Re(Visiting) Large Language Models in Finance

Eghbal Rahimikia and Felix Drinkall*

July 27, 2025

Abstract

This study evaluates the effectiveness of specialised large language models (LLMs) developed for accounting and finance. Empirical analysis demonstrates that these domain-specific models, despite being nearly 50 times smaller, consistently outperform state-of-the-art general-purpose LLMs in return prediction. By pre-training the models on year-specific financial datasets from 2007 to 2023, the study also mitigates look-ahead bias, a common limitation of general-purpose LLMs. The findings highlight the critical importance of addressing look-ahead bias to ensure reliable results. Extensive robustness checks further validate the superior performance of these models.

Keywords: Natural Language Processing, Large Language Models, Asset Pricing, Return Prediction, Machine Learning.

JEL: G10, G11, G14, C22, C23, C45, C55, C58

*Eghbal Rahimikia (corresponding author) (eghbal.rahimikia@manchester.ac.uk) is at the University of Manchester, Alliance Manchester Business School (AMBS), and Felix Drinkall (felix.drinkall@eng.ox.ac.uk) is at the Oxford-Man Institute of Quantitative Finance and Oxford e-Research Centre, University of Oxford. We are grateful for the valuable feedback and insights received from participants at the following seminars and conferences: Mathematical and Computational Finance Seminar, Mathematical Institute, University of Oxford, 7th Fintech North Conference, University of Manchester, Frontiers in Quantitative Finance Seminar, University of Oxford and Citibank, London, International Finance and Banking Society (IFBS) Conference, Saïd Business School, Oxford, Wolfe Research 8th Annual Quantitative and Macro Investment Conference, London, Royal Statistical Society (RSS), Merseyside Local Group, University of Liverpool Management School, Oxford Human-Algorithm Interaction Workshop, Saïd Business School, Oxford, and International Conference on Empirical Economics, Pennsylvania State University at Altoona, USA. The authors would like to acknowledge the assistance provided by Research IT and the use of the Computational Shared Facility at The University of Manchester. This work also made use of the facilities of the N8 Centre of Excellence in Computationally Intensive Research (N8 CIR), funded by the N8 research partnership and EPSRC (Grant No. EP/T022167/1). The Centre is coordinated by the Universities of Durham, Manchester, and York. We further acknowledge the financial support provided by Digital Futures at the University of Manchester, the Alan Turing Institute, and AMBS. We thank AMBS and the Oxford-Man Institute of Quantitative Finance for providing access to their internal HPC servers for this study.

1 Introduction

In recent years, the field of natural language processing (NLP) has undergone significant transformation, driven primarily by rapid advancements and the widespread adoption of large language models (LLMs). These models have revolutionised NLP, shifting the focus of research in accounting and finance from traditional, simpler techniques to more sophisticated approaches. Despite the initial progress made in applying LLMs to accounting and finance, a significant gap remains in the development—or more precisely, the pre-training—of specialised LLMs tailored to the unique requirements of these domains. Researchers in accounting and finance face specific challenges and considerations when utilising LLMs in their work, and existing models often fall short of addressing these needs. Consequently, there is a critical need for models that are specifically developed and optimised to tackle the unique complexities and demands of these disciplines.

Textual analysis in accounting and finance is not a novel concept but has deep roots in early academic research. Historically, scholars in the field recognised the potential of textual data to provide insights beyond traditional quantitative measures. Early studies, such as Antweiler and Frank (2004), demonstrated that internet message boards could serve as valuable sources of information for forecasting market volatility. Similarly, Tetlock (2007) analysed Wall Street Journal columns and used textual content to extract sentiment indicators that could predict stock market movements. Loughran and McDonald (2011) further contributed to the field by developing a specialised dictionary designed specifically to capture the tone of financial reports. The Loughran-McDonald dictionary (LMD) has since become a cornerstone in financial textual analysis, enabling researchers to assess the sentiment of financial documents with greater accuracy. These foundational studies laid the groundwork for the broader application of textual analysis in accounting and finance, underscoring its relevance and utility long before the advent of LLMs that now dominate the field of NLP. Loughran and McDonald (2016) and Gentzkow et al. (2019) provided comprehensive reviews of textual analysis within these domains.

In recent years, a new wave of models has significantly advanced the field of NLP. The starting point for this revolution was the introduction of word embeddings by Mikolov et al. (2013), which allowed words to be represented as vectors in a continuous space, capturing semantic relationships between them. Building on this, the introduction of the attention mechanism by Vaswani (2017) paved the way for further breakthroughs. Notably, the Bidirectional Encoder Representations from Transformers (BERT) model was introduced by Devlin (2018), marking a substantial leap forward by enabling the context of a word to be understood based on its surrounding words in both directions.

The advancements continued with the introduction of the Robustly Optimised BERT Pre-training Approach (RoBERTa) by Liu et al. (2019), which further improved the performance of the BERT model. This continued with the introduction of even more sophisticated models, each pushing the boundaries of what is possible in textual analysis. One of the most notable developments in recent years has been the emergence of the Generative Pre-trained Transformer (GPT) model family by OpenAI, with commercial GPT-3 and GPT-4 being particularly influential. Introduced by Radford et al. (2019) and Brown (2020), respectively, these models are designed to generate human-like text by predicting the next word based on the context provided by the preceding text. These LLMs leverage massive datasets and billions of parameters, enabling them to produce text that is coherent and contextually relevant across a wide range of topics. Among open-source LLMs, the Large Language Model Meta AI (LLaMA), including LLaMA 1 (Touvron et al., 2023), LLaMA 2 (Touvron et al., 2023), and LLaMA 3 (Dubey et al., 2024), has garnered significant attention for its contributions to the field. Developed by Meta AI, LLaMA is acknowledged for significantly enhancing the accessibility of LLMs, marking a critical advancement in the open-source LLM landscape.

The integration of advanced models has become a trend in accounting and finance, where these models are increasingly incorporated into various applications. For instance, Rahimikia et al. (2021) developed a financial word embedding derived from 15 years of business news archives, which was used to forecast realised volatility by directly utilising text as input. Similarly, the introduction of FinBERT by Huang et al. (2023) marked a key development in the pre-training of specialised language models, specifically BERT, tailored to accounting and financial texts. Their findings demonstrate that this specialised model outperforms general-purpose LLMs in financial tasks. Moreover, focusing on the application of LLMs, Chen et al. (2022) illustrated that models like LLaMA significantly improve the prediction of stock returns through the analysis of news stories. This model surpasses other models by effectively capturing the context and nuances within the text, thereby uncovering market inefficiencies and offering substantial predictive power across global markets. In another study, Kim et al. (2024a) investigated the ability of GPT-4 to perform financial statement analysis, finding that it can predict future earnings with accuracy comparable to, and sometimes surpassing, that of human analysts. Furthermore, Kim et al. (2024b) explored the effectiveness of generative artificial intelligence (AI) tools like ChatGPT in summarising complex corporate disclosures. The study found that these AI-generated summaries, while significantly shorter, enhance the information content, particularly by amplifying the sentiment of the original documents. Text mining models in accounting and finance, outside the realm of LLMs, also exist, primarily focusing on establishing statistical foundations for textual analysis. Among these contributions, Ke et al. (2019) proposed a supervised learning framework that generated a sentiment score explicitly designed for forecasting financial returns. Furthermore, Zhou et al. (2024) introduced the factor-augmented regularised model for prediction (FarmPredict), which allows the model to directly learn financial return patterns from Chinese news data.

One significant challenge in applying LLMs to accounting and finance research is the potential for look-ahead bias or information leakage. This issue arises because LLMs are pre-trained on vast amounts of textual data spanning multiple time periods. When applied to historical data in accounting and finance, these models may inadvertently incorporate information unavailable at the time of the original data, leading to biased or misleading results. Such retrospective applications risk producing analyses that are not only inaccurate but also fundamentally flawed, as they may reflect insights apparent only with the benefit of hindsight. Sarkar and Vafa (2024) explored temporal look-ahead bias in pre-trained language models, finding evidence of this bias in predicting corporate risks and election outcomes. They suggest mitigating the issue by using models pre-trained exclusively on data from before the out-of-sample period. Lopez-Lira and Tang (2023) and Kim et al. (2024a) addressed this concern by testing the robustness of their results using out-of-sample data drawn from periods outside the pre-training window of the LLM. Similarly, Chen et al. (2023) tested robustness by comparing their findings with another LLM pre-trained prior to their data, ensuring that the observed results were not influenced by potential access to future information. While these methods represent significant progress, they fall short of conclusively demonstrating robustness unless the same LLM is explicitly tested with data fully outside its pre-training period. Furthermore, most LLMs are pre-trained on extensive datasets that include very recent data—sometimes up to recent months—leaving limited data available for rigorous out-of-sample testing. Additionally, a new trend in LLMs involves the use of retrieval-augmented generation (RAG), where real-time retrieval of external data introduces an added layer of complexity.

The primary contribution of this study lies in evaluating the effectiveness of developing specialised LLMs tailored for accounting and finance. Specifically, we assess the effectiveness of these models, referred to as FinText[1], across various scenarios. The models are pre-trained from scratch and include a base version with approximately 125 million parameters and a smaller variant with 51 million parameters. FinText models are meticulously tailored for the accounting and finance domains, drawing on specialised, diverse, and high-quality textual data sources spanning from 2007 to 2023, with one model pre-trained for each year within this period.[2] Given the historical nature of the data, these

[1] For more information, visit the FinText portal.

[2] Although the data is specifically tailored for these domains, the extensive and varied nature of the textual data used in pre-training also makes these models applicable to related fields such as economics, business, marketing, and management.

models are rigorously designed to eliminate look-ahead bias, thereby ensuring that the results obtained are not compromised by any potential information leakage. The models are intentionally designed to be smaller in scale compared to state-of-the-art LLMs to ensure they can be used effectively without requiring high-end graphical processing units (GPUs). We employ a diverse array of textual datasets, including news articles, regulatory filings, intellectual property (IP) records, and key corporate information. Our sources also encompass speeches from the European Central Bank (ECB) and the Federal Reserve (FED), transcripts of corporate events, and data on board members. Furthermore, Wikipedia is incorporated as a general knowledge resource.

We subsequently assess the performance of these models in comparison to the LLaMA models, a widely recognised open-source set of language models ranging from 7 to 8 billion parameters. Following the same framework as Ke et al. (2019), Chen et al. (2022), and Zhou et al. (2024), this evaluation is conducted by constructing a zero-net-investment portfolio, utilising news stories as the basis for analysis. In line with the methodology introduced by Peters et al. (2018), we employ a feature extraction approach, commonly referred to as probing. This technique involves directly utilising the pre-trained model to generate features associated with text data. These features are then applied within a logistic model to estimate a model for predicting return direction. This approach has also been adopted in other studies, such as Chen et al. (2022). Our findings show that FinText consistently outperforms LLaMA 1, 2, 3, FarmPredict, FinBERT, and LMD, achieving a Sharpe ratio of 3.45 in an equal-weighted portfolio. This superior performance is noteworthy, given that the LLaMA models have approximately 50 times more parameters and may also be influenced by potential look-ahead bias. Additionally, we conduct an evaluation using the smaller version of the FinText LLMs. While the small model exhibits lower performance compared to the base model, it still demonstrates strong portfolio performance, particularly given its reduced size, achieving a Sharpe ratio of 2.75. This finding suggests that while model complexity is an important factor in the performance of LLMs in asset pricing, it is not the sole determinant. Our results indicate that smaller models, when pre-trained on high-quality, domain-specific textual data, can achieve performance levels that rival or even surpass those of state-of-the-art larger models.

We assess the robustness of our findings by conducting a variety of robustness tests. First, even after including transaction costs, the superior performance of FinText remains valid compared to other models. Next, after adjusting for the Capital Asset Pricing Model (CAPM) and the Fama-French factor models, we consistently observe that the average return generated by our trading strategy is effectively equivalent to the alpha. This result holds across all tests, underscoring the robustness of the strategy's performance independent of traditional risk factors. We extend our analysis by pre-training two additional series of FinText models—utilising both the base and small versions—for an extended pre-training period. Despite the longer pre-training, the base model demonstrates a decline in portfolio performance. For the small model, although there is a slight improvement, this advantage dissipates once transaction costs are taken into account. We also conduct additional analyses of portfolio performance by examining different trading sizes and yearly models. Our findings consistently demonstrate the strong performance of the FinText models across these varying conditions.

