (IE)	Named entity recognition (NER), relation extraction (RE), numeric labeling
Question Answering (QA)	Being able to answer financial queries that involve information from financial reports, etc.
Text Generation (TG)	Text summarization. For example, summarize earnings call transcripts or news articles.
Risk Management (RM)	Examples involve fraud detection, evaluating the regulatory compliance of a specific document or decision, or giving a credit score for a given scenario.
Forecasting (FO)	Forecasting based on historical prices and social media sources such as Twitter (now X)
Decision-Making (DM)	DM involves more complex and dynamic tasks, such as implementing a trading strategy or performing portfolio optimization.

Information extraction involves extracting structured information from unstructured financial text (news articles, financial reports, earnings call transcripts, etc).

Named entity recognition (NER) is an important first step for processing financial text. Entities in this context can refer to person, location, or organization. By automatically extracting entities from financial text data, it is possible to create relationships between the financial text data and the entity [9]. The NER task is an example of a task that has proved challenging to do well for general-purpose LLMs such as GPT-4 [10]. Therefore, there can be great advantages to fine-tuning an existing LLM to perform better at financial tasks.

Relation extraction refers to the process of extracting triplets from financial text that describes a relationship between two entities. An example of a triplet is Nvidia (head entity) producing (relationship) A100 GPUs (tail entity). Relation extraction can further help create a knowledge graph that connects different entities.

Textual analysis has long been an important focus in the field of AI for finance, and sentiment analysis is an essential task in this category.

News headline classification consists of classifying news headlines into classes such as Future Price News, Past Price News, Price Up, or Price Down [11]. The classes can vary depending on the classification's end goal and can be adapted depending on which dataset is used for the analysis.

Multi-class classification refers to the classification problem where a financial text can be classified according to different classes/topics that have been defined. An example of classes can be seen in the MultiFIN dataset that is used for benchmarking [12]. The text data is classified according to high-level topics such as technology, finance, etc., and low-level topics that further detail what topic is relevant to the given financial text.

2.3. Financial Sentiment Analysis

We elaborate further on sentiment analysis in this section as this will have a larger focus in this paper. Sentiment analysis is typically framed as a classification problem for text with the classes positive, neutral, and negative. By processing large amounts of financial text with sentiment analysis, it is possible to compute sentiment metrics that can gauge the market sentiment. By indicating the market sentiment, this information can be used by investors and decision-makers to make investment decisions.

One of the challenges that are identified in the literature is that several methods focus on performing SA on the whole text. This can be referred to as sequence-level, as it refers to the whole text sequence. This can be challenging for more complex texts that involve multiple entities and a discussion that can imply both a positive and negative sentiment. Therefore, there have been efforts to perform more fine-grained SA where the sentiment is classified on entity-level instead, and the datasets SEntFIN and FinEntity have been made available to the public [13], [14].

Financial sentiment analysis also presents some unique challenges compared to general sentiment analysis, as financial text contains unique vocabulary and context for interpreting sentiment.

To solve some of these challenges, finance-specific LLMs have been developed recently that have specifically focused on sentiment analysis, such as FinLlama [15] and FinBERT [16]. However, general-purpose LLMs such as ChatGPT have also proved to work pretty well on the benchmark datasets PhraseBank, FiQA, and TweetFinSent [10].

While some LLMs have been developed specifically for sentiment analysis, other LLMs for finance have been developed to work across a broad spectrum of tasks and also different modalities, such as FinTral [17] and FinMA [18].

2.4. Open source LLMs

It is important to emphasize some of the critical developments that have enabled the research to proliferate for LLMs in finance, as the general-purpose LLMs from OpenAI are closed in terms of architecture and the weights of the model. Meta released the Llama model in February 2023, where the architecture was described [19]. Later in the year, the weights were also released, so a fully capable LLM was now available for public research. Mistral 7B is another open-source LLM that was released in October 2023 [20]. The Mistral model has proven to have a unique feature in that it is able to run with lower memory requirements and quicker speed compared to Llama, for example. Several other open-source LLMs have been developed. Still, we highlight these two models since they have been crucial for opening up LLMs for public research and have achieved comparable performance on tasks compared to the closed LLMs.

2.5. Fine-tuning of LLMs

In this section, we will focus on some of the advancements that have been made that have enabled the field to move forward in applying LLMs to the financial domain with greater success.

One of the key developments to be able to take a general-purpose open-source LLM and apply it to a specific domain is a fine-tuning technique called Low-Rank Adaptation of Large Language models (LoRA) [21]. With this technique, it is possible to use existing weights for an LLM and freeze them. Then a smaller set of weights is captured in what can be called update matrices. These matrices have a lower rank than the full matrix with model weights, and therefore, fewer model weights need to be updated. The concept is illustrated in a simplified manner by Figure 1.

A complete description of the process is beyond the scope of this literature review. Still, it is important to highlight that the key idea behind LoRA is that it is possible to

focus on fine-tuning a smaller set of weights, and still achieve comparable performance compared to full-tuning of the weights. Using LoRA makes it possible to fine-tune an LLM with lower memory requirements and the fine-tuning is done faster. LoRA can be combined with other optimization methods, such as quantization, to further reduce the memory requirements and increase the fine-tuning speed [22].

While some unstructured data is common and their information is expected, some are long and complicated, and LLM, even the best closed ones fail to correctly assess and extract information from these complicated tasks. Many organizations tackle this issue by fine-tuning the LLMs with Direct Preference Optimization (DPO) or Reinforcement Learning from Human Feedback (RLHF). The problem with the stated option is in the cost of not only finetuning the model but also in the creation of the data that is used to finetune the model. Because if a task is too complicated for the model to accomplish alone, human annotators are usually involved, and this setup is both costly and time-consuming.

2.6. Multi-agent systems

Multi-agent systems have been an important part of the field of artificial intelligence and are particularly useful when tackling complex and dynamic tasks. Multi-agent systems consist of models that are set up as autonomous agents that often have a defined role with defined behavior. Together these agents can collaborate to solve challenging problems where each agent is responsible for a particular part of the problem.

LLMs have been successfully applied to be set up as a single autonomous agent where it assumes a specialized role in solving various tasks. As LLMs have been very successful as single autonomous agents, there has been an increased focus from academia and industry to use LLMs in a multi-agent system [23]. In such a system, different LLM agents with unique roles collaborate to solve a complex task, where each agent focuses on an individual subtask, either in series or in parallel. The main advantage of setting an LLM up as an agent is that it can continue to make decisions or answer their own questions to reach the desired output, in contrast to the question-answer (QA) setup that is the typical interaction between a user and an LLM.

To facilitate the development of LLM-based multi-agent systems, there is a need for a general framework that can orchestrate the communication and collaboration between the agents.

crewAI [24], Autogen [25], MetaGPT [26], and CAMEL [27] are examples of such agent orchestration frameworks. They are all very recent frameworks, and there has been quite some development to facilitate setting up multi-agent systems in a more streamlined fashion, as opposed to creating a custom solution for each study. In this project, we have focused on using the crewAI to facilitate communication between our agents.

crewAI is a framework created to coordinate AI agents to collaborate to complete tasks together efficiently. “Crew” represents a collaborative group of agents working together to achieve a set of tasks where each crew defines the strategy for task execution and agent collaboration. One of the features of crewAI is the ability to connect any agent to one of the closed or open-source LLMs that are available. This makes the framework quite flexible. Each agent is defined by the role, goal, and backstory. The tasks are defined by a description and an expected output. The agents can also be assigned various tools that will help them solve their tasks. Examples of tools can be file reading, website scraping and different search tools. The tasks are executed in what are called processes, and these processes can organize the tasks either in a sequential or hierarchical structure. If the tasks are organized in a hierarchical manner, then a manager agent that coordinates the tasks among the agents is necessary.

