Large Language Models in Finance: A Survey

Yinheng Li*
yl4039@columbia.edu
Columbia University
New York, NY, USA

Shaofei Wang*
sw3316@columbia.edu
Columbia University
New York, NY, USA

Han Ding*
hd2412@columbia.edu
Columbia University
New York, NY, USA

Hang Chen*
hc2798@nyu.edu
New York University
New York, NY, USA

Abstract

Recent advances in large language models (LLMs) have opened new possibilities for artificial intelligence applications in finance. In this paper, we provide a practical survey focused on two key aspects of utilizing LLMs for financial tasks: existing solutions and guidance for adoption.

First, we review current approaches employing LLMs in finance, including leveraging pretrained models via zero-shot or few-shot learning, fine-tuning on domain-specific data, and training custom LLMs from scratch. We summarize key models and evaluate their performance improvements on financial natural language processing tasks.

Second, we propose a decision framework to guide financial professionals in selecting the appropriate LLM solution based on their use case constraints around data, compute, and performance needs. The framework provides a pathway from lightweight experimentation to heavy investment in customized LLMs.

Lastly, we discuss limitations and challenges around leveraging LLMs in financial applications. Overall, this survey aims to synthesize the state-of-the-art and provide a roadmap for responsibly applying LLMs to advance financial AI.

Keywords: Large Language Models, Generative AI, Natural Language Processing, Finance

1 Introduction

Recent advances in artificial intelligence, especially in natural language processing, have led to the development of powerful large language models (LLMs) like ChatGPT[33]. These models have demonstrated impressive capabilities in understanding, generating, and reasoning about natural language. The finance industry could benefit from applying LLMs, as effective language understanding and generation can inform trading, risk modeling, customer service, and more.

In this survey, we aim to provide a practical overview focused on two key aspects of utilizing LLMs for financial applications:

- Existing solutions and models that employ LLMs for various finance tasks. We summarize key techniques like finetuning pretrained LLMs and training domain-specific LLMs from scratch.
- Guidance on the decision process for applying LLMs in finance. We discuss factors to consider regarding whether LLMs are suitable for a task, cost/benefit trade-offs, risks, and limitations.

By reviewing current literature and developments, we hope to give an accessible synthesis of the state-of-the-art along with considerations for adopting LLMs in finance. This survey targets financial professionals and researchers exploring the intersection of AI and finance. It may also inform developers applying LLM solutions for the finance industry. The remainder of the paper is organized as follows. Section 2 covers background on language modeling and recent advances leading to LLMs. Section 3 surveys current AI applications in finance and the potential for LLMs to advance in these areas. Sections 4 and 5 provide LLM solutions and decision guidance for financial applications. Finally, Sections 6 and 7 discuss risks, limitations, and conclusions.

2 Basics of Language Models

A language model is a statistical model that is trained on extensive text corpora to predict the probability distribution of word sequences [4]. Let's consider a sequence of words denoted as $W = w_1, w_2, ..., w_n$, where w_i represents the i-th word in the sequence. The goal of a language model is to calculate the probability $P(W)$, which can be expressed as:

$$
\begin{aligned}
P(W) &= P(w_1, w_2, ..., w_n) \\
&= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \\
&\quad ...P(w_n|w_1, w_2, ..., w_{n-1})
\end{aligned}
$$

The conditional probability $P(w_i|w_1, w_2, ..., w_{i-1})$ captures the likelihood of word w_i given the preceding words. Over the past few decades, language model architectures have undergone significant evolution. Initially, n-gram models represented word sequences as Markov processes [3], assuming that the probability of the next word depends solely on the preceding $(n-1)$ words. For example, in a bigram model,

*All authors contributed equally to this research. Order is random

the probability of a word is only conditioned on the previous word.

Later, Recurrent Neural Network (RNN)-based models like LSTM [20] and GRU [10] emerged as neural network solutions, which are capable of capturing long-term dependencies in sequential data. However, in 2017, the introduction of the transformer architecture [46] revolutionized language modeling, surpassing the performance of RNNs in tasks such as machine translation. Transformers employ self-attention mechanisms to model parallel relationships between words, facilitating efficient training on large-scale datasets. Prominent transformer-based models include GPT (Generative Pretrained Transformer) [5, 48], which is decoder-only framework, BERT (Bidirectional Encoder Representations from Transformers) [13], which is encoder-only framework, and T5 (Text-to-Text Transfer Transformer) [38], which leverages both encoder and decoder structures. These models have achieved state-of-the-art results on various natural language processing (NLP) tasks through transfer learning.

It is important to note that the evolution of language models has mainly been driven by advancements in computational power, the availability of large-scale datasets, and the development of novel neural network architectures. These models have significantly enhanced language understanding and generation capabilities, enabling their application across a wide range of industries and domains.