We further extend our analysis to examine the impact of look-ahead bias by testing the models at various levels, ranging from the linguistic level to the portfolio level. Our findings indicate evidence of look-ahead bias at the linguistic and classification levels. However, most of these traces diminish at the portfolio level, which can be attributed to the post-processing steps applied in the transition to portfolio performance results. These findings underscore the importance of conducting comprehensive and nuanced analyses when employing language models, particularly larger models that may encapsulate more extensive information than those utilised in this study, which could significantly influence the results due to look-ahead bias.

Moving to the next analysis, a crucial aspect of effectively utilising LLMs lies in their fine-tuning for specialised downstream tasks. Therefore, we also extend our investigation by specifically focusing on the direction of return prediction as a downstream task. We assess model performance by annually fine-tuning FinText models to predict the direction of returns. Our findings indicate that, after testing the models across various batch sizes—each impacting the computational demands of fine-tuning—all models under-perform compared to the simpler approach previously employed. Finally, due to their relatively smaller sizes, FinText models facilitate the application of explainable AI (XAI) methods. By leveraging XAI, we identify the specific groups of words that influence long and short portfolios.

The rest of this paper is organised as follows: Section 2 delves into the theoretical foundations of LLMs. Section 3 details the FinText model, covering the LLM setup and pre-training corpus. Section 4 presents the pre-processing steps, estimation setup, and portfolio setup for return prediction. Section 5 reports the results of the empirical evaluation, comparing the performance of the FinText model against other leading models. Finally, Section 6 summarises our findings and discusses the implications for future research.

2 Large Language Models (LLMs)

LLMs represent a subset of AI that can comprehend, generate, or manipulate human language, depending on their specific design and capabilities, using extensive textual data. These models are constructed using deep learning methodologies, particularly neural networks composed of many layers and usually millions or billions of parameters, which allow them to discern complex linguistic patterns and structures. The evolution of LLMs is deeply rooted in the broader discipline of NLP. However, their development significantly accelerated following the introduction of the Transformer architecture by Vaswani (2017). The attention mechanism of the Transformer model, particularly the self-attention component, facilitated more efficient training processes and improved the management of long-range dependencies within text, catalysing a swift advancement in LLM capabilities. Prominent models such as BERT and RoBERTa are based on this architecture. These advancements, alongside the emergence of OpenAI's GPT family, have illustrated the potential of LLMs to execute a diverse array of language-related tasks with minimal task-specific adjustments, thereby promoting widespread adoption and continuous innovation within the field.

2.1 Model and Tokenisation

The BERT model represented a significant advancement in the development of LLMs, primarily due to its bidirectional approach to processing text. BERT processes entire sequences of words simultaneously, allowing it to better capture the context of each word based on its surrounding words. This bidirectional nature enables BERT to excel in tasks that require a deep understanding of context, such as question answering and sentiment analysis. BERT's architecture comprises multiple layers of Transformer encoders, and its pre-training involves two key tasks: masked language modelling (MLM) and next sentence prediction (NSP). In MLM, a percentage of the input words are masked, and the model is trained to predict these masked words, which helps it develop a deep understanding of language structure. NSP, on the other hand, helps the model understand the relationship between sentences, enhancing its effectiveness in tasks that require sentence-level comprehension.

Building on the foundation laid by BERT, RoBERTa enhances the original model by refining the pre-training process. RoBERTa discards the NSP task, which was found to be less effective, and instead focuses on pre-training with more data, larger batches, longer pre-training periods, and dynamic masking, where the masked tokens vary across pre-training epochs. These adjustments result in a model that delivers superior performance on various downstream tasks by providing more robust text representations. RoBERTa's advancements underscore the importance of both model architecture and pre-training strategies in enhancing language model performance, demonstrating the potential to refine existing models for superior results without fundamentally altering their underlying structure.

As the field of LLMs continues to advance, new models such as LLaMA have emerged, designed to deliver high performance while prioritising open-source access. These models are built upon a streamlined architecture that reduces computational costs while maintaining high accuracy. Additionally, they excel in causal language modelling (CLM), which is essential for tasks like text generation, where the model predicts the next word in a sequence based on the context of previous words. This CLM capability enables the generation of coherent and contextually relevant text, making these models suitable for various applications, including chatbots, content creation, and real-time language translation. Furthermore, these models strike a balance between size and efficiency, offering configurations that allow users to choose the model size best suited to their resource constraints and application requirements. However, even the smaller models still require GPUs for use.

Tokenisation is a fundamental step in the pre-processing pipeline of LLMs. It involves breaking down text into smaller units, known as tokens, which can be words, subwords, or characters. These tokens are subsequently processed by the model. Different models employ various tokenisation strategies, each with distinct advantages and implications for model performance. For instance, BERT utilises WordPiece tokenisation, a subword-based method that segments words into smaller, more manageable units if they are not present in the model’s vocabulary. This approach enables BERT to handle out-of-vocabulary words effectively by representing them as a combination of known subwords. WordPiece tokenisation is particularly advantageous for languages with rich morphology and for managing variations in word forms, thereby enhancing the model’s ability to generalise across different contexts. In contrast, RoBERTa employs byte-pair encoding (BPE) for tokenisation, a technique that iteratively merges the most frequent pairs of bytes or characters into larger subword units. This method allows RoBERTa to efficiently manage extensive vocabularies and generate more detailed text representations by decomposing rare or unfamiliar words into recognisable subword components. The flexibility of BPE is particularly beneficial for pre-training on large and diverse datasets, as it adapts to a wide range of linguistic subtleties. Notably, BPE tokenisation is also utilised in the LLaMA models.

Given the prevalent use of BPE tokenisers in LLMs, it is essential to provide a practical example of their operation. Consider the sentence: ‘The company’s market capitalisation exceeded expectations in Q2, driven by strong revenue growth.’ When processed by a BPE tokeniser, the sentence is decomposed into subword units such as ‘comp’, ‘any’, ‘’s’, ‘capital’, and ‘isation’. This approach enables the model to effectively capture the complete semantic meaning of specialised financial terms like ‘capitalisation’ while maintaining adaptability for variations or previously unseen words. The resulting tokenised sequence, which includes [‘The’, ‘company’, ‘’s’, ‘market’, ‘capital’, ‘isation’, ‘exceeded’, ‘expectations’, ‘in’, ‘Q2’, ‘,’, ‘driven’, ‘by’, ‘strong’, ‘revenue’, ‘growth’], illustrates how BPE strikes a balance between precision in understanding financial language and the robustness necessary for accurate modelling.

2.2 Pre-Training

During pre-training, models are exposed to vast amounts of textual data to learn the underlying structures and patterns of language. The model is typically tasked with predicting masked tokens within a sentence (as in BERT's or RoBERTa's MLM) or predicting the next token in a sequence (as in LLaMA's CLM). This process enables the model to develop a deep understanding of grammar, context, semantics, and even some aspects of world knowledge, all without any specific task-based supervision. The pre-training phase is computationally demanding, requiring extensive textual datasets and substantial computational power. As a result, most existing research primarily focuses on utilising pre-trained models. However, this investment can yield a language model that is more specialised and effective for the targeted domain compared to general-purpose LLMs.

The success of pre-training LLMs hinges on the sophisticated tuning of several hyperparameters and the careful selection of pre-training data. Critical hyperparameters, such as the learning rate—responsible for controlling the magnitude of adjustments made to the model's parameters in response to the estimated error during weight updates—play a pivotal role in optimising the model's performance. Similarly, the batch size, which determines the number of pre-training examples used to estimate the error gradient, significantly influences the efficiency and effectiveness of the pre-training process. Moreover, the pre-training data must be sufficiently diverse and representative to ensure that the model generalises well across various subdomains. For instance, pre-training on a dataset rich in accounting and financial texts could produce a model more adept at handling accounting and financial tasks. However, such domain-specific pre-training is often constrained by the availability of relevant data.

The most widely adopted open-source LLM pre-trained specifically for accounting and financial applications is FinBERT, as introduced by Huang et al. (2023). FinBERT is based on the BERT architecture and has been pre-trained on three primary datasets: corporate filings (including 10-K and 10-Q reports from Russell 3000 firms between 1994 and 2019), financial analyst reports (covering S&P 500 firms from 2003 to 2012), and earnings conference call transcripts (from public firms between 2004 and 2019). These datasets collectively comprise 4.9 billion tokens. Despite being the first model of its kind, FinBERT has certain limitations, particularly the inconsistency in the timeframes of the data used, as each dataset covers a distinct range of years. Moreover, the number of tokens utilised during pre-training and the diversity of the data are not as optimised as those in state-of-the-art LLMs. Recent studies, such as those by Chen et al. (2022), have raised questions regarding the performance of this specialised LLM compared to general-purpose LLMs in portfolio construction and trading.

Moving to general-purpose LLMs, LLaMA is one of the most well-known open-source models. These models are frequently reported to exhibit performance comparable to that of GPT models. LLaMA 1 was trained on approximately 1.4 trillion tokens sourced from datasets such as Common Crawl, C4, GitHub, BooksCorpus, and English Wikipedia. The text data predominantly covers content up to the year 2021 and is primarily focused on English. LLaMA 2 expanded the dataset to over 2 trillion tokens, incorporating data up to 2022. It also employed more aggressive filtering of low-quality content, included a broader range of multilingual data, and added specialised texts from scientific and academic sources, including arXiv. LLaMA 3 features a dataset exceeding 3 trillion tokens, with data extending through 2023.

2.3 Fine-Tuning

Fine-tuning, which follows the pre-training phase, is a crucial step in adapting LLMs to specific tasks. During fine-tuning, the pre-trained model undergoes additional training on a smaller, more specialised dataset that is closely aligned with the intended application. This process enables the LLM to refine its understanding and improve its performance within the target domain, leveraging the linguistic comprehension developed during pre-training. Fine-tuning typically involves adjusting the model's parameters with a lower learning rate compared to pre-training, to prevent significant shifts in the knowledge already acquired while still allowing the model to adapt to new data patterns. While pre-training requires extensive textual datasets and substantial computational power, fine-tuning is considerably less resource-intensive.

To develop an effective fine-tuned model, it must be trained on high-quality samples that have been meticulously labelled with their corresponding tags, a process typically carried out by human annotators. The effectiveness of fine-tuning is significantly influenced by several factors, including the size and quality of the fine-tuning dataset, the choice of hyperparameters, and the similarity between the data used for pre-training and the fine-tuning task. This underscores the importance of the pre-trained model and the knowledge it encapsulates, as these directly impact the quality and effectiveness of the fine-tuned model. In the fields of accounting and finance, various fine-tuning tasks are of considerable interest. One common task is sentiment analysis, where LLMs are fine-tuned to accurately identify and extract sentiment from text. The range of potential fine-tuning tasks is broad, with the essential requirement being access to high-quality, labelled samples.

3 FinText: Pre-Trained Financial LLMs

This section presents our series of specialised pre-trained LLMs, detailing their architectures, the data used for pre-training, and the methodological considerations at each stage of development. We emphasise the importance of transparency in these processes, aiming to equip researchers with the knowledge needed to develop specialised models in this field. All FinText models utilised in this study were pre-trained from scratch, employing the setup in Subsection 3.1 and the pre-training corpus described in Subsection 3.2.

3.1 LLM Setup

The current trend in the development of LLMs emphasises the incorporation of more diverse textual data, including the expansion of multilingual capabilities, the integration of code, and the inclusion of other text forms. These approaches aim to enhance the models' ability to generalise across different linguistic contexts. Alongside these efforts, increasing the size of LLMs has become a priority, as larger models tend to capture a broader spectrum of knowledge. Researchers are also continually refining the architectures of these models to improve their performance across various scenarios. As a result, LLMs undergo frequent updates, leading to significant changes over time, which can cause studies conducted with different versions of the same model to yield substantially different results. This issue is particularly prevalent with commercial LLMs, such as OpenAI's GPT family, where access to previous versions is often restricted and eventually discontinued.