The core values of crewAI’s design focus on being modular, simple to use, flexible, and seamlessly fitting in with other AI technologies. Its component-based design makes it

easy to tailor and scale accordingly. Moreover, the framework is designed to integrate with tools within the llama-index [28] framework. With collaborative intelligence being the main factor here, crewAI pushes AI agents to work together on more complex problems than a single-agent framework. crewAI can be used for various tasks, right from content creation to problem-solving, representing a major change in AI agent frameworks towards efficiency and collaboration.

2.7. Multi-agent systems for financial applications

In this section, we are going to highlight a selection of publications that have utilized LLMs as agents in a multi-agent framework.

TradingGPT [29] exemplifies multi-agent systems in trading scenarios by utilizing a memory structure similar to human cognition that enables agents to organize financial data into short-term, mid-term, and long-term memory. This enhances decision-making capabilities based on varying market viewpoints of risk-seeking or risk-averse behaviors exhibited by individual agents. Through the communication between agents and effectively engaging in debates, TradingGPT boosts the process of decision-making, leading to trading outcomes and the ability to adjust to the dynamic market conditions.

Xing presented a framework that utilizes Large Language Model (LLM) agents for Financial Sentiment Analysis (FSA) [30]. Each agent in the framework has a role to play to enhance accuracy levels by targeting different errors that are commonly found in sentiment analysis. The collaboration and debate among multiple agents encourage critical thinking, resulting in more precise sentiment detection in the financial markets. We would like to highlight below the different networks for multi-agents that are discussed in the article. A homogenous network is a network where the text is used as input to multiple agents in parallel, where the agents interact and reach a consensus through debate. A network where the agents have different roles is a heterogenous network. A network structure named Heterogeneous multi-Agent Discussion (HAD) is the unique development of Xing's work. Each agent is prompted to behave differently and focus on a class of errors that LLMs can typically make when applied to sentiment analysis tasks. The output is generated by setting up a discussion between the heterogeneous agents.

Similarly, Gijsbertha introduced a study on multi-agent systems (MSA) that concentrates on sentiment analysis in cryptocurrency [31]. This study involved the use of specialized agents to analyze the sentiments expressed on social media platforms and how they impact the market trends and the value of cryptocurrency by combining quantitative market information and qualitative sentiment insights. The multi-agent workflow that was used in the study consisted of five specialist agents that were clearly defined in a prompt for the LLM, and each agent focused on a unique part of the sentiment analysis of a text (emotion, bias, context, etc). Then these agents are managed by a group chat manager, that decides who can speak and facilitates the conversation.

These studies highlight the effectiveness of multi-agent systems in enhancing sentiment analysis through collaborative agent interactions.

2.8. Competitor Analysis

In this section, we will elaborate further on the models mentioned in the previous sections. We will treat general-purpose LLMs and more finance-specific LLMs separately. For finance-specific LLMs that are applicable to sentiment analysis, we describe FinMA and FinTral models and show an example of their performance.

FinMA was fine-tuned in 3 different versions based on the Llama model (7B, 7B-full and 30B) and was developed as part of the PIXIU study [18]. A uniquely developed instruction tuning dataset was used based on different open-source datasets encompassing

sentiment analysis, news headline classification, named entity recognition, question answering, and stock market prediction.

FinTral was built using the Mistral 7B model. In comparison to the FinMA, the FinTral model was developed using more advanced methods such as pre-training on a large dataset named FinSet, instruction tuning with LoRA with quantization and DPO. As a result of the development two models were created, FinTral-INST and FinTral-DPO. The performance of the models on sentiment analysis datasets are described below.

We present results relevant to sentiment analysis from the recent FinBen benchmark study for comparison of the performance of multiple LLMs, and the finance-specific LLM FinMA [8]. The metric used for comparison is the F1 metric, which is common for classification problems. It is a combination metric that takes the harmonic mean between precision and recall. F1 metric ranges from 0 to 1, where higher is better. The best-performing model for each dataset is highlighted in bold.

Table 2: F1-score performance on sentiment analysis benchmark data [8].

Dataset	GPT4	Gemini	Mixtral 7B	LLaMA2 70B	LLaMA3 8B	FinMA 7B
FPB	0.78	0.77	0.29	0.73	0.52	**0.88**
FiQA-SA	0.8	0.81	0.16	**0.83**	0.7	0.79

FPB = Financial Phrase Bank [32], FiQA-SA = Financial Opinion Mining and Question Answering - Sentiment Analysis (news and tweets) [33]

The comparative analysis in the FinBen study shows that a fine-tuned Llama model (FinMA) and the largest Llama model (70B) are able to perform well on the benchmark datasets FPB and FiQA-SA. It also shows that the closed general-purpose LLMs GPT4 and Gemini have good performance as well. The Mixtral 7B model has poor performance, and together with the performance of the Llama 3 8B might be indicative that either the model has to have sufficient parameters or has to be fine-tuned to perform well on these sentiment analysis datasets.

In the development of the FinTral model, a comparison between different LLM models was performed as well, and the results relevant to sentiment analysis are highlighted below. The benchmark datasets that were used were FPB, FiQA-SA as mentioned in FinBen study. In addition, a FOMC dataset [34] and a news headline dataset [11] were used. The accuracy score presented in Table 3 is the average score across the benchmark datasets.

Table 3: Avg. accuracy score across 4 benchmark datasets [17]

Model	Type	Avg. accuracy
FinTral-INST	Instruction fine-tuned	0.82
FinTral-DPO	RL-tuned	0.81
Mistral-7B-Instruct-v0.1	Instruction fine-tuned	0.49
Mistral-7B-v0.1	Pre-trained	0.25
GPT-4	RL-tuned	0.79
ChatGPT 3.5-turbo	RL-tuned	0.7
FinMA-7B-full	Fine-tuned	0.78
FinMA-7B	Fine-tuned	0.72
Llama-2-13b-chat-hf	RL-tuned	0.58
Llama-2-7b-chat-hf	RL-tuned	0.54
Llama-2-7b-hf	Pre-trained	0.26

The comparative analysis results shown in Table 3 show that a fine-tuned model based on Mistral can perform very well on the SA benchmark datasets (FinTral), even though the default Mistral model performs poorly. The general-purpose and closed LLM GPT-4 performs comparable to that of the FinTral model. The FinMA model that is mentioned earlier also performs decently, whereas the fully fine-tuned model performs the best. The FinMA model is based on the Llama model, so it shows that it is possible to fine-tune the Llama model to achieve good performance for sentiment analysis as was done with the FinTral model.

Based on the results presented in the literature and the previous sections, we highlight some of the features of general-purpose and finance-specific LLMs in this SWOT analysis. We include both closed (GPT-4, Gemini) and open-source (Llama, Mistral) here as well. We only consider single LLM systems in this SWOT analysis as multi-agent systems are still at an early stage.