3 Overview of AI Applications in Finance

3.1 Current AI Applications in Finance

Artificial Intelligence (AI) has witnessed extensive adoption across various domains of finance in recent years [19]. In this survey, we focus on key financial applications, including trading and portfolio management [67], financial risk modeling [30], financial text mining [21, 36], and financial advisory and customer services [41]. While this list is not exhaustive, these areas have shown significant interest and high potential with the advancement of AI.

Trading and portfolio management have been early adopters of machine learning and deep learning models within the finance industry. The primary objective of trading is to forecast prices and generate profits based on these predictions. Initially, statistical machine learning methods such as Support Vector Machines (SVM) [23], Xgboost [68], and tree-based algorithms were utilized for profit and loss estimation. However, the emergence of deep neural networks introduced techniques like Recurrent Neural Networks (RNN), particularly Long Short-Term Memory (LSTM) networks [40], Convolutional Neural Networks (CNN), and transformers [51], which have proven effective in price forecasting. Additionally, reinforcement learning [47] has been applied to automatic trading and portfolio optimization.

Financial risk modeling encompasses various applications of machine learning and deep learning models. For instance, McKinsey & Company has developed a deep learning-based solution for financial fraud detection by leveraging user history data and real-time transaction data [39]. Similar approaches have been employed in credit scoring [29, 52] and bankruptcy or default prediction [8].

Financial text mining represents a popular area where deep learning models and natural language processing techniques are extensively utilized. According to [35], there are over 40 research publications on this topic. Financial text mining aims to extract valuable information from large-scale unstructured data in real-time, enabling more informed decision-making in trading and risk modeling. For example, [15] employs financial market sentiment extracted from news articles to forecast the direction of the stock market index.

Applying AI in financial advisory and customer-related services is an emerging and rapidly growing field. AI-powered chatbots, as discussed in [32], already provide more than 37% of supporting functions in various e-commerce and e-service scenarios. In the financial industry, chatbots are being adopted as cost-effective alternatives to human customer service, as highlighted in the report "Chatbots in consumer finance" [2]. Additionally, banks like JPMorgan are leveraging AI services to provide investment advice, as mentioned in a report by CNBC [42].

The current implementation of deep learning models offers significant advantages by efficiently extracting valuable insights from vast amounts of data within short time frames. This capability is particularly valuable in the finance industry, where timely and accurate information plays a crucial role in decision-making processes. With the emergence of LLMs, even more tasks that were previously considered intractable become possible, further expanding the potential applications of AI in the finance industry.

3.2 Advancements of LLMs in Finance

LLMs offer numerous advantages over traditional models, particularly in the field of finance. Firstly, LLMs leverage their extensive pre-training data to effectively process common-sense knowledge, enabling them to understand natural language instructions. This is valuable in scenarios where supervised training is challenging due to limited labeled financial data or restricted access to certain documents. LLMs can perform tasks through zero-shot learning [26], as demonstrated by their satisfactory performance in sentiment classification tasks across complex levels [65]. For similar text mining tasks on financial documents, LLMs can automatically achieve acceptable performance.

Compared to other supervised models, LLMs offer superior adaptation and flexibility. Instead of training separate models for specific tasks, LLMs can handle multiple tasks

by simply modifying the prompt under different task instructions [6]. This adaptability does not require additional training, enabling LLMs to simultaneously perform sentiment analysis, summarization, and keyword extraction on financial documents.

LLMs excel at breaking down ambiguous or complex tasks into actionable plans. Applications like Auto-GPT [1], Semantic Kernel [31], and LangChain [7] have been developed to showcase this capability. In this paper, we refer to this as **Tool Augmented Generation**. For instance [37], Auto-GPT can optimize a portfolio with global equity ETFs and bond ETFs based on user-defined goals. It formulates detailed plans, including acquiring financial data, utilizing Python packages for Sharpe ratio optimization, and presenting the results to the user. Previously, achieving such end-to-end solutions with a single model was unfeasible. This property makes LLMs an ideal fit for financial customer service or financial advisory, where they can understand natural language instructions and assist customers by leveraging available tools and information.

While the application of LLMs in finance is really promising, it is crucial to acknowledge their limitations and associated risks, which will be further discussed in Section 6.

4 LLM Solutions for Finance

4.1 Utilizing Few-shot/Zero-shot Learning in Finance Applications

Accessing LLM solutions in finance can be done through two options: utilizing an API from LLM service providers or employing open-source LLMs. Companies like OpenAI[1], Google[2], and Microsoft[3] offer LLM services through APIs. These services not only provide the base language model capabilities but also offer additional features tailored for specific use cases. For example, OpenAI's APIs include functionalities for chat, SQL generation, code completion, and code interpretation. While there is no dedicated LLM service exclusively designed for finance applications, leveraging these general-purpose LLM services can be a viable option, especially for common tasks. An example in this work [16] demonstrates the use of OpenAI's GPT4 service for financial statement analysis.