In contrast, researchers in fields such as accounting and finance require stable, standardised models that serve as reliable foundations for their work. Time series analysis is also common in these disciplines, and using the latest version of an LLM can introduce significant risks, such as look-ahead bias and potential information leakage. These risks arise because many LLMs are pre-trained on vast datasets that include both recent and historical information about financial markets, companies, political events, and more. Applying such models to historical data can, therefore, produce misleading results. Sarkar and Vafa (2024) highlighted this critical issue, providing evidence of look-ahead bias in financial tasks. The authors conducted direct tests for look-ahead bias in pre-trained language models by assuming certain events were unpredictable within a given information set, analysing instances where the models predicted these events accurately despite lacking prior context. They recommended using models trained exclusively on data from before the out-of-sample period to mitigate this problem. However, the rapid evolution of LLM development—with many models being developed and updated continuously over the past two years—often means that access to previous versions is un-

Table 1: FinText LLM Configuration Details

Hyperparameter	**Base**	**Base (further)**	**Small**	**Small (further)**
Number of Layers	12	12	8	8
Hidden Size	768	768	512	512
Number of Attention Heads	12	12	8	8
Number of Parameters (Millions)	124.65	124.65	51.48	51.48
Input Sequence Length	512	512	512	512
Vocabulary Size	50265	50265	50265	50265
Optimiser	AdamW	AdamW	AdamW	AdamW
ϵ	1e-6	1e-6	1e-6	1e-6
β_1	0.90	0.90	0.90	0.90
β_2	0.98	0.98	0.98	0.98
Weight Decay	0.01	0.01	0.01	0.01
Learning Rate	6e-4	3e-4	6e-4	3e-4
Learning Rate Decay	Linear	Linear	Linear	Linear
Warmup Ratio	0.05	0.05	0.05	0.05
Dropout	0.1	0.1	0.1	0.1
Attention Dropout	0.1	0.1	0.1	0.1
Activation Function	GELU	GELU	GELU	GELU
Tensor Type	BF16	BF16	BF16	BF16
Total Batch Size	7728	7728	7728	7728
Epochs	5	10(5+5)	5	10(5+5)

Note: This table presents an overview of the model configurations utilised in various versions of the FinText models.

available. Consequently, researchers face the challenge of either accepting the risk of look-ahead bias or restricting their analysis to a very short timeframe, which is impractical for many studies.

To address these complexities, we pre-trained a distinct LLM for each year from 2007 to 2023, with each model trained exclusively on data available up until the last day of its respective year. We selected RoBERTa as the LLM architecture for several key reasons. Firstly, models like RoBERTa, trained with MLM, excel at natural language understanding tasks such as classification, which are generally more important for accounting and finance. In contrast, CLM models are designed primarily for text generation. Secondly, textual data in accounting and finance is often quite short, sometimes consisting of only a single sentence. RoBERTa is particularly well-suited for this context as it omits the NSP task, which requires more than one sentence. Finally, Liu et al. (2019) demonstrated that RoBERTa outperforms BERT, achieving significantly better performance and faster convergence during pre-training.

Table 1 provides an overview of the model configurations utilised across various versions of the FinText LLMs. With the exception of the number of epochs, total batch size, and optimiser, the base and base with further pre-training models align with the hyperparameters proposed by Liu et al. (2019). The small and small with further pre-training models are designed to replicate the dimensions of the medium-sized BERT model described by Turc et al. (2019). The original RoBERTa model, as proposed by Liu et al. (2019), was pre-trained over 40 epochs, made possible by substantial computational resources, including access to thousands of GPUs for pre-training a single LLM. In contrast, each model type listed in Table 1 includes 17 distinct pre-trained models, totalling 68 models. This marks the largest release of specialised LLMs to date in terms of the number of models developed. We also introduce a series to explore the impact of extended pre-training. These models are pre-trained for an additional 5 epochs, extending the total pre-training epochs to 10. During this extended pre-training, a reduced learning rate is employed to ensure the stability of the pre-trained parameters. Also, to ensure comparability across all models, the pre-training process was initiated using a controlled random number generator (RNG) set to the same state for all models, enhancing the reproducibility of the results.

In this study, we introduce two different model sizes. The base model consists of 124.65 million parameters, while the small model contains 51.48 million parameters. Compared to state-of-the-art LLMs, these models are relatively lightweight and compact, making them easier to use and implement. We employ stage 1 zero redundancy optimiser (ZeRO) as implemented in DeepSpeed (Rajbhandari et al., 2020), a technique that significantly reduces memory footprint and enhances the scalability of the pre-training process. Additionally, we utilise Bfloat16 (BF16) precision, which improves computational efficiency and reduces memory usage. Similar to the optimisation approach used in BERT, RoBERTa was optimised using adaptive moment estimation (Adam) (Kingma and Ba, 2014). To further enhance optimisation efficiency, we employ AdamW (Loshchilov and Hutter, 2017), a variant of the Adam optimiser that decouples weight decay from the gradient update process.

Additionally, we provide further details regarding the model pre-training in the Appendix. Figures Figure A1 and Figure A2 display the training and evaluation loss for all models across the full range of pre-training steps. The statistics are derived from the performance outcomes of 17 FinText models spanning the period from 2007 to 2023. The performance improvement is evident across all figures, although, as anticipated, the magnitude of improvement is less pronounced for the models subjected to further pre-training in Figure A2. The time spent for pre-training per model type per year is also reported in Table A1. The pre-training process was performed utilising various GPU configurations, including setups with up to four NVIDIA A100 GPUs (80GB) and two NVIDIA Grace-Hopper GPUs (GH200 480). The average time reported for pre-training the base and small models was approximately 39.96 and 22.67 hours, respectively, extending to 80.33 and 46.67 hours for further pre-training. The cumulative time spent on all models totalled 1365.66 hours for the base models and 793.36 hours for the small models. We also provided a report on the electricity usage and its cost (in pounds) for pre-training the base and small models over time, as detailed in Table A2. Since the costs for further pre-training are comparable to those of the main models, we have omitted those details from this table. The entire electricity used is fully traceable and sourced exclusively from renewable energy. In addition to the electricity consumption and cost data, we also report the corresponding CO_2 emissions, calculated using the greenhouse gas reporting conversion factors provided by the UK Department for Business, Energy and Industrial Strategy.[3]

3.2 Pre-Training Corpus

One of the critical factors influencing the quality of LLMs is the corpus used during pre-training. In NLP, a corpus refers to a large collection of textual data utilised for pre-training language models. We considered several key factors in constructing our corpus: the quality of the data is paramount; the data should encompass comprehensive and valuable knowledge not only in accounting and finance but also in related fields; the dataset should be substantial, either in terms of the number of samples and tokens or as a significant source of information within these domains; and the source should permit the creation of historical pre-training datasets.

We utilise the extensive dataset provided by Capital IQ's Key Developments to incorporate essential market information into our financial models. Accessible via WRDS[4], this dataset offers concise summaries of significant news events and other critical information that may impact the market valuation of securities. The dataset covers a wide range of events, including, but not limited to, executive leadership changes, mergers and acquisitions rumours, updates in corporate guidance, delayed regulatory filings, and inquiries from the Securities and Exchange Commission (SEC). This dataset encompasses information from 165 countries and over 800,000 companies globally, making it an invaluable textual resource for developing specialised LLMs. We focus on the 'situation' data, as it provides more comprehensive information compared to the 'headline' data.

We use the Dow Jones (DJ) Newswires data source as our primary source for news stories. DJ Newswires is a financial news service that provides news to financial professionals worldwide, covering a broad spectrum of financial instruments, including stocks, bonds, commodities, and currencies. Following the data cleaning methodology outlined in Rahimikia et al. (2021), we conducted a comprehensive cleaning process on the textual dataset. Each news article in the dataset includes both a headline and body text; while the headline is consistently present, the body text is occasionally absent. To maximise the utilisation of available information, we employed a combination of headline and body.

[3] It is important to note that while high-end GPUs are crucial for pre-training LLMs, specialised servers with fast CPUs and terabytes of RAM are equally essential for data pre-processing, which can take considerable time even on advanced systems. Though this study does not report the associated costs of utilising these servers, future pre-training efforts should take these expenses into account for proper planning.

[4] Wharton Research Data Services.

IP plays a crucial role in accounting and finance research. The ORBIS IP data source is a comprehensive resource for accessing global patent data, enabling users to analyse patent information and track IP trends across various industries. In this study, we focus exclusively on patents registered in the United States and the United Kingdom. To refine our dataset, we select only patents valued at over $500,000, ensuring that our models incorporate major patents. It is important to note that the valuation process entails a delay, which results in a limited number of evaluated samples for the years 2022 and 2023. Additionally, rather than incorporating the full content of these patents, we use only their abstracts, as the technical complexity of many patents may not contribute substantial value to our models.

Filing data constitutes an important component of the pre-training corpus for our models. The Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system, maintained by the SEC, serves as a critical resource by offering public access to financial disclosures from companies mandated to report to the SEC. We utilise 10-Q and 10-K reports, incorporating filings from companies listed on the Russell 3000 index for extraction. The index undergoes annual changes; therefore, we monitor it on a yearly basis and update the list of companies accordingly. All filings undergo meticulous preprocessing and cleaning to ensure that the textual quality meets the standards required for effective model pre-training. The preprocessing steps are detailed in the first part of Appendix B.

The Capital IQ Transcripts data source, available via WRDS, provides comprehensive access to historical transcripts of corporate earnings calls, M&A announcements, and other significant financial events, covering over 10,600 companies globally since 2005. The data is granular, distinguishing individual speakers within each conference call. We combined the speaker-level data to create a single consolidated dataset for each conference call. Each transcript is available in multiple formats. Therefore, we refined the dataset by filtering and retaining only the verified and proofread versions for our corpus. Additionally, ECB and FED speeches were manually downloaded from the official portals of the ECB and FED to supplement this source. Although this dataset is relatively small compared to others used in this study, its inclusion is crucial to ensure comprehensive coverage of relevant economic information.

While textual sources in accounting and finance are critical for developing our specialised LLMs, incorporating general knowledge is also essential. To achieve this broader context, we utilised the latest Wikipedia data dump and analysed the historical edit timelines of each page to construct a comprehensive historical snapshot at the end of each year. Specifically, we focused on level four

Wikipedia pages, which are part of the 'Vital Articles' project[5]. This project categorises articles according to their significance in providing a general understanding of human knowledge, with level one being the most essential and level five covering a broader array of topics. Level four comprises approximately 10,000 articles that are important but not as central as those in the higher levels. Our analysis monitored the number of pages across different levels and revealed that level four contains a substantial number of articles. Moreover, the content provided by level four articles is sufficiently detailed yet not overly broad, making it particularly suitable for enriching a specialised LLM.

Finally, board information is crucial for a comprehensive understanding of both the current and historical profiles of board members. However, there is no existing comprehensive historical textual dataset that captures this information. To address this gap, we utilised BoardEx data, accessible via WRDS, to systematically generate historical textual records for each board member in the U.S. This process involved the annual conversion of available data into textual format by meticulously tracking and documenting specific changes observed for each individual over time. Additionally, we incorporated various templates for each data component to enhance the diversity of the generated samples. This dataset provides valuable insights, such as age, gender, positions held, educational qualifications, and other relevant attributes. The incorporated steps for this conversion are detailed in the second part of Appendix B.

The starting year for all data sources is set to 1997, except for transcript data, which begins in 2005. Board information and Wikipedia data are updated annually, with each year's data reflecting the most current information available at the time of the update. Other data sources are collected cumulatively over time. For example, the 2009 filing data includes information spanning from 1997 to 2009. To maintain consistency across all models, an identical number of tokens is used during pre-training for each year. Hoffmann et al. (2022) propose that increasing the parameter-to-token ratio beyond 1:20 generally does not lead to substantial performance improvements in LLMs, primarily raising computational costs, though this finding is based on models other than RoBERTa. By integrating all data sources and analysing the token count over time, we set the total number of tokens for all models to 8.422 billion. In terms of data size, the final dataset used across all years totals approximately 1.1 TB, with each yearly model accounting for around 65.5 GB of data.