Table 4: SWOT analysis of general-purpose and finance-specific LLMs

Strengths	**Weaknesses**
Closed LLMs can be used directly without fine-tuning to some extent. This applies to the larger models. Smaller LLMs can be fine-tuned and achieve good performance. That means they can run with lower hardware requirements Able to process large volumes of data in a quick fashion. Compared to pre-LLM methods, LLMs can understand context and also different languages.	The open-source smaller LLMs (7B) perform poorly on sentiment benchmarks Smaller LLMs need to be fine-tuned to perform well, which in some instances, can be an expensive process even with several optimization methods. LLMs overall: - can be considered black-box and have little explainability. - can be quite resource-intensive to run - does have randomness inherent in the model structure, the answers can be unreliable to some extent
Opportunities	**Threats**
With more advanced and timely sentiment analysis, it is possible to integrate the results from LLMs with trading strategy, risk management, or portfolio management. By being able to process large volumes accurately, LLMs can be able to uncover new insights or patterns in market sentiment.	Regulations can create restrictions on AI for finance, especially models that have low explainability. The field moves very fast so that models that are put into production can become obsolete in a short time period. As the performance of LLM models is quite impressive for sentiment analysis, it is possible that the models can be left with little human oversight. This might cause challenges if the model starts behaving badly.

2.9. Current challenges for existing solutions

With respect to the usage of single LLMs without any pre-defined logic, system, or environment, it often is the case that the LLM is either too much for a simple task or too simple for a complex task. As an easy but expensive solution, many companies are going forward with finetuning the LLM for the more complicated tasks, and as mentioned previously, finetuning LLMs is not only expensive in terms of computing but also expensive in terms of the cost of human resources that is spent creating and evaluating the datasets.

In the paper that introduces an annotation method called Multi-News+ [35] the authors emphasize the expense of using human annotators. In their case, it is used to remove the noise from a dataset, and the efficiency on relying only on human annotators, especially for large-scale projects. Another article that introduces a fine-tuning technique called LaFFi [36] suggests that a hybrid approach is more cost-effective. Both papers agree that human involvement is resource-intensive and thus too expensive. Our approach aims to minimize the cost significantly by focusing on an agentic approach that can replicate high results without the need to finetune any models.

2.10. Literature Review Summary

In this literature review, we have surveyed the state-of-the-art developments regarding LLMs for financial tasks, and where we have focused particularly on financial text analysis and sentiment analysis. Even though the main focus in the literature has been single model systems, we also highlighted a selection of studies that have applied a multi-agent system for sentiment analysis applications.

It is remarkable to notice the advancement of the field during the last years, the level of collaboration and the state of the most advanced models. Open-source LLMs, together with fine-tuning techniques such as LoRA and quantization, have enabled researchers across the globe to take advantage of these new technologies and apply them to the financial domain.

There are challenges related to the computational costs of training and developing such models, and thus by either fine-tuning smaller models or creating multi-agent systems, the computational costs can potentially be reduced.

3. Methodology

As the outcome of this study is an open-source web app, we describe our approach to developing the app and also highlight obstacles that were encountered. We further describe how these obstacles were dealt with.

3.1. Overview

The following steps were performed to develop the FinMAS web app:

1. As a first step, we will establish a framework for the web app with the necessary components where we can connect user input with model responses. The framework should support choosing a ticker, different specifications for the model, and display data such as charts and text from agent responses.

2. Develop/integrate different tools for use by the agents. For the agents to be able to use updated data, we developed tools that are able to fetch recent news articles (Benzinga News via Alpaca News API), SEC filings or market data (Alpha Vantage API). For textual data, we used the llama-index query engine for developing the agent tool, and for market data, we developed a custom tool to feed the LLM agents with financial figures in a Markdown table format.

3. Evaluated different LLM models and agent architectures to reach our desired outcome. Especially investigate setups that are able to use open-source LLM models (Llama), and compare that to setups using closed LLM models (OpenAI). The web app will be flexible in terms of where the LLM models are hosted and that the type of model can be user configurable.

4. We performed a qualitative evaluation of the output from the multi-agent systems (crews). We compared and evaluated parameters like speed, accuracy, and complexity of the response.

5. To ensure proper usage and description of the web app, documentation was developed to describe the different features of the app.

3.2. Obstacles to overcome

When developing apps that utilize LLM models for multi-agent systems, there are several obstacles to overcome while developing the solution. LLMs are inherently

compute-intensive models that require powerful hardware to run. Smaller models (around 8B in parameter size) are able to run locally on a 16 GB RAM computer, but normally, a cloud-hosted LLM would be necessary to run an LLM model of sufficient size and speed for financial tasks. We evaluated different options available for an efficient setup but also make the system configurable by the user so that the user can use paid API (OpenAI) or a free API (Groq hosted models).

There has been an explosive proliferation of open-source models and supporting tools in recent times. To find the right tools and models that can perform well, it was necessary to perform various experiments to evaluate which components have good performance on financial tasks and which components are not applicable in this context.

3.3. Solutions applied in FinMAS

3.3.1 Cloud-hosted small and efficient LLMs: In recent months, we have seen the new releases of small and efficient LLMs such as Llama 3.2, Mistral Nemo, and Gemma 2-2b. There are multiple tasks where there is no need for a larger general-purpose model where smaller models can perform well. With FinMAS it is possible to investigate the performance of smaller models in a multi-agent system, and such models can provide great benefits in terms of compute and energy cost. We utilize Groq [37] hosted models for open-source LLMs such as Llama and OpenAI for gpt-4o and gpt-4o-mini models. Groq provide a LPU (Language Processing Unit) Inference Engine that has deliveres an fast speed for their hosted models. In this manner they are able to provide a higher free API rate limit compared to competing solutions. This has been instrumental in developing the app as we are able to test and debug the system within the free limits.

3.3.2 Configurable Multi-agent system with crewAI: Creating a system design that modular and configurable is essential. With the FinMAS app we are able to connect different data sources with different crews, and this can be controlled in the UI of the app.

Using an LLM model as an agent means that we are giving the model a defined role, goal, and backstory. In addition, we can also give the agents specific tools that they can use to solve their tasks. The tools can be fetching data from a news source or fetching fundamental data. An important feature of an LLM agent is the concept of memory as well. The agent would be able to use knowledge from past actions in their task handling. This contrasts the use of LLM models via a chat interface, where commonly the user has to give a lot of context before getting the desired answer from the model. A dedicated agent setup that is pre-configured makes the model an efficient tool for the user to perform an analysis.

In FinMAS, the user is able to evaluate different roles and structures of the multi-agent system to find a configuration with a good performance. Examples of agent roles are Fundamental Analyst, Researcher (uses sentiment analysis), a summarizer role, and a quality assurance role. It is important to give an agent a dedicated focus area for their specific tasks and a few specific tools necessary for them to give answers with updated data.

3.4. Web app architecture

The web app architecture consists of a UI framework where the user interacts with the system, data sources, LLM providers, and crews. The connections between these components are illustrated in Figure 2. The user can inspect the data sources directly from the app itself to provide transparency. The crews use the data sources through the use of tools and the LLMs that power the crew of LLM agents are hosted on one of the current LLM providers. We have implemented a configurable setup so that it is easy to switch

between different LLM providers, and it is possible to run a performance comparison between different LLMs by running the crew analysis with different setups.