In addition to LLM services provided by tech companies, open-source LLMs can also be applied to financial applications. Models such as LLaMA [45], BLOOM [54], Flan-T5 [12], and more are available for download from the Hugging Face model repository[4]. Unlike using APIs, hosting and running these open-source models would require self-hosting.

Similar to using LLM APIs, zero-shot or few-shot learning approaches can be employed with open-source models. Utilizing open-source models offers greater flexibility as the model's weights are accessible, and the model's output can be customized for downstream tasks. Additionally, it provides better privacy protection as the model and data remain under user's control. However, working with open-source models also has its drawbacks. Reported evaluation metrics suggest a performance gap between open-source models and proprietary models. For certain downstream tasks, zero-shot or few-shot learning may not yield optimal performance. In such cases, fine-tuning the model with labeled data, expertise, and computational resources is necessary to achieve satisfactory results. This may explain why, at the time of writing this paper, no direct examples of open-source models applied to financial applications have been found. In Section 5, we provide a more detailed discussion of which option is more favorable under different circumstances.

4.2 Fine-tuning a Model

Fine-tuning LLMs in the finance domain can enhance domain-specific language understanding and contextual comprehension, resulting in improved performance in finance-related tasks and generating more accurate and tailored outputs.

4.2.1 Common Techniques for LLM Fine-tuning. Modern techniques for fine-tuning LLMs typically fall into two main categories: standard fine-tuning and instructional fine-tuning.

In standard fine-tuning, the model is trained on the raw datasets without modification. The key context, question, and desired answer are directly fed into the LLM, with the answer masked during training so that the model learns to generate it. Despite its simplicity, this approach is widely effective.

Instruct fine-tuning [34] involves creating task-specific datasets that provide examples and guidance to steer the model's learning process. By formulating explicit instructions and demonstrations in the training data, the model can be optimized to excel at certain tasks or produce more contextually relevant and desired outputs. The instructions act as a form of supervision to shape the model's behavior.

Both methods have their merits: standard fine-tuning is straightforward to implement, while instructional fine-tuning allows for more precise guidance of the model. The ideal approach depends on the amount of training data available and the complexity of the desired behaviors. However, both leverage the knowledge already embedded in LLMs and fine-tune them for enhanced performance on downstream tasks.

In addition to the above methods, techniques such as Low-Rank Adaptation (LoRA)[22] and quantization[18] can enable fine-tuning with significantly lower computational requirements.

---

[1]https://openai.com/product
[2]https://bard.google.com/
[3]https://azure.microsoft.com/en-us/products/cognitive-services/openai-service
[4]https://huggingface.co/models

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen

Table 1. Quick Overview of Finetuned Finance LLM

Model Name	Finetune data size (samples)	Training budget	Model architecture	Release time
FinMA-7B	Raw: 70k, Instruction: 136k	8 A100 40GB GPUs	LLaMA-7B	Jun 2023
FinMA-30B	Raw: 70k, Instruction: 136k	128 A100 40GB GPUs	LLaMA-30B	Jun 2023
Fin-GPT(V1/V2/V3)	50K	< $300 per training	ChatGLM, LLaMA	July 2023
Instruct-FinGPT	10K Instruction	8 A100 40GB GPUs, ~1 hr	LLaMA-7B	Jun 2023
Fin-LLaMA[53]	16.9K Instruction	NA	LLaMA-33B	Jun 2023
Cornucopia(Chinese)[61]	12M instruction	NA	LLaMA-7B	Jun 2023

Table 2. Quick Overview of from scratch trained Finance LLMs

Pretrained LLM	Corpus size(tokens)	Training budget(A100·hours)	Model architecture	Release time
BloomBergGPT	363B Finance tokens + 345B public tokens	1,300,000	50B-BLOOM	May 2023
XuanYuan2.0	366B for pre-training + 13B for finetuning	Not released	176B-BLOOM	May 2023
Fin-T5	80B Finance tokens	Days/weeks	770M-T5	Feb 2023

LoRA allows for fine-tuning the low-rank decomposed factors of the original weight matrices instead of the full matrices. This approach drastically reduces the number of trainable parameters, enabling training on less powerful hardware and shortening the total training time.

Another impactful approach is to use reduced numerical precisions such as bfloat16 [24] or float16 instead of float32. By halving the bit-width, each parameter only occupies 2 bytes instead of 4 bytes, reducing memory usage by 50%. This also accelerates computation by up to 2x since smaller data types speed up training. Moreover, the reduced memory footprint enables larger batch sizes, further boosting throughput.