Table 2 provides a breakdown of the yearly token sizes (in millions) for all textual data sources from 2007 to 2023. The total available token size for each year is given in the last column. The target total token size (8442 million tokens) is presented at the bottom of the table, along with the approximate

[5] More information about this project is available here.

Table 2: Token Sizes Per Year and Sampling Percentages of Textual Data Sources

Year	Key Info	News	IP	Filling	ECB	FED	Transcript	Wikipedia	Board Info	Total
2007	**296.5**	4667.5	**7.3**	4549.3	**4.1**	**3.0**	**40.1**	**5.1**	**44.0**	9616.8
2008	**374.9**	5316.8	**8.9**	5075.0	**4.7**	**3.3**	**130.5**	**8.7**	**49.1**	10971.9
2009	**459.2**	5958.7	**10.5**	5638.6	**5.3**	**3.5**	**235.9**	**16.0**	**54.1**	12381.7
2010	564.9	6637.0	**12.1**	6283.7	**5.8**	**3.8**	**388.9**	**18.6**	**59.3**	13973.9
2011	670.7	7312.9	**13.6**	7101.0	**6.4**	**4.0**	**594.2**	**25.4**	**64.7**	15792.9
2012	775.7	7922.3	**15.0**	8296.5	**6.9**	**4.2**	**802.6**	**31.7**	**70.5**	17925.3
2013	874.5	8385.8	**16.4**	9579.7	**7.4**	**4.4**	**1043.4**	**31.2**	**76.4**	20019.3
2014	970.3	8870.4	**17.5**	10509.7	**7.8**	**4.6**	**1295.2**	**34.8**	**82.5**	21792.8
2015	1072.0	9337.4	**18.5**	11449.6	**8.2**	**4.9**	1542.3	**36.1**	**89.0**	23557.8
2016	1170.3	9819.8	**19.2**	12407.3	**8.5**	**5.1**	1782.9	**37.6**	**95.2**	25345.8
2017	1288.9	10427.2	**20.7**	13491.1	**9.1**	**5.4**	2070.7	**42.0**	**103.0**	27458.2
2018	1404.7	10890.4	**21.0**	14492.0	**9.5**	**5.6**	2379.1	**40.1**	**108.6**	29351.0
2019	1501.1	11361.2	**21.1**	15527.1	**9.9**	**5.9**	2688.4	**41.8**	**113.4**	31269.8
2020	1606.5	11898.3	**21.6**	16730.3	**10.2**	**6.1**	3065.8	**59.0**	**119.5**	33517.3
2021	1909.5	13640.6	**25.1**	20131.3	**12.1**	**7.2**	3956.0	**48.6**	**130.8**	39861.2
2022	2035.0	14396.1	**25.3**	21703.1	**12.6**	**7.4**	4485.7	**61.7**	**133.8**	42860.7
2023	2123.1	14980.0	**25.3**	23034.6	**12.9**	**7.8**	4947.1	**68.5**	**134.0**	45333.2
Share (%)	6	36	1	33	0.25	0.25	18	3	2.5	100
Token size	505	3032	84	2779	21	21	1516	253	211	8422

Note: This table provides a breakdown of the yearly token sizes (in millions) for all textual data sources from 2007 to 2023. The columns represent data sources used for pre-training our LLMs, including Key Information, News, intellectual property (IP), Filing, European Central Bank (ECB) and Federal Reserve (FED) speech, Transcript, Wikipedia, and Board Information. The total token size for each year is given in the last column.

number of tokens selected from each data source per year. The table also includes the percentage representation of each data source utilised within the corpus at the bottom. Additionally, to ensure that certain smaller data sources contributed meaningfully to the corpus, we deliberately allocated a higher share to them, such as IP, Wikipedia, and board information. Although this weighting scheme is somewhat arbitrary, it was essential for achieving a relatively balanced and diverse dataset in the final corpus. Finally, if the total number of tokens for each data source in a given year falls below the desired token size (as indicated in the last row), over-sampling is employed to meet the target size; otherwise, under-sampling is applied. Values highlighted in bold indicate cases where over-sampling was necessary. The yearly breakdown of the sample size is also shown in Table A3.

A crucial step in constructing a historical corpus is ensuring that the over-sampling or under-sampling processes are conducted in a manner that maintains comparability across different models. Without careful consideration, models risk being pre-trained on a disparate mixture of randomly selected samples from the historical data. Under-sampling is employed when the total available tokens exceed the required number of tokens. This process involves selecting a subset of samples such that the overall token count meets the desired threshold, using a weighting method that prioritises more recent data to reflect temporal relevance. This process involves repeatedly sampling the dataset, with replacement, to inflate the token count to the required level, ensuring that the final dataset is representative in terms of both size and diversity. Conversely, over-sampling is applied when there are fewer available tokens than needed. A key consideration here is that recent data samples are often more important than older ones. Therefore, the weighting method employed in the sampling process prioritises more recent samples by assigning weights to each sample based on its age. Let $D = \{D_1, D_2, \ldots, D_n\}$ denote the set of all time differences D_i, where each D_i represents the number of days between the timestamp of sample i and the oldest date in the dataset. For example, in the case of 2010 news data, D is calculated using all news stories from the period spanning 1997 to 2010. The weight w_i for each sample i (where i ranges from 1 to n) is computed using an exponential decay function as follows:

$$w_i = \exp\left(\frac{D_i}{\max(D)}\right) \quad \text{for } i = 1, 2, \ldots, n \tag{1}$$

where $\max(D)$ refers to the maximum value within the set D. To maintain consistency in our sample selection, we track the IDs of the chosen samples for each year. For instance, when sampling news stories for 2010, we start with the samples selected in 2009—which include data from 1997 to 2009—and then incorporate all samples from 2010 into this existing set. This method allows us to control the composition of the corpus, ensuring that the models developed over time are not pre-trained on completely randomised samples from different time periods. Additionally, consistent with the approach outlined in Subsection 3.1, the same RNG is employed for all sampling processes.

Another critical step involves the method used to tokenise the samples, as this directly influences the quality of the resulting models. To prevent truncation from occurring mid-sentence and to preserve the integrity of the data, we perform sentence segmentation prior to tokenisation. This approach enables us to split the samples into individual sentences, which are then tokenised separately using BPE. After tokenisation, the sentences are reassembled to ensure that the input length remains within the 512-token limit, which is the maximum number of tokens the RoBERTa model supports. Although this process is computationally intensive, it is crucial for enhancing the quality of the samples used for pre-training. The vocabulary size, consistent with that of the RoBERTa model, is set at 50,265 tokens.

4 Return Prediction

We evaluate the effectiveness of pre-trained models in predicting the direction of returns. The focus on directional prediction is motivated by its potential to serve as a foundation for future fine-tuning of these LLMs into a classification framework, as discussed later in this section.

Table 3: Data Preprocessing Breakdown

	Untreated	Tag and Ticker Availability	Duplicate	Redundancy	Return Availability
Train Data	20,594,097	2,058,543	1,668,211	1,377,285	1,358,130
Test Data	20,594,097	2,058,543	1,668,211	1,377,285	1,319,016

Note: This table presents a summary of the data preprocessing steps applied to both the train and test datasets spanning the period from 2013 to 2023.

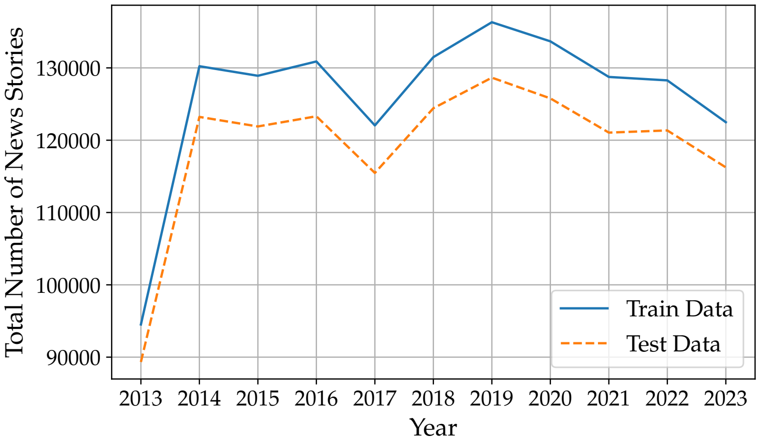
4.1 News Data and Pre-Processing

We utilise the DJ Newswires as the source of news for trading. This data classifies news stories using three distinct tags based on their relevance to the companies referenced. A news story is labelled as 'significant' when it contains breaking news of substantial importance to a specific company. If a story pertains to a company but is not considered highly significant, it is categorised under the 'about' tag. For cases where a company is only briefly mentioned and is not the central focus of the story, the 'mentioned' tag is applied. For our analysis, we specifically use news stories tagged as 'significant' to ensure that the information is directly relevant to the companies in question. Consequently, our data is limited to the period beginning on February 11, 2013, due to the unavailability of the 'significant' tag before this date. To maximise the number of news stories available, we incorporate both the headlines and the body of news stories as inputs for the models.

In accordance with the methodology outlined by Tetlock (2007), we exclude news articles published between 9:00 AM and 9:30 AM EST from the test data. This exclusion is necessary because we require trades to occur at the market open, with a minimum delay of half an hour. However, all available news stories are included in the train data. Specifically, for the train data, we utilise news stories published from 9:00 AM on day t to 9:30 AM on day $t+1$ to predict the direction of returns for day $t+1$. In contrast, the test data is restricted to news stories published from 9:30 AM on day t to 9:00 AM on day $t+1$. The returns are aligned with this framework and are computed on an open-to-open basis. The stock data used in this study is obtained from CRSP.

Table 3 provides a breakdown of the data preprocessing steps for both the train and test datasets. The 'Untreated' column reports the number of samples before data preprocessing. The 'Tag and Ticker Availability' step involves filtering out news stories that lack an associated ticker in the 'significant' tag or include multiple tickers within this tag. We also track companies that have multiple tickers traded in the market. Moreover, only news stories with tickers that match entries in the CRSP database are retained. In the 'Duplicate' step, only the earliest instance of a news story—corresponding to the first time it is released to the market—is preserved, while any subsequent duplicates are removed from the dataset. The 'Redundancy' step is applied to eliminate similar news stories that are published within

Figure 1: Yearly Trend in News Stories Volume: Train vs. Test Data



Note: This figure shows the yearly trend in the total number of news stories for both the train data (solid line) and the test data (dashed line) after applying preprocessing steps.

a short temporal window. Following Chen et al. (2022), this process calculates the cosine similarity between news stories published within overlapping windows of five consecutive business days. News stories exhibiting a cosine similarity of 0.90 or higher are deemed redundant, with only the earliest article being retained.[6] Lastly, the 'Return Availability' step refines the dataset by ensuring that only news stories with corresponding CRSP data available within the required time window are retained. At this step, it is apparent that the test data sample size is smaller than that of the train data.

Figure 1 illustrates the yearly trend in the total number of news stories for both the train data (solid line) and the test data (dashed line) after applying preprocessing steps. The data covers the period from February 11, 2013, to December 28, 2023. The lower number of news stories in 2013 can be attributed to the absence of the 'significant' tag in the early months of that year. Furthermore, the test data consistently contains fewer samples over time, a discrepancy that arises from the differing time spans of the news data used for training and testing. Despite this, the overall sample size remains relatively stable over time.