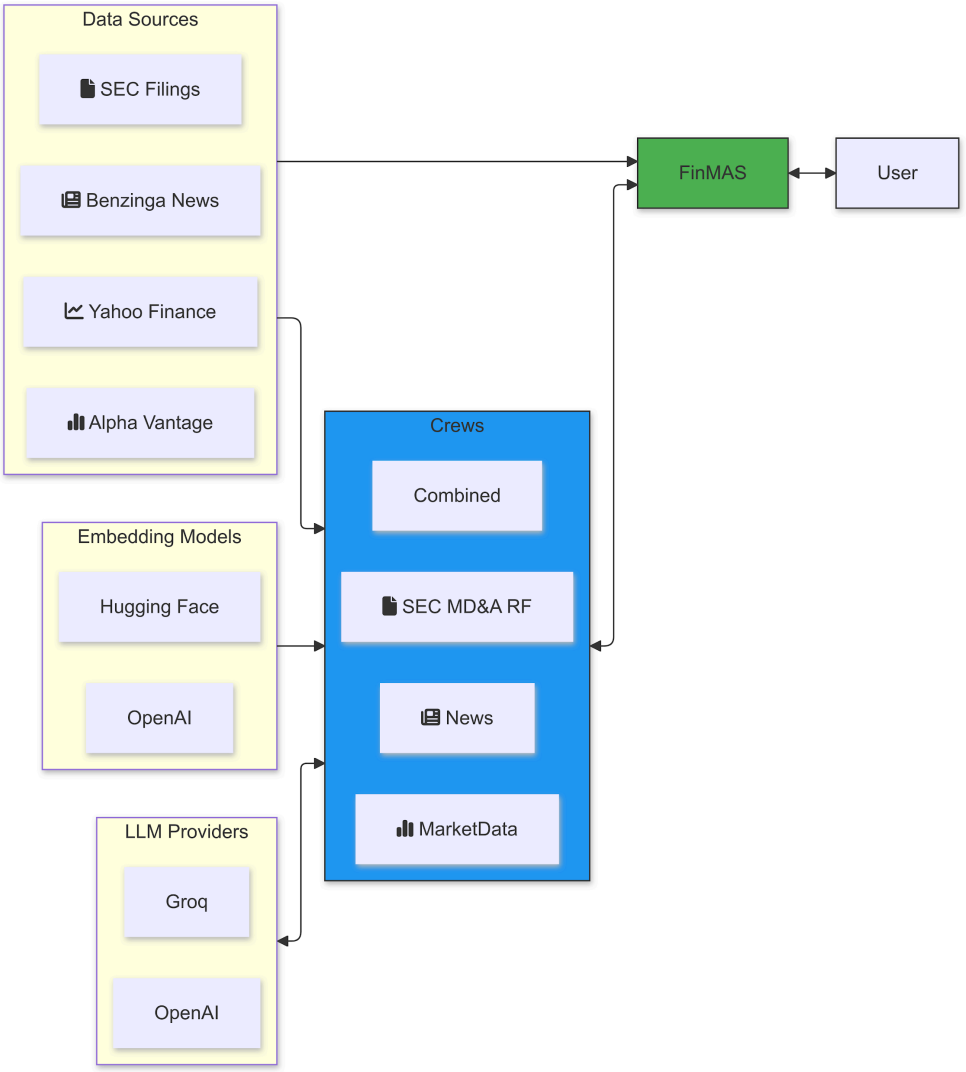

Figure 1: Architecture of FinMAS app

3.5. Data sources

We are using the following data sources that are also highlighted in Figure 1.
Benzinga News articles are retrieved via Alpaca Historical News API. This API has a free tier for users with an Alpaca account. The news articles consist of a headline, publish date, HTML content, author information, and a list of tickers that are relevant to the news source. By using the selected ticker, we are able to retrieve historical news articles for a given range of dates. Then, the article is formatted and processed to go through sentiment analysis and further processing that is pre-defined in the agentic flow. The processing consists of selecting news articles that are focused on the specified ticker and not a large collection of tickers. The HTML content is stripped from images and tables to enable the agent to efficiently retrieve the content that is relevant for the task.

We retrieve SEC filings using the edgartools python package that facilitates the retrieval of the latest filings. We use SEC filings like 10-K (annual) and 10-Q (quarterly). These reports contain critical data on the risks, financial health, and overall performance of a company that can be missed in other data sources. The HTML content of the filings is processed where the tables and images are stripped from the report to make it easier for an

embedding model to retrieve the relevant chunks of text to feed the LLM agent together with their query.

For fundamental data, Alpha Vantage API is used to retrieve historical time series of the income statement and balance sheet. A challenge for using this data source is that the free tier is very restrictive in terms of the number of API calls that can be done per day. Caching of the results is therefore utilized to prevent excessive API calls for the same data. Ratios like Price-to-earnings (P/E), basic Earnings-per-share (EPS), Price-to-sales (P/S), and Debt-to-equity (D/E) are calculated from the raw data in the income statement and balance sheet.

For stock price data, Yahoo Finance is used to retrieve the latest year's price data. The price data is resampled on a weekly time frame and a few technical indicators are shown in the UI such as 20-week and 50-week moving averages together with RSI and Bollinger Band Percentage.

3.6. Tools

An essential feature of an LLM agent is the possibility to use tools to extract data from different sources, which makes it possible for the LLM agent to use the data when generating their answer. We have set up tools for News, SEC filings, fundamental data, and technical analysis.

3.6.1. Embedding models: The tools for textual analysis rely on embedding models that heavily impact the performance of the tool. When a Groq-hosted LLM is used, then an embedding model is fetched from HuggingFace and run locally. The default HuggingFace embedding model is the bge-small-en-v1.5 from the Beijing Academy of Artificial Intelligence (BAAI), which is a small and efficient general-purpose embedding model for English language text. It generates embeddings in 384 dimensions. It is not tuned particularly on financial vocabulary, so it is expected that some inaccuracies may occur compared to using an embedding model that is especially focused on financial text.

When using an OpenAI model like gpt-4o-mini, then the default embedding model is the OpenAI hosted model text-embedding-3-small. Even though its usage is paid, it is a very cost-effective model and generates embeddings in 512 dimensions.

3.6.2. News analysis tool: The news analysis tool relies on the LlamaIndex project [28] to parse the news articles into a vector database. The LlamaIndex is a data framework built to make it easy to connect data sources into LLM models. The news articles are first cleaned to make them ready for being processed by an embedding model. The goal of the text cleaning is to extract clean text content from the HTML content of the news article.

Further, an embedding model is used on the cleaned news articles. The transformation of text into embedding vectors by an embedding model is an important step in the tool. Embedding vectors represent the text with numbers in a vector, and this helps the LLM interpret meaning and similarity between different words. The embedding model type can be configured and can also impact the result of the system. LlamaIndex provides helper functions to create a vector database from the embedding vectors. We then feed this vector database into the tool that the News Analysis Crew can use to generate their result.

Figure 2: Simplified flow for news tool

3.6.3. SEC analysis tool: This tool extracts text content from the selected filing (e.g., 10-K, 10-Q) in HTML format. The HTML content is parsed to extract the table of contents so that specific sections can be identified. The focus of the tool is either the

Management's Discussion & Analysis section or the Risk Factors section. The selected section is extracted via the parsing methodology. The parser strips the HTML content for tables, images, and other irrelevant HTML tags. As SEC filings are quite large documents, it is important that for efficient and correct processing by the embedding models that we target a reasonably sized portion of the content to be processed. In this manner, we are able to reduce the number of tokens spent when processing the content and also stay within the context window of the LLM.

Then, the clean text content is processed in a similar fashion as the news analysis tool to create a vector store index of the embeddings. Then relevant content can be easily retrieved by the LLM agent during the crew execution.