4.2.2 Fine-tuned finance LLM evaluation. The performance of fine-tuned finance LLMs can be evaluated in two categories: finance classification tasks and finance generative tasks. In finance classification, we consider tasks such as Sentiment Analysis and News Headline Classification. In finance generative tasks, our focus is on Question Answering, News Summarization, and Named Entity Recognition. Table 1 provides detailed information about all the fine-tuned finance LLMs. Among the various fine-tuned LLMs, we will focus on discussing three of them: (1) PIXIU (also known as FinMA)[56], fine-tuned LLaMA on 136K task-specific instruction samples. (2) FinGPT[58], it presents a end-to-end framework for training and applying FinLLMs in the finance industry. FinGPT utilizes the lightweight Low-rank Adaptation (LoRA) technique to fine-tune open-source LLMs (such as LLaMA and ChatGLM) using approximately 50k samples. However, FinGPT's evaluation is only limited to finance classification tasks. (3) Instruct-FinGPT[63], on the other hand,

fine-tunes LLaMA on 10k instruction samples derived from two Financial Sentiment Analysis Datasets and also solely evaluates performance on finance classification tasks.

Based on the reported model performance, we summarize our findings as below:

- Compared to the original base LLM (LLaMA) and other open-source LLMs (BLOOM, OPT[64], ChatGLM[14, 62]), all fine-tuned finance LLMs exhibit significantly better performance across all finance-domain tasks reported in the papers, especially classification tasks.
- The fine-tuned finance LLMs outperform BloombergGPT[55] in most finance tasks reported in the papers.
- When compared to powerful general LLMs like Chat-GPT and GPT-4, the fine-tuned finance LLMs demonstrate superior performance in most finance classification tasks, which indicates their enhanced domain-specific language understanding and contextual comprehension abilities. However, in finance generative tasks, the fine-tuned LLMs show similar or worse performance, suggesting the need for more high-quality domain-specific datasets to improve their generative capabilities.

4.3 Pretrain from Scratch

The objective of training LLMs from scratch is to develop models that have even better adaptation to the finance domain. Table 2 presents the current finance LLMs that have been trained from scratch: BloombergGPT, Xuan Yuan 2.0 [66], and Fin-T5[28].

As shown in Table 2, there is a trend of combining public datasets with finance-specific datasets during the pretraining

phase. Notably, BloombergGPT serves as an example where the corpus comprises an equal mix of general and finance-related text. It is worth mentioning that BloombergGPT primarily relies on a subset of 5 billion tokens that pertain exclusively to Bloomberg, representing only 0.7% of the total training corpus. This targeted corpus contributes to the performance improvements achieved in finance benchmarks.

Both BloombergGPT and Fin-T5 have demonstrated superior performance compared to their original models like BLOOM176B and T5, respectively. These tasks encompass activities such as market sentiment classification, multi-categorical and multi-label classification, and more. BloombergGPT achieves an impressive average score of 62.51, surpassing the open-source BLOOM176B model, which only attains a score of 54.35. Similarly, Fin-T5 demonstrates its excellence with an average score of 81.78, outperforming the T5 model's score of 79.56. Notably, BloombergGPT was evaluated using an internal benchmark specifically designed by Bloomberg. The results of this evaluation showcased remarkable improvements, as BloombergGPT achieved an average score of 62.47, surpassing the performance of BLOOM176B, which only attained a score of 33.39. This outcome highlights that even when the internal private training corpus constitutes less than 1% of the total training corpus, it can still lead to substantial enhancements in evaluating tasks within the same domain and distribution.

On finance-related generative tasks such as Question Answering, Named Entity Recognition, summarization, both models exhibited significantly better results compared to their respective general models by a considerable margin. Specifically, BloombergGPT achieved an impressive score of 64.83, surpassing BLOOM-176B's score of 45.43. Similarly, Fin-T5 outperformed T5 with a score of 68.69, while T5 scored 66.06. These findings further highlight the models' superior performance in generating finance-related content when compared to their general-purpose counterparts.

Although these models are not as powerful as closed-source models like GPT-3 or PaLM[11], they demonstrate similar or superior performance compared to similar-sized public models. In evaluations on various general generative tasks, such as BIG-bench Hard, knowledge assessments, reading comprehension, and linguistic tasks, BloombergGPT exhibited comparable or superior performance compared to similar-sized public models, albeit slightly inferior to larger models like GPT-3 or PaLM. Overall, BloombergGPT showcased commendable performance across a wide range of general generative tasks, positioning it favorably among models of comparable size. This indicates that the model's enhanced capabilities in finance-related tasks do not come at the expense of its general abilities.

5 Decision Process in Applying LLM to Financial Applications

5.1 Determining the Need for a LLM

Before exploring LLM solutions, it is essential to ascertain whether employing such a model is truly necessary for the given task. The advantages of LLMs over smaller models can be summarized as follows, as outlined in the work by Yang et al. [59]:

Leveraging Pretraining Knowledge: LLMs can utilize the knowledge acquired from pretraining data to provide solutions. If a task lacks sufficient training data or annotated data but requires common-sense knowledge, an LLM may be a suitable choice.

Reasoning and Emergent Abilities: LLMs excel at tasks that involve reasoning or emergent abilities [49]. This property makes LLMs well-suited for tasks where task instructions or expected answers are not clearly defined, or when dealing with out-of-distribution data. In the context of financial advisory, client requests in customer service often exhibit high variance and complex conversations. LLMs can serve as virtual agents to provide assistance in such cases.