4.2 Estimation Setup

The data from 2013 to 2016, covering a period of four years, is used for training (estimation). The subsequent period, from 2017 to 2023, encompassing seven years, is used for testing (out-of-sample evaluation). A rolling window approach is employed, where the training data is advanced by one year at a time to forecast each successive year in the testing dataset. Following the methodology introduced

[6] The decision to employ a high similarity threshold is based on the observation that business news frequently exhibits a consistent structure, especially in cases where the news consists solely of a headline without accompanying body text. This pattern is particularly prevalent in the reporting of key financial figures and economic statistics. Lower similarity thresholds often result in the exclusion of important news stories from the dataset. Therefore, a higher threshold is justified to ensure the inclusion of a more comprehensive set of news stories.

by Peters et al. (2018), we employ a feature extraction approach, commonly referred to as probing. This technique involves directly utilising the pre-trained model to generate features associated with text data. Each input text is processed by the pre-trained model, resulting in the generation of a vector features for each token within text. These features are then applied within a logistic model to estimate the direction of returns. The model is formulated as follows:

$$\mathbb{E}(y_{i,t+1} \mid \mathbf{f_{i,t}}) = \sigma(\mathbf{f'_{i,t}}\boldsymbol{\beta}), \quad \text{where} \quad \sigma(z) = \frac{\exp(z)}{1 + \exp(z)} \tag{2}$$

In Equation (2), the variable $\mathbf{f}_{i,t}$ represents the set of features extracted from the i-th news story at time t. The binary variable $y_{i,t+1}$ indicates the predicted direction of the return for the i-th news story on the subsequent day, $t+1$. The function $\sigma(z)$ denotes the logistic link function. Notably, this framework allows for the substitution of the logistic model with any ML model, offering the potential to capture more complex relationships. However, the logistic model is particularly advantageous due to its lack of hyperparameters, which eliminates the need for hyperparameter tuning, and its typically faster estimation process compared to other alternatives. We use the sign of the return as the direction of the return. Following the methodology outlined by Ke et al. (2019), we estimate the logistic model by employing a three-day return window, spanning from one day before to one day after the target day. Specifically, for day t, we calculate the return using the open prices of days $t-2$ and $t+1$ to determine the return and its corresponding direction. This approach is utilised solely for the estimation of the model, ensuring that no look-ahead bias is introduced into our out-of-sample analysis. This approach enhances the accuracy of the estimation by increasing the model's learning efficiency through the mitigation of noise within the data.

In the feature extraction phase from news stories, we apply the FinText model from the previous year for each respective testing year. For instance, when predicting the year 2017, we utilise the FinText model pre-trained on data from 1997 to 2016. This model is then applied to the train dataset covering the period from 2013 to 2016. This approach ensures that our analysis remains free from look-ahead bias. Notably, no existing LLM incorporates the historical features integrated into the FinText models. Nevertheless, we conduct a comparative analysis with LLaMA 1, LLaMA 2, LLaMA 3, FarmPredict, FinBERT, and LMD to evaluate our model's performance. As previously mentioned, with the exception of the FarmPredict and LMD, the remaining models are vulnerable to look-ahead bias. This bias can compromise the accuracy of their predictions, potentially leading to erroneous investment decisions.

We utilise the smallest available models for each version of LLaMA, including the 7-billion-parameter models for LLaMA 1 and LLaMA 2, and the 8-billion-parameter model for LLaMA 3. Additionally, we incorporate FinBERT, which consists of 110 million parameters, and the base version of FinText, which comprises 125 million parameters. The models differ in their input sequence lengths. Specifically, both FinText and FinBERT accept input sequences with a maximum length of 512 tokens. In comparison, LLaMA 1 can process sequences of up to 2,048 tokens, LLaMA 2 can handle sequences of up to 4,096 tokens, and LLaMA 3 supports sequences as long as 8,192 tokens. This substantial variation in token capacity and model size results in a significant increase in the dimensionality of the vector $\mathbf{f}_{i,t}$, as described in Equation (2). This dimensionality ranges from 768 for smaller models like FinText to 4096 for larger models such as LLaMA. Consequently, the increase in dimensionality leads to a notable rise in the computational time required for estimating the logistic model.

FarmPredict, as introduced by Zhou et al. (2024), requires tuning several hyperparameters. In our study, data from 2013 to 2015 were used for training, while the 2016 data were employed as the validation set. The fine-tuning process followed the methodology outlined by the original study. Initially, the number of factors was set to nine, based on the procedure described by Fan et al. (2022), and is depicted in Figure A3. This method identified four primary factors and five secondary ones. Next, we optimised the remaining hyperparameters using a grid search approach, with the results summarised in Table A4. For the LMD, to construct the feature set, we included all sentiment categories, such as 'Negative,' 'Positive,' 'Uncertainty,' 'Litigious,' 'Strong modal,' 'Weak modal,' and 'Constraining.'[7] This feature set was updated on a rolling window basis, consistent with the overall estimation framework. Additionally, we applied the term frequency-inverse document frequency (TF-IDF) technique, as recommended by Loughran and McDonald (2011). To avoid look-ahead bias, we ensured that, for each year in the sample, only the terms available in the dictionary up to that year were included, thereby excluding any terms introduced in subsequent years.

4.3 Portfolio Setup

To evaluate the performance of LLMs in the context of asset pricing, we construct a zero-net-investment portfolio. Each trading day, we allocate \$1 to long positions and \$1 to short positions, holding these positions for one day. The stocks selected for long and short positions are identified using the estimated logistic model applied to the test data, which generates daily probabilities of positive returns. These probabilities, derived from news stories, are used to rank the stocks in descending order. The top 20% of stocks, based on these probabilities, are selected for long positions, while the bottom 20% are selected for short positions. We present findings for equal-weighted (EW) and value-weighted (VW) portfolios.

[7] 'Complexity' is excluded from the analysis due to its recent introduction, resulting in limited data coverage for the

Table 4: Model Classification Performance

Year	FinText	Llama 1	Llama 2	Llama 3	FarmPredict	FinBERT	LMD
2017	0.5240	0.5199***	0.5217***	0.5227*	0.5035***	0.5200***	0.4976***
2018	0.5236	0.5194	0.5179***	0.5137***	0.5031***	0.5160***	0.5064***
2019	0.5264	0.5279***	0.5240***	0.5247***	0.5089***	0.5223***	0.5128***
2020	0.5128	0.5157***	0.5189***	0.5174***	0.4860***	0.5134***	0.4949***
2021	0.5171	0.5187***	0.5215***	0.5232***	0.5072***	0.5166***	0.5047***
2022	0.5140	0.5142**	0.5146***	0.5158***	0.5002***	0.5123	0.4951***
2023	0.5219	0.5255***	0.5275***	0.5247***	0.5043***	0.5197***	0.5020***
Overall	0.5122	0.5118	0.5139***	0.5123***	0.4993***	0.5120***	0.4993***
Average	0.5200	0.5202	0.5209	0.5203	0.5019	0.5172	0.5019

Note: This table presents the annual performance metrics for all models utilised in the news classification task. The results are derived by aggregating news stories on a daily basis for each individual ticker. In cases where multiple news stories are available for a single ticker on the same day, the mode of the predicted directions is selected as the final predicted direction for that ticker. *, **, and *** denote significance at the 10%, 5%, and 1% levels, respectively, based on the McNemar test for each year or for all years combined (overall), comparing the specified model with FinText. These significance levels assess whether the observed differences in classification performance between the specified model and FinText are statistically significant.

5 Results

Before examining the portfolio results, it is essential to review the accuracy of the outcomes. Table 4 presents the annual performance metrics for all models utilised in the news classification task. The models evaluated include FinText, LLaMA 1, LLaMA 2, LLaMA 3, FarmPredict, FinBERT, and LMD. The reported values represent the classification accuracy for each model across different years, as well as the overall and average accuracies achieved. The symbols *, **, and *** represent statistical significance at the 10%, 5%, and 1% levels, respectively, according to the McNemar test. Significance is evaluated for each year individually and for the aggregate data across all years (overall), comparing the performance of the specified model against FinText.

Despite FinText being significantly smaller—by factors of 56 and 64—compared to the LLaMA models, it demonstrates comparable and, in some years, superior performance. Although year-to-year variations are observed, LLaMA 2 emerges as the best-performing model in terms of average and overall classification accuracies. FinBERT, on the other hand, exhibits comparatively lower performance relative to the other models evaluated. FarmPredict and LMD are among the models with the lowest classification performance in this evaluation. With respect to statistical significance, except for LLaMA 1, where the difference between this model and FinText is not statistically significant for the overall data, the other models generally exhibit statistically significant differences both on a yearly basis and overall.

To further explore the model's accuracy, Table 5 presents the news classification accuracy metrics of

time period under consideration.

Table 5: Evaluation of FinText LLM Performance in News Classification

Year	Total Acc.	Up Acc.	Down Acc.	Up	Up (%)	Down	Down (%)	Total News
2017	0.5240	0.5248	0.5233	11830	47.8928	12871	52.1072	24701
2018	0.5236	0.5645	0.4850	12569	48.6021	13292	51.3979	25861
2019	0.5264	0.4939	0.5580	13048	49.3271	13404	50.6729	26452
2020	0.5128	0.4545	0.5646	12551	47.0798	14108	52.9202	26659
2021	0.5171	0.6297	0.4078	13731	49.2521	14148	50.7479	27879
2022	0.5140	0.5502	0.4805	13540	48.1405	14586	51.8595	28126
2023	0.5219	0.5284	0.5158	12624	48.5296	13389	51.4704	26013
Overall	0.5122	0.5364	0.5043	89893	48.4034	95798	51.5966	-

Note: This table reports the performance metrics of the base version of FinText in the news classification task. The results are derived by aggregating news stories on a daily basis for each individual ticker. In cases where multiple news stories are available for a single ticker on the same day, the mode of the predicted directions is selected as the final predicted direction for that ticker.

FinText by year and all years combined (overall), based on out-of-sample data. The 'Total Accuracy' reflects the overall classification accuracy across all news stories. 'Up Accuracy' and 'Down Accuracy' indicate the classification accuracy for positive and negative returns, respectively. Additionally, the table shows the realised number and percentage of news stories associated with positive and negative returns. The 'Total News' column represents the total number of news stories per year. Despite annual fluctuations in the accuracy of predicting positive and negative returns, the total accuracy remains above 0.50. It is also evident that the realised percentages of positive and negative returns per year, as well as overall, are close to each other, indicating generally balanced out-of-sample data.

5.1 Portfolio Performance

While assessing the accuracy of models is crucial, it does not fully capture their true performance within the context of asset pricing. Table 6 reports the portfolio performance metrics across various models for both EW and VW portfolios. In this table, 'AR' denotes the annualised return, 'SD' the standard deviation, 'SR' the Sharpe ratio, and 'DR (bps)' the average daily return in basis points. The metrics are provided for both long and short, as well as long-short portfolios. The performance of the VW portfolio across all models is observed to be lower compared to the EW portfolio. This suggests that news stories may serve as better predictors of future returns for small-cap stocks compared to large-cap stocks.[8] Additionally, FinText achieves the highest long-short Sharpe ratio of 3.446, surpassing all other models, with an average daily return of 61.211 bps. Notably, despite the LLaMA models being significantly larger in size, pre-trained on the substantially larger corpus, and capable of handling much longer input sequences, FinText outperformed them. Specifically, the LLaMA 1 and 2 models are approximately 54 times larger than FinText, with LLaMA 3 being up to 64 times larger. These comparisons highlight FinText's performance, even though it is developed with a considerably

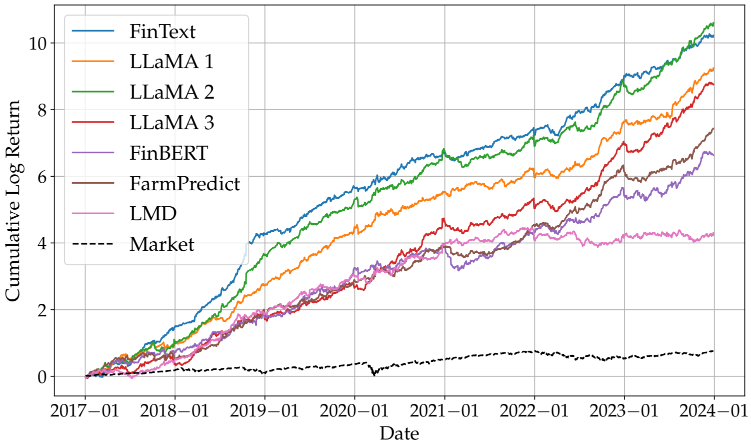
[8] This behaviour is consistent with the findings of Ke et al. (2019) and Chen et al. (2022).