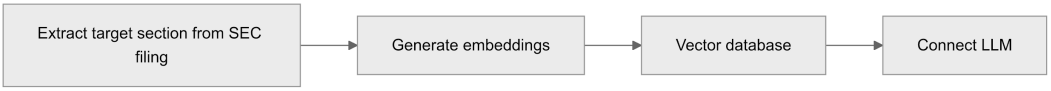

Figure 3: Simplified flow for SEC filings tool

3.6.3. Fundamentals tool: The fundamentals tool fetches the income statement and balance sheet. The tool computes ratios and growth rates for the last 8 quarters and formats the data in a Markdown table. A cleanly formatted Markdown table with an intro text preceding the table is one method for an LLM to process tabular data. It is beneficial if the table is not too extensive for the LLM to focus on the relevant data. Year-over-year Growth rates on trailing twelve months (TTM) are included so that the LLM can evaluate the performance of the company for the recent quarters.

3.6.4. Technical Analysis tool: The technical analysis tool resamples the daily price data into a weekly timeframe and computes a selection of indicators that represent trend (moving averages), momentum (RSI), and volatility (Bollinger Band Percentage). The data is formatted in a Markdown table, together with an intro text that gives context to the table.

3.7. Crews

The crews we have defined are a News crew, SEC Filing crew, Market Data analysis crew, and a Combined analysis crew. The crews are defined in terms of agents, tasks, and tools. In practice, these are configured in a YAML file. The full configuration is displayed for the user in the UI before running the crew analysis. The user is able to adjust the configuration of the crew by changing the goals of the agents or changing the task description, for example.

The News crew consists of a news analyzer agent, a sentiment analyst, and a news summarizer agent. The news analyzer agent has the goal of fetching the news from the attached tool and generating an analysis of key insights from the data provided. Then, the sentiment analyzer agent will generate a sentiment analysis based on the provided content. Lastly, the summarizer (also called report writer) summarizes the key information from both agents and produces a condensed summary report.

The SEC Filing crew focused on two sections: Management's Discussion & Analysis and the Risk Factors. A single agent is responsible for analyzing each section and subsequently, the results are summarized and combined by a final agent.

The Market Data crew consists of a Fundamentals analyst, a Technical analyst, and a Stock Advisor. The analysts use their respective tools to provide insights into their responsibilities. Their analysis is sent to the Stock Advisor who summarizes their findings and provides a final recommendation of buy, hold, or sell for the specific stock.

The Combined Analysis crew structure is illustrated in Figure 4, and is effectively similar to the Market Data crew, but with a News Analyst and a SEC Filing Analyst. The

objective of the crew is to combine market data with financial textual data and in the end produce a summarized report that can be used to evaluate the financial health and performance of the stock.

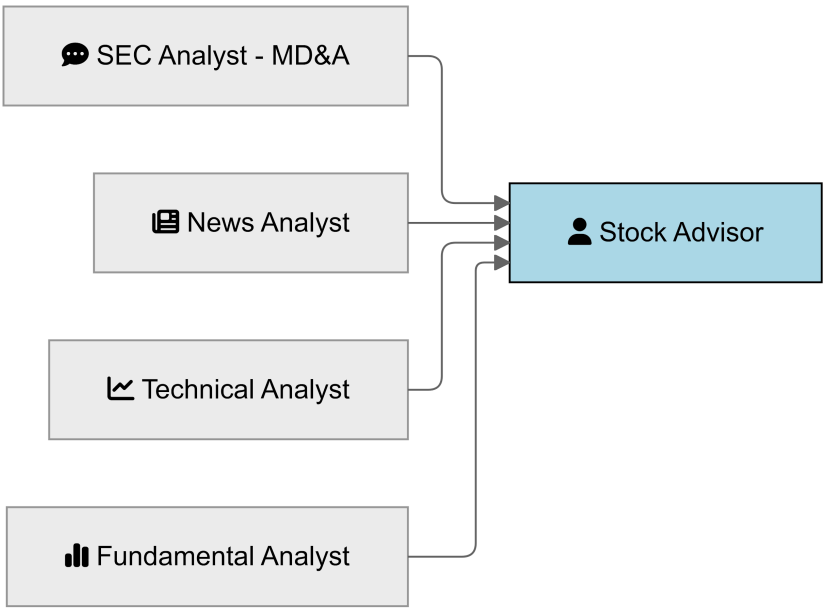

Figure 4: Multi-agent structure for the Combined analysis crew

3.8. Large Language Models Configuration

The results were produced by using Groq and OpenAI as the LLM providers. For Groq hosted model, the LLM llama3-8b-8192 was used. This is a relatively small and efficient open-source model, which can be used together with Groq free tier. The temperature was set to 0.0, and max tokens per request to 1024. The temperature for the LLM controls how much "creativity" and randomness that is permitted in the response. When the temperature is set to 0, then it is expected that the crew will produce results that are similar between runs. There will still be some randomness in the result, even with the temperature set to 0.0. While Groq does provide hosted versions of the newer Llama models (3.1 and 3.2), they have proven to not be particularly fast because of restrictions on tokens per minute. Thus, the newer Llama models were not favorable to use as the LLM for the agents.

When using OpenAI as the LLM provider, the models gpt-4o and gpt-4o-mini were used, which were released in May and July 2024, respectively [38]. gpt-4o-mini is a smaller and cost-efficient LLM that has a larger context window compared to the Llama 3-8b model. The pricing of gpt-4o-mini makes it a much more affordable alternative compared to the gpt-4o model. Table 4 below shows a summary of some key information of the models. OpenAI has not officially disclosed any information about the parameter size of the gpt-4o models.

Table 4: Summary of Large Language Models used

Model Id	Context Window	Parameter Size	Released	Input Cost ($/MT)	Output Cost ($/MT)
gpt-4o-mini	128k	Not disclosed	2024-07-18	0.15	0.6
gpt-4o	128k	Not disclosed	2024-05-13	2.5	10
llama3-8b-8192	8192	8b	2024-04-18	Free	Free

4. Results

In this section, we present the results from the different crews that are configured in the app. The output consists of a summarized report in Markdown format and the output is shown in the UI as well as exported as a file for review. The app is focused on the analysis of a ticker, and therefore, the data that is used as input to the crews is ticker-focused. As a case study to evaluate and demonstrate the capabilities of the crews, a selection of US tickers was used. To ensure the reproducibility of the results, input data are fixed to default values, and also the news and fundamental data are available as fixed datasets from the code repository. The example outputs referenced here are available for full review in the code repository and in the examples section of the documentation.

4.1. News analysis crew

The crew was set up to process news articles between 2024-10-15 to 2024-11-10 from Benzinga News, and extract key insights, and analyze the overall sentiment from the small dataset of news articles. We present some excerpts from the crews here. The full output is available at the GitHub repo for the project and is linked to in the appendix. The crew spends about 1 - 2 minutes per run.

We illustrate here how different crew analysis runs perform with the same data but with different LLM and embedding models.

Table 5: Comparison of major events section for News Analysis crew

Major Events (llama3-8b-8192)

TSLA's Q4 Earnings Announcement: Tesla reported a record net income of $5.5 billion, largely due to the sale of its Bitcoin holdings, showcasing the company's financial performance and ability to generate significant profits.
TSLA's Self-Driving Technology Update: Tesla introduced a new chip that promises improved performance and reduced errors, highlighting the company's advancements in autonomous driving technology.
TSLA's Customer Satisfaction Survey: A recent survey found that Tesla's customer satisfaction ratings have improved, but still trail behind those of other automakers, indicating mixed results.
TSLA's Environmental Impact Report: Tesla published an environmental impact report, detailing its efforts to reduce waste and promote sustainability in its manufacturing processes, highlighting the company's commitment to environmental responsibility.