Orchestrating Model Collaboration: LLMs can act as orchestrators between different models and tools. For tasks that require collaboration among various models, LLMs can serve as orchestrators to integrate and utilize these tools together [1, 7, 31]. This capability is particularly valuable when aiming for a robust automation of a model solution pipeline.

While LLMs offer immense power, their use comes with a significant cost, whether utilizing a third-party API [33] or fine-tuning an open-source LLM. Therefore, it is prudent to consider conventional models before fully committing to LLMs. In cases where the task has a clear definition (e.g., regression, classification, ranking), there is an ample amount of annotated training data, or the task relies minimally on common-sense knowledge or emerging capabilities like reasoning, relying on LLMs may not be necessary or justified initially.

5.2 A general decision guidance for applying LLMs on finance tasks

Once the decision has been made to utilize LLMs for a finance task, a decision guidance framework can be followed to ensure efficient and effective implementation. The framework, illustrated in Figure 1, categorizes the usage of LLMs into four levels based on computational resources and data requirements. By progressing through the levels, costs associated with training and data collection increase. It is recommended to start at Level 1 and move to higher levels (2, 3, and 4) only if the model's performance is not satisfactory. The following section provides detailed explanations of the decision and

Figure 1. Decision process flow chart

action blocks at each level. Table **??** presents an approximate cost range for different options, based on pricing from various third-party services like AWS and OpenAI.

5.2.1 Level 1: Zero-shot Applications. The first decision block determines whether to use an existing LLM service or an open-source model. If the input question or context involves confidential data, it is necessary to proceed with the 1A action block, which involves self hosting an open-source LLM. As of July 2023, several options are available, including LLAMA[45], OpenLLAMA[17], Alpaca[44], and Vicuna[9]. LLAMA offers models with sizes ranging from 7B to 65B, but they are limited to research purposes. OpenLLAMA provides options for 3B, 7B, and 13B models, with support for commercial usage. Alpaca and Vicuna are fine-tuned based on LLAMA, offering 7B and 13B options. Deploying your own LLM requires a robust local machine with a suitable GPU, such as NVIDIA-V100 for a 7B model or NVIDIA-A100, A6000 for a 13B model.

If data privacy is not a concern, selecting third-party LLMs such as GPT3.5/GPT4 from OpenAI or BARD from Google is recommended. This option allows for lightweight experiments and early performance evaluation without significant deployment costs. The only cost incurred would be the fees associated with each API call, typically based on input length and the token count of the model's output.

5.2.2 Level 2: Few-shot Applications. If the model's performance at Level 1 is not acceptable for the application, few-shot learning can be explored if there are several example questions and their corresponding answers available. Few-shot learning has shown advantages in various previous works [5, 48]. The core idea is to provide a set of example questions along with their corresponding answers as context in addition to the specific question being asked. The cost associated with few-shot learning is similar to that of the previous levels, except for the requirement of providing examples each time. Generally, achieving good performance may require using 1 to 10 examples. These examples can be

Options	Development Computational Cost($)	Development Data Cost(samples)	Deployment Computational Cost ($/1k tokens generated)
OpenSource-ZeroShot	-	-	0.006 - 0.037
3rd party-ZeroShot	-	-	0.002 - 0.12
OpenSource-FewShot	-	-	0.006 - 0.037
3rd party-FewShot	-	-	0.002 - 0.12
OpenSource Tool Augmented Generation	Cost of developing tools	-	0.006 - 0.037
3rd party Tool Augmented Generation	Cost of developing tools	-	0.002 - 0.12
OpenSource-Finetune	4-360,000	10,000 - 12,000,000	0.0016 - 0.12
3rd party-Finetune	30-30,000	10,000 - 12,000,000	0.002 - 0.12
Train from Scratch	5,000,000	700,000,000	0.0016 - 0.12

Table 3. Costs of Different LLM Options: This table gives an approximate range of requirements of data and dollar cost. The data and dollar cost requirements for development are estimated based on previous works listed in 2. The third-party deployment costs are listed in https://openai.com/pricing. The open source deployment costs are calculated based on https://openai.com/pricing and https://aws.amazon.com/ec2/pricing/on-demand/. We assume using NVIDIA A100 GPU. The cost of $/ tokens = $ / second * second / 1k tokens, and it typically takes 3 to 33 seconds to generate 1k tokens, depending on model size.

the same across different questions or selected based on the specific question at hand. The challenge lies in determining the optimal number of examples and selecting relevant ones. This process involves experimentation and testing until the desired performance boundary is reached.