Table 6: Portfolio Performance Metrics

Model	Metric	EW			VW		
		Long	Short	Long-Short	Long	Short	Long-Short
FinText	AR	0.309	1.124	1.449	0.098	0.256	0.371
	SD	0.320	0.422	0.420	0.268	0.343	0.343
	SR	0.965	2.666	3.446	0.365	0.749	1.082
	DR (bps)	14.328	48.226	61.211	5.319	12.513	17.055
LLaMA 1	AR	0.340	0.952	1.308	0.063	0.222	0.301
	SD	0.304	0.433	0.415	0.281	0.336	0.333
	SR	1.120	2.197	3.151	0.224	0.659	0.904
	DR (bps)	15.351	41.564	55.482	4.070	11.045	14.148
LLaMA 2	AR	0.327	1.160	1.503	0.039	0.049	0.105
	SD	0.333	0.459	0.461	0.276	0.364	0.363
	SR	0.984	2.527	3.262	0.142	0.135	0.288
	DR (bps)	15.253	50.292	64.094	3.066	4.571	6.764
LLaMA 3	AR	0.267	0.955	1.238	0.138	0.033	0.187
	SD	0.310	0.461	0.435	0.287	0.362	0.367
	SR	0.860	2.072	2.848	0.479	0.092	0.510
	DR (bps)	12.515	42.159	52.985	7.105	3.922	10.092
FarmPredict	AR	0.235	0.798	1.049	0.068	0.029	0.113
	SD	0.310	0.417	0.404	0.274	0.333	0.327
	SR	0.757	1.913	2.596	0.247	0.088	0.346
	DR (bps)	11.233	35.177	44.959	4.166	3.371	6.612
FinBERT	AR	0.136	0.782	0.934	0.123	0.077	0.216
	SD	0.319	0.469	0.478	0.290	0.361	0.368
	SR	0.425	1.667	1.954	0.423	0.214	0.587
	DR (bps)	7.413	35.416	41.634	6.546	5.632	11.265
LMD	AR	0.001	0.583	0.600	0.115	0.008	0.140
	SD	0.337	0.419	0.427	0.266	0.315	0.305
	SR	0.003	1.392	1.407	0.433	0.027	0.460
	DR (bps)	2.326	26.624	27.464	5.989	2.306	7.404

Note: This table reports the portfolio performance metrics across various models for both equal-weighted (EW) and value-weighted (VW) portfolios.

Figure 2: Portfolio Performance of Long-Short Portfolios



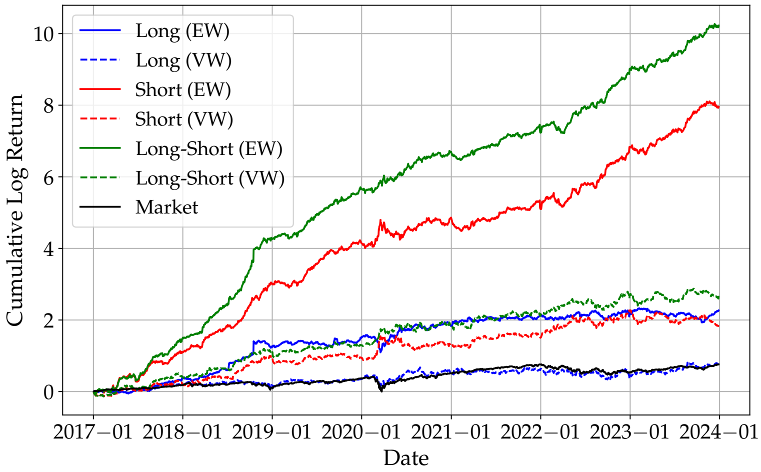
Note: This figure illustrates the cumulative log returns of long-short portfolios (equal-weighted), comparing the performance of models from January 2017 to December 2023.

smaller model size and data. Moreover, when analysing long and short portfolios separately, it is evident that all models perform better on the short leg than the long leg. Notably, FinText achieves the highest Sharpe ratio for the short leg and is the second-best model, following LLaMA 1, on the long leg. It is important to emphasise that FinText, FarmPredict, and LMD are the only models that are entirely free from look-ahead bias. Among them, although FarmPredict performs better than LMD, it still does not reach the performance of FinText.

Figure 2 illustrates the cumulative log returns of long-short portfolios, comparing the performance of models from January 2017 to December 2023. The market performance, represented by the S&P 500, is depicted with a dashed black line. All models consistently outperformed the market during the observation period. Although Llama 2 achieved a slightly higher cumulative log return by the end of the period, FinText generally exhibited superior performance across most of the dates. Among the models evaluated, LMD demonstrated the weakest performance. Figure 3 shows the cumulative log returns for the long, short, and long-short portfolios (each using EW and VW) based on the FinText LLM. These strategies are benchmarked against the market baseline, represented by the S&P 500. The figure indicates that the EW portfolio consistently outperforms the VW portfolio, with short legs generally yielding higher cumulative log returns than long legs over time.

The time required for feature extraction from LLMs is a critical factor, particularly in practical trading scenarios. Due to its relatively small size, the FinText model allows for efficient feature extraction using a CPU. However, this approach becomes impractical for larger models, such as LLaMA, which require a GPU for efficient processing. On an Nvidia A100 GPU, feature extraction from the LLaMA model takes approximately 35 to 45 hours. In contrast, feature extraction from the FinBERT model averages 3 hours, while the FinText model requires only about 2.5 hours for the same task.

Figure 3: FinText Portfolio Performance Across Different Strategies



Note: This figure illustrates the cumulative log returns of FinText under various portfolio strategies.

These results underscore the significantly lower computational demands of FinText for feature extraction. Additionally, as discussed in Subsection 4.2, the larger size of LLaMA inherently increases the dimensionality of the extracted features, leading to longer estimation times. Also, models such as FarmPredict require extensive hyperparameter tuning, which can be computationally intensive, particularly when applied to datasets of the scale used in this study.

5.2 Alpha-Adjusted Portfolio Performance

Table 7 compares the AR for the FinText, evaluated using various factor models. The models analysed include the CAPM, the Fama-French 3-Factor model (FF3), the Fama-French 3-Factor model augmented with momentum (FF3+MOM), as proposed by Carhart (1997), the Fama-French 5-Factor model (FF5), and the Fama-French 5-Factor model with momentum (FF5+MOM). The results for both EW and VW portfolio strategies are reported. The alpha values represent the portfolio's excess returns after controlling for the factors, while the R^2 values denote the proportion of the variance in the portfolio's returns that is explained by each model.

For the EW portfolio, the maximum R^2 achieved is 12.76% on the long leg and 18.96% on the short leg, both under the FF5 model. In the case of the long-short portfolio, the maximum R^2 is 2.76%. Additionally, across all scenarios for the EW portfolio, the AR of the benchmark is closely aligned with the alpha values. These findings are consistent with the VW portfolio as well. Thus, in all instances, the FinText LLM-driven portfolio demonstrates minimal exposure to standard risk factors, and the portfolio's performance is driven more by unique factors than traditional market-wide factors.

Table 7: Alpha-Adjusted Performance of FinText LLM-Driven Portfolio

Model	Metric	EW			VW		
		Long	Short	Long-Short	Long	Short	Long-Short
Benchmark	AR	36.11	121.54	159.27	13.40	31.53	46.56
CAPM	α	30.15	129.85	161.62	8.24	37.86	47.73
	R^2	7.96%	9.56%	0.74%	9.09%	8.36%	0.28%
FF3	α	32.03	127.01	160.65	8.07	36.90	46.59
	R^2	12.54%	18.02%	2.37%	11.00%	12.45%	2.02%
FF3+MOM	α	31.97	126.88	160.47	8.03	36.74	46.40
	R^2	12.59%	18.23%	2.76%	11.04%	12.90%	2.72%
FF5	α	33.26	123.82	158.71	8.20	35.28	45.11
	R^2	12.76%	18.96%	2.73%	11.05%	12.91%	2.57%
FF5+MOM	α	31.97	126.88	160.47	8.03	36.74	46.40
	R^2	12.59%	18.23%	2.76%	11.04%	12.90%	2.72%

Note: This table presents annualised returns (AR) and alpha estimates for a portfolio constructed using the FinText LLM, analysed across several factor models.

5.3 Transaction Costs

Up to this point, the results presented have not accounted for transaction costs. Given the high trading turnover associated with the constructed portfolios, the impact of transaction costs on performance could be significant. Table 8 presents the performance metrics across different models, including transaction costs. Following Frazzini et al. (2012), the average market impact cost for trading large-cap stocks is approximately 11.21 bps. In contrast, the market impact cost is significantly higher for small-cap stocks, averaging 21.27 bps. The categorisation of stocks into large-cap and small-cap groups is determined by the daily median market capitalisation of the stocks listed on the NYSE.

As anticipated, the inclusion of transaction costs and the subsequent comparison of portfolio performance metrics in Table 6 reveal a clear decline in the Sharpe ratios across all models and portfolios. Specifically, for the EW portfolio, FinText exhibits a Sharpe ratio of 1.493, followed closely by LLaMA 2 at 1.492 and LLaMA 1 at 1.220 for the long-short portfolio. In the VW portfolio, including transaction costs results in all Sharpe ratios turning negative. Overall, even after accounting for transaction costs, FinText achieves a notable Sharpe ratio, coupled with an average daily return of 28.531 bps, maintaining its superiority compared to other models. In contrast, LMD shows the weakest performance among all models.

5.4 Alternative LLM Configuration and Portfolio Construction

We further extend our analysis to include alternative FinText LLM configurations. Table 9 provides a comparative analysis of portfolio performance between the base model and various alternative

Table 8: Portfolio Performance Metrics (Including Transaction Costs)

Model	Metric	EW			VW		
		Long	Short	Long-Short	Long	Short	Long-Short
FinText	AR	-0.072	0.685	0.629	-0.189	-0.048	-0.220
	SD	0.320	0.422	0.421	0.269	0.343	0.3434
	SR	-0.225	1.622	1.493	-0.703	-0.139	-0.643
	DR (bps)	-0.786	30.732	28.531	-6.067	0.444	-6.407
LLaMA 1	AR	-0.036	0.526	0.507	-0.223	-0.073	-0.280
	SD	0.304	0.434	0.416	0.282	0.336	0.333
	SR	-0.117	1.214	1.220	-0.792	-0.217	-0.840
	DR (bps)	0.428	24.638	23.568	-7.271	-0.644	-8.888
LLaMA 2	AR	-0.047	0.719	0.689	-0.247	-0.251	-0.482
	SD	0.333	0.460	0.462	0.276	0.364	0.363
	SR	-0.141	1.565	1.492	-0.895	-0.689	-1.327
	DR (bps)	0.397	32.757	31.627	-8.299	-7.327	-16.496
LLaMA 3	AR	-0.109	0.514	0.421	-0.149	-0.266	-0.398
	SD	0.311	0.461	0.435	0.288	0.363	0.367
	SR	-0.351	1.113	0.967	-0.516	-0.733	-1.084
	DR (bps)	-2.397	24.606	20.460	-4.252	-7.949	-13.137
FarmPredict	AR	-0.139	0.379	0.257	-0.219	-0.262	-0.464
	SD	0.311	0.418	0.405	0.274	0.334	0.327
	SR	-0.446	0.907	0.634	-0.798	-0.785	-1.419
	DR (bps)	-3.587	18.518	13.435	-7.186	-8.184	-16.292
FinBERT	AR	-0.241	0.357	0.133	-0.164	-0.224	-0.372
	SD	0.320	0.470	0.479	0.291	0.362	0.369
	SR	-0.751	0.759	0.277	-0.565	-0.618	-1.008
	DR (bps)	-7.510	18.502	9.754	-4.851	-6.293	-12.059
LMD	AR	-0.405	0.192	-0.197	-0.176	-0.278	-0.438
	SD	0.338	0.419	0.427	0.267	0.316	0.305
	SR	-1.200	0.457	-0.463	-0.660	-0.881	-1.435
	DR (bps)	-13.783	11.071	-4.223	-5.578	-9.051	-15.517

Note: This table reports the portfolio performance metrics across various models for both equal-weighted (EW) and value-weighted (VW) portfolios including transaction costs.