Major Events: (gpt-4o-mini)

Stock Performance Surge:
Following the election of Donald Trump as the 47th U.S. president, TSLA's stock surged nearly 29% in the week post-election, closing at $321.22, an 8.19% gain on the last trading day reported. This rally signals strong investor confidence in potential policy changes favoring the electric vehicle market.

Sales Performance in China:
In October, Tesla sold a total of 68,280 vehicles in China, with 40,485 units sold domestically. This represents a decline in domestic sales, the lowest since April, but still a 41.43% increase year-over-year. Year-to-date, Tesla has surpassed 500,000 vehicle sales in China, reflecting an 8.29% increase from the previous year, underscoring the importance of this market for Tesla's overall performance.

Financial Results:
Tesla's third-quarter revenue reached $25.18 billion, marking an 8% year-over-year increase. However, this figure fell short of analyst expectations, raising concerns about the company's growth trajectory. Despite this, Tesla's market capitalization has exceeded $1 trillion, driven by optimism regarding its future growth, particularly in the autonomous vehicle sector.

4.2. SEC MD&A and Risk Factor analysis crew

The SEC analysis crew processes a selected 10-K or 10-Q filings to extract key insights from the Management's Discussion & Analysis and Risk Factors section. Below are some excerpts from the crew's output, which shows the key insights extracted from SEC filing. The full version of the output can be found in the project's GitHub repository. The runtime for SEC crew takes about 1 minute per run and depends on the size of the filing. We highlight a case study for NVDA 10-K (annual) report with the filing date 2024-02-21, and the comparison is between the LLM models gpt-4o and llama3-8b-8192. Keep in mind that the default embedding models are different, as explained in the methodology section.

Table 6: Excerpts from MD&A section for NVDA 10-K analysis

Key Insights from Management's Discussion and Analysis: (llama3-8b-8192)

- NVDA's financial performance is driven by the demand for its products and services, particularly in the areas of visual computing, professional visualization, and gaming.
- The company expects to continue growing its business by delivering innovative products and services that meet the evolving needs of its customers.
- NVDA is focused on addressing the growing demands for AI, data analytics, and cloud computing, and believes that its GPU technology is well-positioned to enable these trends.
- The company is investing in the development of new technologies and products to address emerging markets and trends, such as augmented and virtual reality.

Key Insights from Management's Discussion and Analysis: (gpt-4o)
Market Segments Explored: NVIDIA operates in two main segments:
Compute & Networking: This segment focuses on data center systems and products, AI platforms, and networking solutions. The company has experienced a significant increase in demand for data center products, driven by the growing need for AI and accelerated computing solutions.
Graphics: This segment includes gaming, professional visualization, and automotive solutions. NVIDIA has reported revenue growth in these areas, supported by the launch of new GPUs and partnerships in the automotive industry.

Management's Forward-Looking Statements: NVIDIA anticipates continued strong demand for its data center products and is actively managing supply chain complexities to meet this demand. The company is also focused on navigating product transitions, which involve challenges in balancing supply and demand and managing inventory provisions. NVIDIA's strategic initiatives include expanding its AI inference platforms and enhancing its automotive partnerships.

Table 7: Excerpts from Risk Factors section for NVDA 10-K analysis

Risk Factors: (llama3-8b-8192)

- Competition from other companies in the semiconductor industry
- Dependence on a limited number of customers and suppliers
- Fluctuations in global demand for semiconductors and other products
- The company's ability to develop and manufacture new and innovative products
- The company's ability to protect its intellectual property
- The company's ability to manage its supply chain and manufacturing operations
- The company's ability to comply with laws and regulations related to the environment, health, and safety
- The company's ability to manage its cash flow and liquidity
- The company's ability to maintain its relationships with its customers and suppliers
- The company's ability to adapt to changes in the market and industry trends

Key Insights from Risk Factors: (gpt-4o)
Industry and Market Dynamics: NVIDIA faces risks related to evolving industry needs and competition. The company must continuously innovate to meet market demands and maintain its competitive edge.
Demand, Supply, and Manufacturing Risks: Challenges include accurately estimating customer demand, dependency on third-party suppliers, and potential product defects. These factors could impact NVIDIA's ability to deliver products efficiently.
Global Operational Risks: The company is exposed to adverse economic conditions, international sales and operations challenges, cybersecurity threats, and potential business disruptions.
Regulatory, Legal, and Stock-Related Risks: NVIDIA must navigate complex laws and regulations, address scrutiny over corporate sustainability practices, and manage intellectual property and data privacy issues. Additionally, the company faces risks related to indebtedness and anti-takeover provisions that could affect corporate control.

4.3. Market Data Analysis crew

For the Market Data analysis crew we showcase a study for META using the last 8 quarters for the fundamental data and the last 8 weeks for the technical analysis data. The notebook for Market Data analysis in the code repository gives further insight into how the data is formatted into tables and the exact numbers for review.

Table 8: Excerpts from Fundamental section for META

Fundamental section: (llama3-8b-8192) The fundamental data analysis reveals a company with a strong financial position, with a growing revenue, increasing profitability, and high operational efficiency. The net profit margin has been consistently high, ranging from 34.5% to 35.5% over the past few years, indicating a strong ability to maintain profitability. The EPS has also been increasing, with a growth rate of 16.5% from 2020 to 2022, suggesting a healthy pace of earnings growth.
Fundamental section: (gpt-4o) **Revenue and Profitability Growth:** META has demonstrated strong financial health with a consistent upward trend in total revenue, which reached $40.589 billion in the latest quarter. The net income has also seen substantial growth, reaching $15.688 billion, indicating a significant improvement in profitability. The net profit margin has improved to 38.65%, showcasing enhanced operational efficiency. **Earnings Per Share (EPS):** The EPS has increased significantly from $1.76 in December 2022 to $6.20 in September 2024, reflecting strong earnings growth. **Growth Potential:** META's revenue growth of 23.19% and net income growth of 86.79% year-over-year highlight its robust growth potential. The EPS growth of 89.94% further underscores the company's potential for future profitability. **Valuation Metrics:** While the price to earnings and price to sales ratios have fluctuated, the overall financial metrics suggest that META is a potentially attractive investment opportunity.

Table 9: Excerpts from Technical analysis section for META

Technical analysis section: (llama3-8b-8192) The technical analysis reveals a strong underlying trend, with a consistent upward trend in price and momentum. The RSI indicator suggests that the stock may be due for a correction or consolidation period, which could provide a buying opportunity for investors. The Bollinger Bands indicator suggests a relatively low level of volatility, which may be a sign of a stable market.
Technical analysis section: (gpt-4o) **Price Trends:** META's stock has shown a strong upward trend, with increasing Simple Moving Averages (SMA) for both 50-week and 20-week periods. This indicates a bullish trend in the stock's price. **Momentum:** The Relative Strength Index (RSI) is at 64.82, below the overbought threshold of 70, suggesting there is still room for upward movement without immediate risk of a reversal. **Volatility:** The Bollinger Band Percentage is at 84.52%, below the overbought level of 100%, indicating moderate volatility and stable price movement.

4.4. Combined Analysis crew

We have applied this crew for analysis of the NVDA stock. We have not been able to successfully apply llama3-8b-8192 model as it seems the model and the embedding model struggle to handle all 4 data sources and combine the information with a final agent. Therefore, for this crew we compare the output of gpt-4o and gpt-4o-mini. This crew have a lot going on with 4 agents processing different data sources, so there is an expectation that the results may vary between each model. The final agent is a stock advisor, and both models are able to produce a report that gives a rationale for the recommendation to whether buy, hold, or sell the stock. They both arrive at the same conclusion with Hold for the stock but produce quite different arguments with respect to news and SEC filing MD&A section.