5.2.3 Level 3: Tool-Augmented Generation and Finetuning.
If the task at hand is extremely complicated and in-context learning does not yield reasonable performance, the next option is to leverage external tools or plugins with the LLM, assuming a collection of relevant tools/plugins is available. For example, a simple calculator could assist with arithmetic-related tasks, while a search engine could be indispensable for knowledge-intensive tasks such as querying the CEO of a specific company or identifying the company with the highest market capitalization.

Integrating tools with LLMs can be achieved by providing the tool's descriptions. The cost associated with this approach is generally higher than that of few-shot learning due to the development of the tool(s) and the longer input sequence required as context. However, there may be instances where the concatenated tool description is too long, surpassing the input length limit of LLMs. In such cases, an additional step such as a simple tool retrieval or filter might be needed to narrow down the tools for selection. The deployment cost typically includes the cost of using the LLMs as well as the cost of using the tool(s).

If the above options fail to produce satisfactory performance, finetuning the LLMs can be attempted. This stage requires a reasonable amount of annotated data, computational resources (GPU, CPU, etc.), and expertise in tuning language models, as listed in Table ??.

5.2.4 Level 4: Train Your Own LLMs from Scratch.
If the results are still unsatisfactory, the only option left is to train domain-specific LLMs from scratch, similar to what BloombergGPT did. However, this option comes with significant computational costs and data requirements. It typically requires millions of dollars in computational resources and training on a dataset with trillions of tokens. The intricacies of the training process are beyond the scope of this survey, but it is worth noting that it can take several months or even years of effort for a professional team to accomplish.

By following this decision guidance framework, financial professionals and researchers can navigate through the various levels and options, making informed choices that align with their specific needs and resource constraints.

5.3 Evaluation
The evaluation of LLMs in finance can be conducted through various approaches. One direct evaluation method is to assess the model's performance on downstream tasks. Evaluation metrics can be categorized into two main groups: accuracy and performance, based on the taxonomy provided by [57]. The accuracy category can further be divided into metrics for regression (such as MAPE, RMSE, R^2) and metrics for classification (Recall, Precision, F1 score). The performance

category includes metrics or measurements that directly assess the model's performance on the specific task, such as measuring total profit or Sharpe Ratio in a trading-related task. These evaluations can be conducted using historical data, backtest simulations, or online experiments. While performance metrics are often more important in finance, it is crucial to ensure that accuracy metrics align with performance to ensure meaningful decision-making and guard against overfitting.

In addition to task-specific evaluations, general metrics used for LLMs can also be applied. Particularly, when evaluating the overall quality of an existing LLM or a fine-tuned one, comprehensive evaluation systems like the one presented in [27] can be utilized. This evaluation system covers tasks for various scenarios and incorporates metrics from different aspects, including accuracy, fairness, robustness, bias, and more. It can serve as a guide for selecting a language model or evaluating one's own model in the context of finance applications.

5.4 Limitations

While significant progress has been made in applying LLMs to revolutionize financial applications, it is important to acknowledge the limitations of these language models. Two major challenges are the production of disinformation and the manifestation of biases, such as racial, gender, and religious biases, in LLMs [43]. In the financial industry, accuracy of information is crucial for making sound financial decisions, and fairness is a fundamental requirement for all financial services. To ensure information accuracy and mitigate hallucination, additional measures like retrieve-augmented generation [25] can be implemented. To address biases, content censoring and output restriction techniques (such as only generating answers from a pre-defined list) can be employed to control the generated content and reduce bias.

LMMs poises potential challenges in terms of regulation and governance. Although LLM offers more interpretability compared to conventional deep learning models by providing reasoning steps or thinking processes for the generated answers when prompted correctly [50] [60], LLM remains a black box and explainability of the content it generates is highly limited.

Addressing these limitations and ensuring the ethical and responsible use of LLMs in finance applications is essential. Continuous research, development of robust evaluation frameworks, and the implementation of appropriate safeguards are vital steps in harnessing the full potential of LLMs while mitigating potential risks.

6 Conclusion

In conclusion, this paper has conducted a timely and practical survey on the emerging application of LLMs for financial AI. We structured the survey around two critical pillars: solutions and adoption guidance.

Under solutions, we reviewed diverse approaches to harnessing LLMs for finance, including leveraging pretrained models, fine-tuning on domain data, and training custom LLMs. Experimental results demonstrate significant performance gains over general purpose LLMs across natural language tasks like sentiment analysis, question answering, and summarization.

To provide adoption guidance, we proposed a structured framework for selecting the optimal LLM strategy based on constraints around data availability, compute resources, and performance needs. The framework aims to balance value and investment by guiding practitioners from low-cost experimentation to rigorous customization.

In summary, this survey synthesized the latest progress in applying LLMs to transform financial AI and provided a practical roadmap for adoption. We hope it serves as a useful reference for researchers and professionals exploring the intersection of LLMs and finance. As datasets and computation improve, finance-specific LLMs represent an exciting path to democratize cutting-edge NLP across the industry.

References

[1] 2023. Auto-GPT: An Autonomous GPT-4 Experiment. https://github.com/Significant-Gravitas/Auto-GPT.