Table 9: Portfolio Performance Comparison Across Alternative Model Configurations

Excluding Transaction Costs						
	Base (Benchmark)			Small		
	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.309	1.124	1.449	0.384	0.981	1.381
SD	0.320	0.422	0.420	0.355	0.479	0.502
SR	0.965	2.666	3.446	1.080	2.047	2.748
DR (bps)	14.328	48.226	61.211	17.806	43.384	59.829
	Base (Further)			Small (Further)		
AR	0.234	1.121	1.371	0.276	0.989	1.282
SD	0.321	0.431	0.423	0.337	0.420	0.426
SR	0.729	2.599	3.241	0.820	2.354	3.010
DR (bps)	11.366	48.284	58.160	13.268	42.822	54.614
Including Transaction Costs						
	Base (Benchmark)			Small		
	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.072	0.685	0.629	-0.002	0.542	0.556
SD	0.320	0.422	0.421	0.356	0.480	0.503
SR	-0.225	1.622	1.493	-0.005	1.129	1.105
DR (bps)	-0.786	30.732	28.531	2.508	25.908	26.983
	Base (Further)			Small (Further)		
AR	-0.146	0.683	0.553	-0.108	0.548	0.456
SD	0.321	0.432	0.424	0.337	0.421	0.427
SR	-0.455	1.581	1.305	-0.321	1.303	1.069
DR (bps)	-3.715	30.823	25.549	-1.995	25.275	21.744

Note: This table provides a comparative analysis of the portfolio performance metrics between the base model and several alternative configurations of the FinText model. The equal-weighted portfolio, denoted as EW, and the value-weighted portfolio, denoted as VW, are considered.

configurations of the FinText model, as detailed in Subsection 3.1. The benchmark (the base model in Subsection 5.1) serves as the baseline, while the small model represents a reduced-parameter version. The 'Further' models are pre-trained for an additional 5 epochs. All reported results are based on an EW portfolio. The performance metrics are presented both with and without the inclusion of transaction costs.

Excluding transaction costs, the Sharpe ratio of the long-short portfolio decreases from 3.446 to 2.748 when using the small model, with a corresponding decline in the average daily return from 61.211 to 59.829 bps. This reduction in performance persists after accounting for transaction costs. A comparison between the long and short legs reveals that the decrease in the Sharpe ratio is primarily attributable to the weaker performance of the short leg in the small model. Overall, the importance of model complexity as a critical factor influencing portfolio performance is evident. Notably, transitioning from the base model to the small model reduces the number of parameters from 124.65 million to 51.48 million—a substantial reduction in scale.

We also investigate the effects of further pre-training on portfolio performance. For the base

Table 10: Portfolio Performance Comparison Across Alternative Trading Sizes

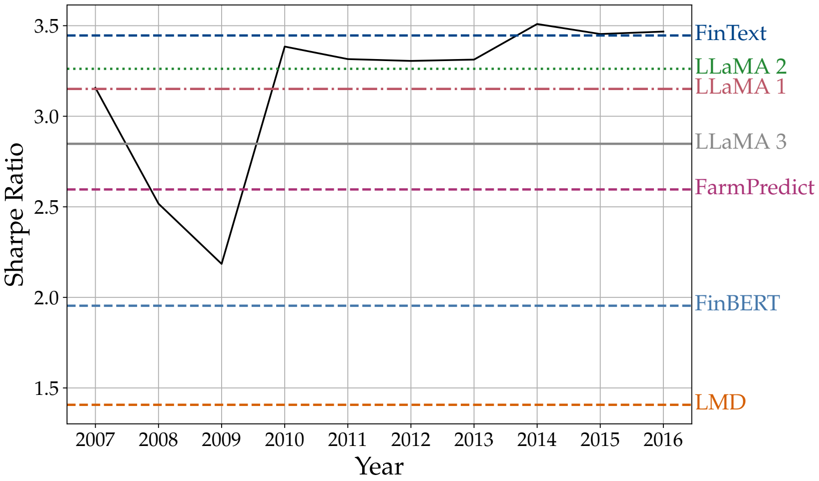
	EW			**VW**		
	Long	Short	Long-Short	Long	Short	Long-Short
	Top/Bottom 20% (Benchmark)					
AR	0.309	1.124	1.449	0.098	0.256	0.371
SD	0.320	0.422	0.420	0.268	0.343	0.343
SR	0.965	2.666	3.446	0.365	0.749	1.082
DR (bps)	14.328	48.226	61.211	5.319	12.513	17.055
	Top/Bottom 15%					
AR	0.423	1.005	1.444	0.088	0.313	0.416
SD	0.358	0.477	0.495	0.285	0.367	0.368
SR	1.181	2.106	2.918	0.308	0.853	1.133
DR (bps)	19.456	44.443	62.426	5.081	15.080	19.222
	Top/Bottom 25%					
AR	0.285	1.107	1.408	0.125	0.261	0.403
SD	0.303	0.391	0.371	0.274	0.315	0.309
SR	0.942	2.832	3.799	0.458	0.829	1.302
DR (bps)	13.150	47.054	58.781	6.461	12.343	17.897

Note: This table presents a comparative analysis of performance metrics across different trading sizes. The equal-weighted portfolio, denoted as EW, and the value-weighted portfolio, denoted as VW, are considered.

model, extended pre-training results in a reduction in the Sharpe ratio of the long-short portfolio, decreasing from 3.446 to 3.241. Additionally, the average daily return declines from 61.211 to 58.160 bps. This pattern persists even after accounting for transaction costs. A similar trend is observed in both the long and short legs of the portfolio. For the small model, excluding transaction costs, there is a marginal improvement in the Sharpe ratio, accompanied by a decrease in the average daily return. However, this improvement in the Sharpe ratio is nullified when transaction costs are included. When comparing the long and short legs, the observed improvements are primarily attributed to enhancements in the performance of the portfolio's short leg. Overall, the base models exhibit superior portfolio performance, and further pre-training does not yield additional gains, despite the significant computational resources required for such extended pre-training.

We also present findings for alternative trading sizes, as shown in Table 10. The percentages (Top/Bottom X%) represent the proportion of stocks allocated to the long and short legs based on the probabilities derived from the model. In the EW portfolio, reducing this proportion increases the average daily return due to decreased diversification; however, this results in a lower Sharpe ratio for the long-short leg. Conversely, increasing the proportion to 25% enhances the Sharpe ratio, albeit at the expense of reducing the average daily return. The notably higher performance of the EW portfolio relative to the VW portfolio is consistent with the benchmark results. These findings suggest that further improvements may be achievable by fine-tuning this parameter, potentially leading to enhanced portfolio performance.

Figure 4: Performance of FinText LLMs Across Time



Note: This figure compares portfolio performance between the rolling and fixed estimation approaches and benchmarks them against models including LLaMA 1, LLaMA 2, LLaMA 3, FarmPredict, FinBERT, and LMD.

5.5 Rolling vs. Fixed Estimation Setup

One of the distinctive features of FinText LLMs, as explained in Subsection 3.2, is the implementation of a weighting method that prioritises the sampling of more recent textual data over older data for each yearly model. This approach ensures that the models remain more closely aligned with the latest information. In our primary estimation setup, described in Subsection 4.2, we utilise a rolling approach in which the FinText model from the preceding year is used to make predictions for the corresponding test year. However, this methodology raises a concern: the superior performance of FinText LLMs might be primarily due to the rolling feature and the incorporated sampling weighting method. To address this concern, we modify our estimation setup by replacing the rolling approach with a fixed estimation approach.

Figure 4 illustrates the portfolio performance comparison between the rolling and fixed FinText models. The X-axis represents the distinct annual versions of FinText from 2007 to 2016, applied to the out-of-sample data from 2017 to 2023. The Sharpe ratio is displayed on the Y-axis. Horizontal reference lines denote the Sharpe ratios achieved by other models, including LLaMA 1, 2, 3, FarmPredict, FinBERT, LMD, and the rolling version of FinText. These Sharpe ratios correspond to the values presented in Subsection 5.1. All results are derived from the long-short portfolio (EW).

The most important finding is that, for the majority of FinText models dating back to 2010, all other models exhibit lower Sharpe ratios. Additionally, there is a notable improvement in Sharpe ratios over time post-2010, with values approaching those of the rolling FinText model. This suggests that even earlier versions of FinText, starting in 2010, possess sufficient predictive power to outperform other models. However, the rolling approach remains a straightforward method for achieving nearoptimal portfolio performance. Another notable finding is the relatively poor performance of the FinText models from the period between 2007 and 2009, particularly the 2009 model. This issue can be attributed to the high degree of over-sampling required due to the limited data available during those years. As demonstrated in Table 2, the token sizes for these specific years were relatively small, necessitating significant over-sampling to reach the target token size. This over-sampling likely diminished the effectiveness of these models compared to those from other years.

5.6 Look-Ahead Bias

A significant challenge in utilising LLMs for accounting and finance research is the risk of look-ahead bias, which can critically compromise the validity of analyses. Several studies have attempted to address this issue by conducting their analyses or testing the robustness of results using data outside the LLM's pre-training period. However, the lack of historical LLMs has limited the ability to perform deeper, multi-level testing. To systematically explore this issue in a controlled environment, we leverage FinText models, starting at the linguistic level and then extending the analysis to return prediction, our target application.

As an initial analysis, we examine the temporal progression of selected term embeddings generated by the base version of the FinText models for the period from 2007 to 2023. Using Principal Component Analysis (PCA) to reduce the high-dimensional embeddings of the selected terms to two principal components, we plot the results in individual subplots. Figure 5 illustrates these dynamics, while Figure 5a to Figure 5h present results for the tokens 'SEC', 'IPO'[9], 'COVID'[10], 'GPT'[11], 'Trump'[12], 'Sunak'[13], 'Brexit'[14], and 'ESG'[15]. The years associated with each data point represent the yearly version of FinText utilised to generate the corresponding representation, with a total of 17 points depicted in each subplot. Certain terms are not anticipated to undergo significant semantic shifts in their meanings across successive historical models. While some changes may have occurred over the analysed time period, they were not substantial enough to result in abrupt or notable transitions. For example, the terms 'SEC' and 'IPO', as illustrated in Figure 5a and Figure 5b, fall into this category. The results for these terms align with our expectations, demonstrating the absence of discernible patterns.

Turning to the 'COVID' in Figure 5c reveals a distinct narrative. Coronavirus began to feature

[9] Initial Public Offering.

[10] COVID-19, caused by the SARS-CoV-2 virus.

[11] Generative Pre-training Transformer, an AI system by OpenAI.