Table 10: Excerpts from NVDA combined analysis

News and SEC Filing section: (gpt-4o-mini) **Positive Market Sentiment:** The current market sentiment towards NVDA is overwhelmingly positive, driven by the successful launch of the new AI model, Llama-3.1-Nemotron-70B-Instruct, which has outperformed competitors. This innovation, along with strong demand for GPUs from hyperscalers, indicates a robust market for NVDA's products. **Strategic Growth Opportunities:** Management's discussion in the SEC filings highlights the company's focus on accelerated computing and expansion into AI and other computationally intensive fields. The formation of strategic partnerships, such as with healthcare startup Aidoc, further diversifies revenue streams and solidifies NVDA's market position.

News and SEC Filing section: (gpt-4o)

Market Sentiment and News:
Positive Developments: NVIDIA's expansion in AI partnerships and its position as a leader in AI supercomputing chips are strong growth drivers. The company's strategic focus on the Data Center segment and significant capital investments further bolster its growth prospects.
Challenges and Risks: Bearish options market activity and potential regulatory challenges, such as new export restrictions on AI chips, pose risks to NVDA's stock performance. The mixed market sentiment reflects these uncertainties.

SEC Filing Insights:
The Management's Discussion and Analysis section highlights strategic initiatives, including a $25.0 billion share repurchase program and capital investments between $3.5 billion and $4.0 billion, indicating confidence in the company's financial position and growth strategy.

5. Discussion

The results show some of the challenges with autonomous agents and also LLM models as they will produce results with a random component. The crews spend several iterations before they produce the final result, and each query or interaction between the agents consumes tokens / API call. For models that have a larger cost per token / API call, like gpt-4o, this is not favorable and needs to be considered if these are to be used in a crew. The max tokens parameter can be used to control the tokens consumption during a crew run. However, being too restrictive in terms of max token usage will limit the capability of the crew.

When there is a need to have more consistent results between crew runs, it is important that the temperature parameter can be adjusted to 0 so that the LLM will produce less “creative” or random results.

It is an interesting finding that a relatively small model such as Llama3-8b is able to process the dataset and extract both key events and sentiment analysis for the recent news dataset. The token consumption is also within the limits of the free tier for Groq hosted model.

5.1. Crew performance

In general, it is important to keep a reasonable number of agents to be able to stay within the usage limits of the LLMs and also to be able to reason about how the crew actually works. For the Combined Analysis crew, there are 5 agents, and the content that the final agent will process may be too extensive for the agent to process. Therefore we consider simplifying the tasks of the agents and keeping the overall setup as simple as possible by dividing a complicated task into more digestible tasks.

There is quite some variation in the results from the crews with different models. Some of the variations can likely be attributed to the different embedding models that are used, not necessarily to the LLM. But also, the context window between llama3-8b-8192 and gpt-4o is quite significant, so it is expected that gpt-4o models are able to process larger chunks of text in the same interaction.

5.2. Llama3-8b model

In general, we see that the Llama open-source model is quite capable of processing much of the data sources that are set up in this project. However, compared to the gpt-4o

models, it is shown through the results that it has limited capability to produce detailed responses that elaborate on a particular topic or include detailed numbers in its response. For example, when considering the Risk Factors for NVDA 10-K report, it produces mainly a condensed list of the risk factors in the filing, while gpt-4o is able to categorize the risk factors. For market data analysis, it produces an output that summarizes the data in an overall and condensed form, while the gpt-4o model is able to produce content where it has extracted the specific relevant numbers from the original data.

5.3. gpt-4o models

Both gpt-4o models are capable of producing content that uses the data that the agent has access to through the tools that are provided. From a cost perspective, the gpt-4o-mini would be favorable to use if the crews were to analyze multiple tickers as part of a daily analysis pipeline. For the combined analysis crew example, the gpt-4o model is able to produce a slightly more elaborate report and highlight different aspects with respect to news data and SEC filing MD&A section.

6. Conclusion

In this practical project, we have applied state-of-the-art technology to build Multi-Agent systems with LLMs that are able to perform financial analysis and produce textual content that can help analysts get a condensed overview of the data sources that have been integrated into the system. A goal of the project was to explore the boundaries of what is possible to do with freely accessible data sources together with cost-effective general-purpose LLMs. Most of the components that the FinMAS system builds upon have been released within the last year, and thus, we intend for this project to showcase how these components can be put together to build an RAG application for solving financial analysis tasks for free or with a very low cost.

A challenge that had to be solved well during the project was data preprocessing before sending the data as context to the LLM when solving the tasks. When the data was handled poorly, we got responses that had a higher likelihood of containing hallucinations. We solved this by targeting specific sections of SEC filings and also stripping unnecessary or unreadable elements from the news and filing content before the embeddings were created. It is important to emphasize that the choice of embedding model may impact the result of the analysis as much as the choice of LLM for the agents. For fundamental and technical analysis data, the data were condensed into tables with a quarterly and weekly timeframe.

In some cases, we see that a smaller model with a small context window like Llama 3 8b is not sufficient for the complex task at hand, and thus, it is necessary to apply a larger model with a larger context window to complete the task.

FinMAS attempts to close the gap between new LLM technologies and practical applications in the financial domain, and we offer a simple yet powerful tool for anyone in the financial domain to easily get actionable insights while having the flexibility to configure the tool to adapt to the user's needs. As financial analysis lays the foundation for decision-making with respect to stock portfolios, it is important to have confidence in the result, and therefore, we have focused on providing transparency into the FinMAS system by showing the user the data that is being fed to the system for review and control.

For future development into RAG systems for financial tasks we see through this project that there needs to be put emphasis on the data processing step. In these types of systems, a data quality component should be present, as any bad-quality data will adversely affect the crew output.

References

[1] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS '20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 9459–9474.

[2] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, in ICML '09. New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 873–880. doi: 10.1145/1553374.1553486.

[3] A. Vaswani *et al.*, "Attention Is All You Need," Aug. 01, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[4] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," Jul. 22, 2020, *arXiv*: arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[5] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," Mar. 04, 2022, *arXiv*: arXiv:2203.02155. doi: 10.48550/arXiv.2203.02155.

[6] Y. Nie *et al.*, "A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges," Jun. 15, 2024, *arXiv*: arXiv:2406.11903. doi: 10.48550/arXiv.2406.11903.

[7] J. Lee, N. Stevens, S. C. Han, and M. Song, "A Survey of Large Language Models in Finance (FinLLMs)," Feb. 03, 2024, *arXiv*: arXiv:2402.02315. doi: 10.48550/arXiv.2402.02315.

[8] Q. Xie *et al.*, "FinBen: A Holistic Financial Benchmark for Large Language Models," Jun. 18, 2024, *arXiv*: arXiv:2402.12659. doi: 10.48550/arXiv.2402.12659.

[9] A. Shah, A. Gullapalli, R. Vithani, M. Galarnyk, and S. Chava, "FiNER-ORD: Financial Named Entity Recognition Open Research Dataset," Sep. 06, 2024, *arXiv*: arXiv:2302.11157. doi: 10.48550/arXiv.2302.11157.

[10] X. Li *et al.*, "Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? A Study on Several Typical Tasks," Oct. 10, 2023, *arXiv*: arXiv:2305.05862. doi: 10.48550/arXiv.2305.05862.