[2] 2023. Chatbots in consumer finance. https://www.consumerfinance.gov/data-research/research-reports/chatbots-in-consumer-finance/chatbots-in-consumer-finance/

[3] Talal Almutiri and Farrukh Nadeem. 2022. Markov models applications in natural language processing: a survey. *Int. J. Inf. Technol. Comput. Sci* 2 (2022), 1–16.

[4] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems* 13 (2000).

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 https://arxiv.org/abs/2005.14165

[7] Harrison Chase. 2022. *LangChain*. https://github.com/hwchase17/langchain

[8] Mu-Yen Chen. 2011. Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches. *Computers & Mathematics with Applications* 62, 12 (2011),

4514–4524. https://doi.org/10.1016/j.camwa.2011.10.030

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs.CL]

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs.CL]

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. arXiv:2210.11416 [cs.LG]

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[14] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.

[15] Bledar Fazlija and Pedro Harder. 2022. Using Financial News Sentiment for Stock Price Direction Prediction. *Mathematics* 10, 13 (2022). https://doi.org/10.3390/math10132156

[16] Peter Foy. 2023. GPT-4 for Financial Statements: Building an AI Analyst. MLQ AI. https://www.mlq.ai/gpt-4-financial-statements-ai-analyst/

[17] Xinyang Geng and Hao Liu. 2023. *OpenLLaMA: An Open Reproduction of LLaMA.* https://github.com/openlm-research/open_llama

[18] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A Survey of Quantization Methods for Efficient Neural Network Inference. arXiv:2103.13630 [cs.CV]

[19] John Goodell, Satish Kumar, Weng Marc Lim, and Debidutta Pattnaik. 2021. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance* 32 (08 2021). https://doi.org/10.1016/j.jbef.2021.100577

[20] Alex Graves. 2014. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850 [cs.NE]

[21] Aaryan Gupta, Vinya Dengre, Hamza Kheruwala, and Manan Shah. 2020. Comprehensive review of text-mining applications in finance. *Journal of Financial Innovation* 6 (11 2020). https://doi.org/10.1186/s40854-020-00205-1

[22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]

[23] Kyoung jae Kim. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55, 1 (2003), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2 Support Vector Machines.

[24] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. 2019. A Study of BFLOAT16 for Deep Learning Training. arXiv:1905.12322 [cs.LG]

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401 [cs.CL]

[26] Yinheng Li. 2023. A Practical Survey on Zero-shot Prompt Design for In-context Learning. International Conference Recent Advances in Natural Language Processing.

[27] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic Evaluation of Language Models. arXiv:2211.09110 [cs.CL]

[28] Dakuan Lu, Hengkui Wu, Jiaqing Liang, Yipei Xu, Qianyu He, Yipeng Geng, Mengkun Han, Yingsi Xin, and Yanghua Xiao. 2023. BBT-Fin: Comprehensive Construction of Chinese Financial Domain Pre-trained Language Model, Corpus and Benchmark. arXiv:2302.09432 [cs.CL]

[29] Cuicui Luo, Desheng Wu, and Dexiang Wu. 2017. A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence* 65 (2017), 465–470. https://doi.org/10.1016/j.engappai.2016.12.002

[30] Akib Mashrur, Wei Luo, Nayyar A. Zaidi, and Antonio Robles-Kelly. 2020. Machine Learning for Financial Risk Management: A Survey. *IEEE Access* 8 (2020), 203203–203223. https://doi.org/10.1109/ACCESS.2020.3036322

[31] Microsoft. 2023. Semantic Kernel. https://github.com/microsoft/semantic-kernel.

[32] Chiara Valentina Misischia, Flora Poecze, and Christine Strauss. 2022. Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science* 201 (2022), 421–428. https://doi.org/10.1016/j.procs.2022.03.055 The 13th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 5th International Conference on Emerging Data and Industry 4.0 (EDI40).

[33] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[34] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]

[35] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. 2020. Deep Learning for Financial Applications : A Survey.

arXiv:2002.05786 [q-fin.ST]

[36] Cynthia Pagliaro, Dhagash Mehta, Han-Tai Shiao, Shaofei Wang, and Luwei Xiong. 2022. Investor Behavior Modeling by Analyzing Financial Advisor Notes: A Machine Learning Perspective. In *Proceedings of the Second ACM International Conference on AI in Finance* (Virtual Event) *(ICAIF '21)*. Association for Computing Machinery, New York, NY, USA, Article 23, 8 pages. https://doi.org/10.1145/3490354.3494388

[37] Igor Radovanovic. 2023. Auto-GPT for finance - an exploratory guide - algotrading101 blog. https://algotrading101.com/learn/auto-gpt-finance-guide/

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG]

[39] Abhimanyu Roy, Jingyi Sun, Robert Mahoney, Loreto Alonzi, Stephen Adams, and Peter Beling. 2018. Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*. 129–134. https://doi.org/10.1109/SIEDS.2018.8374722

[40] Omer Berat Sezer, Murat Ozbayoglu, and Erdogan Dogdu. 2017. A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters. *Procedia Computer Science* 114 (2017), 473–480. https://doi.org/10.1016/j.procs.2017.09.031 Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.

[41] Ashish Shah, Pratik Raj, Pushpam Kumar, Supriya P, and Asha H V. 2020. FinAID, A Financial Advisor Application using AI. , 2282–2286 pages. https://doi.org/10.35940/ijrte.a2951.059120

[42] Hugh Son. 2023. JPMorgan is developing a CHATGPT-like A.I. service that gives investment advice. https://www.cnbc.com/2023/05/25/jpmorgan-develops-ai-investment-advisor.html

[43] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. arXiv:2102.02503 [cs.CL]

[44] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[47] Junhao Wang, Yinheng Li, and Yijie Cao. 2019. Dynamic Portfolio Management with Reinforcement Learning. arXiv:1911.11880 [q-fin.PM]

[48] Yaqing Wang, Quanming Yao, James Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-Shot Learning. arXiv:1904.05046 [cs.LG]

[49] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. arXiv:2206.07682 [cs.CL]

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *CoRR* abs/2201.11903 (2022). arXiv:2201.11903 https://arxiv.org/abs/2201.11903

[51] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2023. Transformers in Time Series: A Survey. arXiv:2202.07125 [cs.LG]

[52] David West. 2000. Neural network credit scoring models. *Computers & Operations Research* 27, 11 (2000), 1131–1152. https://doi.org/10.1016/S0305-0548(99)00149-5

[53] Pedram Babaei William Todt, Ramtin Babaei. 2023. Fin-LLAMA: Efficient Finetuning of Quantized LLMs for Finance. https://github.com/Bavest/fin-llama.

[54] BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin,

Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Lim-
isiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada
Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kas-
ner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan,
Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash
Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini,
Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz
Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis
David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ez-
inwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad,
Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko,
Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia
Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa
Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed
Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott,
Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan
Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Syl-
vain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach
Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Calla-
han, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Ben-
jamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin
Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian
Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel
Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane
Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David
Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu,
Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc
Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias
Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina
Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha See-
lam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner,
Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg,
Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyaw-
ijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Ki-
blawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter,
Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Woj-
ciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan
Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras,
Younes Belkada, and Thomas Wolf. 2023. BLOOM: A 176B-Parameter
Open-Access Multilingual Language Model. arXiv:2211.05100 [cs.CL]

[55] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze,
Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and
Gideon Mann. 2023. BloombergGPT: A Large Language Model for
Finance. arXiv:2303.17564 [cs.LG]

[56] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng,
Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A Large Lan-
guage Model, Instruction Data and Evaluation Benchmark for Finance.
arXiv:2306.05443 [cs.CL]

[57] Frank Xing, Erik Cambria, and Roy Welsch. 2018. Natural language
based financial forecasting: a survey. Artificial Intelligence Review 50
(06 2018). https://doi.org/10.1007/s10462-017-9588-9

[58] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang.
2023. FinGPT: Open-Source Financial Large Language Models.
arXiv:2306.06031 [q-fin.ST]

[59] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang
Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing
the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.
arXiv:2304.13712 [cs.CL]

[60] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L.
Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of
Thoughts: Deliberate Problem Solving with Large Language Models.
arXiv:2305.10601 [cs.CL]

[61] YangMu Yu. 2023. Cornucopia-LLaMA-Fin-Chinese. https://github.
com/jerry1993-tech/Cornucopia-LLaMA-Fin-Chinese.

[62] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming
Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam,
Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu,
Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: An Open
Bilingual Pre-trained Model. In The Eleventh International Conference
on Learning Representations (ICLR). https://openreview.net/forum?id=-
Aw0rrrPUF

[63] Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. 2023. Instruct-
FinGPT: Financial Sentiment Analysis by Instruction Tuning of
General-Purpose Large Language Models. arXiv:2306.12659 [cs.CL]

[64] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen,
Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria
Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel
Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke
Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language
Models. arXiv:2205.01068 [cs.CL]

[65] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong
Bing. 2023. Sentiment Analysis in the Era of Large Language Models:
A Reality Check. arXiv:2305.15005 [cs.CL]

[66] Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2023. XuanYuan 2.0:
A Large Chinese Financial Chat Model with Hundreds of Billions
Parameters. arXiv:2305.12002 [cs.CL]

[67] Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2020. Deep Learn-
ing for Portfolio Optimization. The Journal of Financial Data Science 2,
4 (aug 2020), 8–20. https://doi.org/10.3905/jfds.2020.1.042

[68] Ekaterina Zolotareva. 2021. Aiding Long-Term Investment Decisions
with XGBoost Machine Learning Model. arXiv:2104.09341 [q-fin.CP]