[11] A. Sinha and T. Khandait, "Impact of News on the Commodity Market: Dataset and Results," Sep. 09, 2020, *arXiv*: arXiv:2009.04202. doi: 10.48550/arXiv.2009.04202.

[12] R. Jørgensen, O. Brandt, M. Hartmann, X. Dai, C. Igel, and D. Elliott, "MultiFin: A Dataset for Multilingual Financial NLP," in *Findings of the Association for Computational Linguistics: EACL 2023*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 894–909. doi: 10.18653/v1/2023.findings-eacl.66.

[13] Y. Tang, Y. Yang, A. H. Huang, A. Tam, and J. Z. Tang, "FinEntity: Entity-level Sentiment Classification for Financial Texts," Oct. 18, 2023, *arXiv*: arXiv:2310.12406. doi: 10.48550/arXiv.2310.12406.

[14] A. Sinha, S. Kedas, R. Kumar, and P. Malo, "SEntFiN 1.0: Entity-Aware Sentiment Analysis for Financial News," *J. Assoc. Inf. Sci. Technol.*, vol. 73, no. 9, pp. 1314–1335, Sep. 2022, doi: 10.1002/asi.24634.

[15] T. Konstantinidis, G. Iacovides, M. Xu, T. G. Constantinides, and D. Mandic, "FinLlama: Financial Sentiment Classification for Algorithmic Trading Applications," Mar. 18, 2024, *arXiv*: arXiv:2403.12285. doi: 10.48550/arXiv.2403.12285.

[16] A. H. Huang, H. Wang, and Y. Yang, "FinBERT: A Large Language Model for Extracting Information from Financial Text," *Contemp. Account. Res.*, vol. 40, no. 2, pp. 806–841, 2023, doi: 10.1111/1911-3846.12832.

[17] G. Bhatia, E. M. B. Nagoudi, H. Cavusoglu, and M. Abdul-Mageed, "FinTral: A Family of GPT-4 Level Multimodal Financial Large Language Models," Jun. 14, 2024, *arXiv*: arXiv:2402.10986. doi: 10.48550/arXiv.2402.10986.

[18] Q. Xie *et al.*, "PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance," Jun. 08, 2023, *arXiv*: arXiv:2306.05443. doi: 10.48550/arXiv.2306.05443.

[19] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," Feb. 27, 2023, *arXiv*: arXiv:2302.13971. doi: 10.48550/arXiv.2302.13971.

[20] A. Q. Jiang *et al.*, "Mistral 7B," Oct. 10, 2023, *arXiv*: arXiv:2310.06825. doi: 10.48550/arXiv.2310.06825.

[21] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," Oct. 16, 2021, *arXiv*: arXiv:2106.09685. doi: 10.48550/arXiv.2106.09685.

[22] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 23, 2023, *arXiv*: arXiv:2305.14314. doi: 10.48550/arXiv.2305.14314.

[23] T. Guo *et al.*, "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," Apr. 18, 2024, *arXiv*: arXiv:2402.01680. doi: 10.48550/arXiv.2402.01680.

[24] João Moura, *crewAI*. (Sep. 15, 2024). Accessed: Sep. 15, 2024. [Online]. Available: https://github.com/crewAIInc/crewAI.

[25] Q. Wu *et al.*, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," Oct. 03, 2023, *arXiv*: arXiv:2308.08155. doi: 10.48550/arXiv.2308.08155.

[26] S. Hong *et al.*, "MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework," Nov. 06, 2023, *arXiv*: arXiv:2308.00352. doi: 10.48550/arXiv.2308.00352.

[27] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative Agents for 'Mind' Exploration of Large Language Model Society," Nov. 02, 2023, *arXiv*: arXiv:2303.17760. doi: 10.48550/arXiv.2303.17760.

[28] J. Liu, *LlamaIndex*. (Nov. 2022). Python. Accessed: Oct. 21, 2024. [Online]. Available: https://github.com/jerryjliu/llama_index.

[29] Y. Li, Y. Yu, H. Li, Z. Chen, and K. Khashanah, "TradingGPT: Multi-Agent System with Layered Memory and Distinct Characters for Enhanced Financial Trading Performance," Sep. 07, 2023, *arXiv*: arXiv:2309.03736. doi: 10.48550/arXiv.2309.03736.

[30] F. Xing, "Designing Heterogeneous LLM Agents for Financial Sentiment Analysis," *ACM Trans. Manag. Inf. Syst.*, p. 3688399, Aug. 2024, doi: 10.1145/3688399.

[31] Z. P. Gijsbertha, "Multi-agent conversations for sentiment analysis of the cryptocurrency market," Tilburg University, 2024. [Online]. Available: https://arno.uvt.nl/show.cgi?fid=171914.

[32] P. Malo, A. Sinha, P. Takala, P. Korhonen, and J. Wallenius, "Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts," Jul. 23, 2013, *arXiv*: arXiv:1307.5336. doi: 10.48550/arXiv.1307.5336.

[33] M. Maia *et al.*, "WWW'18 Open Challenge: Financial Opinion Mining and Question Answering," in *Companion Proceedings of the The Web Conference 2018*, in WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2018, pp. 1941–1942. doi: 10.1145/3184558.3192301.

[34] A. Shah, S. Paturi, and S. Chava, "Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis," May 13, 2023, *arXiv*: arXiv:2305.07972. doi: 10.48550/arXiv.2305.07972.

[35] J. Choi, J. Yun, K. Jin, and Y. Kim, "Multi-News+: Cost-efficient Dataset Cleansing via LLM-based Data Annotation," Sep. 23, 2024, *arXiv*: arXiv:2404.09682. doi: 10.48550/arXiv.2404.09682.

[36] Q. Li *et al.*, "LaFFi: Leveraging Hybrid Natural Language Feedback for Fine-tuning Language Models," Dec. 31, 2023, *arXiv*: arXiv:2401.00907. doi: 10.48550/arXiv.2401.00907.

[37] Groq, "Groq is Fast AI Inference." Accessed: Nov. 17, 2024. [Online]. Available: https://groq.com/

[38] OpenAI, "GPT-4o mini: advancing cost-efficient intelligence." Accessed: Nov. 18, 2024. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Appendix

Code Repository

The GitHub repository contains the web app code, notebooks, output examples, and sample data: https://github.com/KevorkSulahian/agentic-llm-for-better-results/

Notebooks

One notebook is setup for each crew configuration, so that it is possible to run a crew analysis directly from a notebook instead of using the UI / panel app. Especially for the market data analysis crew it is possible to see in detail the data that is being sent to the crew.
Notebooks folder:
https://github.com/KevorkSulahian/agentic-llm-for-better-results/tree/main/notebooks

Documentation

For further explanation of the structure of the FinMAS system with tutorials, please visit the documentation at: https://kevorksulahian.github.io/agentic-llm-for-better-results/

Example outputs

The output folder in the code repository contains examples of output from crew runs. Excerpts from these outputs were highlighted in the results section. These examples can also be viewed in the examples section of the documentation site: https://kevorksulahian.github.io/agentic-llm-for-better-results/examples_index/

Experiments Section

For the purposes of development and testing of different tools collaboratively, the team has the utilized folder named experiments where each folder inside is an individual attempt at one of the components that build up the FinMAS app. The folder also included guides and tutorials on how to use and leverage some of the APIs and libraries like CrewAI. The experiments have helped the research team to investigate different parts of the system in an isolated environment and in closer detail before being incorporated into the system.