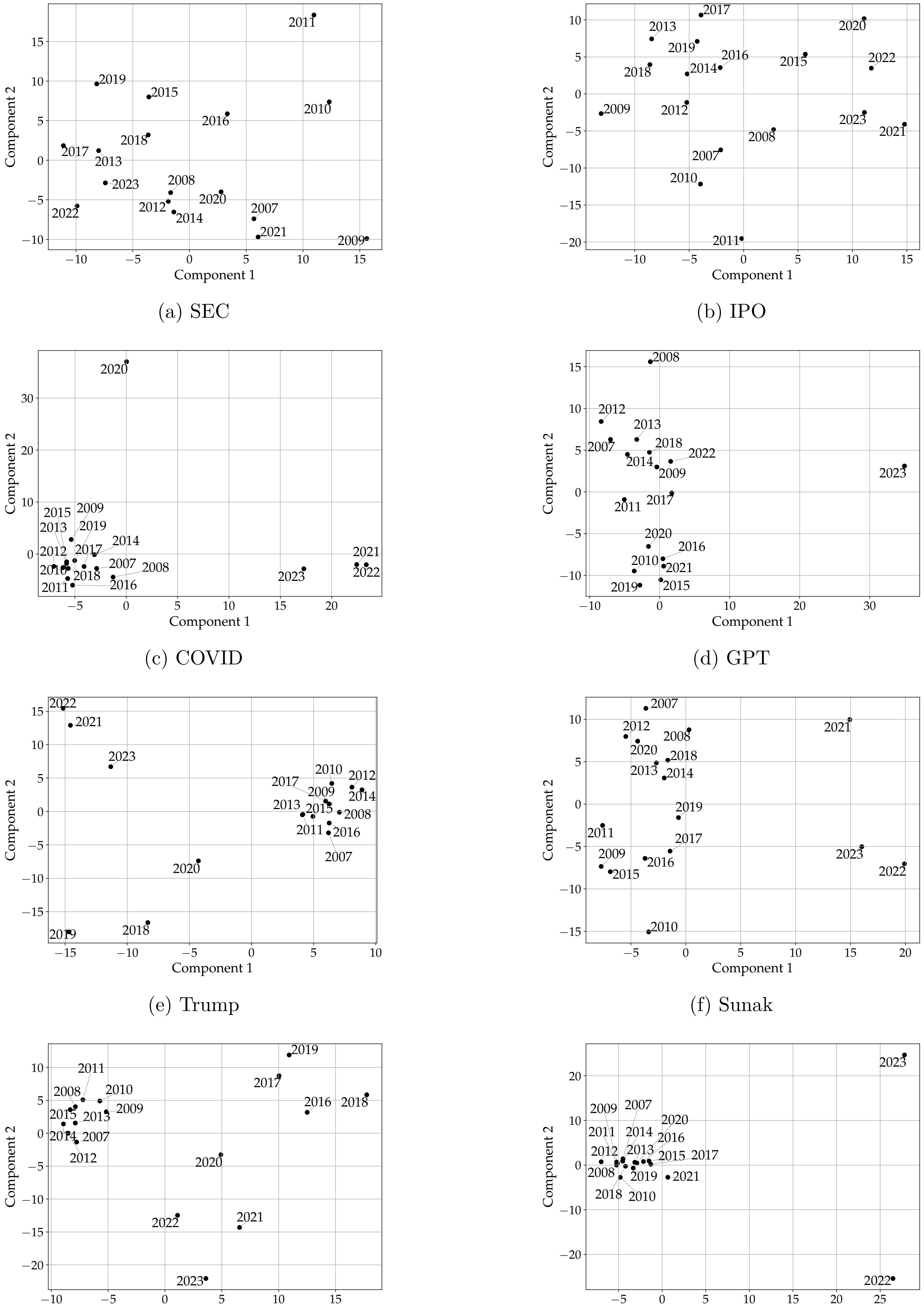
[12] Donald Trump, President of the United States.

[13] Rishi Sunak, Prime Minister of the United Kingdom.

[14] Brexit refers to the United Kingdom's decision to leave the European Union.

[15] Environmental, Social, and Governance, a framework for evaluating a company's sustainability and ethics.

Figure 5: Embedding Dynamics Across Years



(a) SEC

(b) IPO

(c) COVID

(d) GPT

(e) Trump

(f) Sunak

(g) Brexit

(h) ESG

Note: Each figure illustrates the temporal progression of selected term embeddings derived from FinText models spanning the period from 2007 to 2023 (base version). To visualise the high-dimensional embeddings, PCA was employed to reduce their dimensionality to two components for each yearly FinText model. This approach highlights semantic shifts in term meanings across successive historical models.

prominently in media coverage in late December 2019, with the most extensive reporting occurring throughout 2020 as the pandemic expanded globally. Figure 5c clearly illustrates that the year 2020 stands out as distinct from all preceding years, as well as from the subsequent years—2021, 2022, and 2023—which form a separate semantic grouping for this term. These observed changes align with expectations, reflecting a shift in semantics associated with this term across three distinct periods: before 2020, during 2020, and in the years following. Moving to 'GPT' in Figure 5d, this term refers to the LLM developed by OpenAI. The subplot highlights the emergence of two distinct clusters: one encompassing the years 2007 to 2022, and the other confined to 2023. This shift coincides with OpenAI's release of GPT-4 in 2023, an event that garnered substantial media attention and marked the debut of the ChatGPT service. These milestones likely explain the semantic divergence observed for this term in 2023 compared to earlier years.

Figure 5e presents the results for the term 'Trump,' referring to Donald Trump, the President of the United States. The subplot reveals two distinct clusters: the first spans the years 2007 to 2017, while the second covers 2018 to 2023. An examination of his political trajectory highlights that Trump won the U.S. presidential election at the end of 2016, with his presidential term and subsequent significant media coverage primarily occurring from 2017 onward. This shift is clearly reflected in the observed clustering pattern within the subplot. Moving to the next term in Figure 5f, the term 'Sunak' refers to Rishi Sunak, the Prime Minister of the United Kingdom. The analysis reveals two primary clusters: the first spanning 2007 to 2020 and the second covering 2021 to 2023. A major turning point in Sunak's political trajectory occurred in 2020 when he was appointed Chancellor of the Exchequer. Subsequently, he ran for leadership of the Conservative Party in 2022 and assumed the role of Prime Minister in 2023. Consequently, this major shift in semantic meaning is expected to occur during the years 2021 to 2023.

In Figure 5g, the term 'Brexit' represents the United Kingdom's decision to leave the European Union. The subplot reveals the emergence of two distinct clusters: one corresponding to the period from 2007 to 2015, and the other spanning 2016 to 2023. Notably, the year 2016 marks a critical juncture, as the UK held a referendum on its membership in the European Union. This shift reflects the expected semantic divergence between the pre-2016 and post-2016 periods. Finally, 'ESG' in Figure 5h denotes Environmental, Social, and Governance considerations, a widely recognised framework for assessing a firm's sustainability and ethical performance. The subplot reveals two distinct clusters: one extending from 2007 to 2021 and another encompassing 2022 and 2023. These patterns underscore the rapid evolution of ESG-related discourse and practices in recent years. More specifically, during 2022 and 2023, ESG has been subject to heightened political scrutiny and more stringent regu-

Table 11: Evaluation of FinText LLM Performance With and Without Look-Ahead Bias

		Base				Small			
Year	Total news	Benchmark	LAB	Change	Change (%)	Benchmark	LAB	Change	Change (%)
2017	24701	0.5240	0.5238***	6145	24.88	0.5236	0.5229***	6254	25.32
2018	25861	0.5236	0.5198***	6118	23.66	0.5189	0.5192*	6177	23.89
2019	26452	0.5264	0.5285***	6297	23.81	0.5277	0.5223***	6408	24.23
2020	26659	0.5128	0.5136***	6018	22.57	0.5138	0.5155***	6049	22.69
2021	27879	0.5171	0.5154**	6863	24.62	0.5182	0.5150***	6732	24.15
2022	28126	0.5140	0.5120	6382	22.69	0.5094	0.5134***	5919	21.04
2023	26013	0.5219	0.5240***	5672	21.80	0.5173	0.5204***	6212	23.88
Overall	185691	0.5122	0.5125*	43495	23.42	0.5114	0.5117***	43751	23.56

Note: This table reports the annual and overall classification performance of the benchmark model described in Section 5, which is free from look-ahead bias. For comparison, the performance of the benchmark model incorporating look-ahead bias, denoted as LAB in the table, is also presented. The look-ahead bias is introduced by employing the 2023 FinText model to make return predictions for all years. The 'Total News' column corresponds to those presented in Table 5. The 'Change' and 'Change (%)' columns represent the count and percentage, respectively, of news stories where the classification results differ. *, **, and *** denote significance at the 10%, 5%, and 1% levels, respectively, based on the McNemar test for each year or for all years combined (overall), indicating whether the differences are statistically significant. The results are presented for both the base and small models.

latory oversight globally, ultimately increasing accountability and mandating more rigorous reporting standards. The examples illustrate significant shifts in term semantics over time, closely linked to their historical contexts. This temporal evolution introduces look-ahead bias when models using such contextual information are applied to past data, assuming the LLM lacks prior knowledge of these changes.

To deepen our analysis, we proceed to compare the models at the classification level. For this purpose, we introduce a framework that deliberately incorporates look-ahead bias into return prediction. Specifically, we introduce the look-ahead bias by employing the 2023 FinText model to generate return predictions across all years. This approach is analogous to using 2023 model as a fixed model, as described in Subsection 5.5. By doing so, the model is equipped with all information available up to 2023, effectively utilising this information retrospectively to predict returns for prior years. Results are presented in Table 11. This table reports the annual and overall classification performance of the benchmark model described in Section 5, which is free from look-ahead bias. The performance of the benchmark model incorporating look-ahead bias, denoted as LAB in the table, is also presented. The 'Total News' column corresponds to those presented in Table 5. The 'Change' and 'Change (%)' columns represent the count and percentage, respectively, of news stories where the classification results differ. *, **, and *** denote significance at the 10%, 5%, and 1% levels, respectively, based on the McNemar test for each year or for all years combined (overall).

The percentage change in classification results when transitioning from the benchmark model to the LAB model, for both the base and small variants, is approximately 23%. For the base model, the LAB demonstrates superior classification performance compared to the benchmark across the entire

Table 12: Portfolio Performance Comparison With and Without Look-Ahead Bias

	Base						**Small**					
	Benchmark											
	EW			VW			EW			VW		
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.309	1.124	1.449	0.098	0.256	0.371	0.384	0.981	1.381	0.151	0.242	0.409
SD	0.320	0.422	0.420	0.268	0.343	0.343	0.355	0.479	0.502	0.265	0.368	0.360
SR	0.965	2.666	3.446	0.365	0.749	1.082	1.080	2.047	2.748	0.569	0.658	1.137
DR (bps)	14.328	48.226	61.211	5.319	12.513	17.055	17.806	43.384	59.829	7.373	12.288	18.801
	LAB											
	EW			VW			EW			VW		
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.335	1.033	1.383	0.108	0.030	0.155	0.318	0.999	1.333	0.133	0.233	0.382
SD	0.318	0.437	0.425	0.273	0.403	0.389	0.335	0.424	0.433	0.280	0.339	0.334
SR	1.053	2.363	3.252	0.396	0.074	0.397	0.950	2.356	3.079	0.474	0.687	1.147
DR (bps)	15.325	44.838	58.673	5.772	4.391	9.100	14.915	43.275	56.816	6.838	11.555	17.402

Note: This table presents the performance metrics of portfolios evaluated across different model sizes under the benchmark models and the benchmark models with look-ahead bias (denoted as LAB). The equal-weighted portfolio, denoted as EW, and the value-weighted portfolio, denoted as VW, are considered.

sample, and this improvement is statistically significant. Examining the performance by year, 2019, 2020, and 2023 also show statistically significant improvements with the LAB model compared to the benchmark. Similarly, for the small models, the overall classification performance of the LAB model surpasses that of the benchmark, with statistical significance at the 1% level. Yearly results further confirm this pattern, particularly for 2018, 2020, 2022, and 2023, where the LAB model consistently outperforms the benchmark with statistical significance. This observation further provides evidence of look-ahead bias when the most recent model is retrospectively applied in this context. So far, our analysis has concentrated on the language and classification levels. We now shift our focus to the portfolio performance level.

Table 12 presents the performance metrics of portfolios across various model sizes, highlighting comparisons between the benchmark models and their counterparts incorporating look-ahead bias (denoted as LAB in the table). A slight decline in portfolio performance is observed when transitioning from benchmark models to LAB models for the base model. Specifically, under the EW portfolio, the SR decreases from 3.446 to 3.252. However, for the small model under EW portfolios, an improvement in performance is noted, with the SR increasing from 2.748 to 3.079. This finding is consistent with the classification results in Table 11, which demonstrate that, for the small model compared to the base model, the LAB model exhibited higher and statistically significant classification performance over a greater number of years than the benchmark model. Therefore, there are still some indications of look-ahead bias, particularly in the small models. However, the results do not provide as strong support for this bias as observed in the preceding two levels.

It is reasonable to ask why the influence of look-ahead bias becomes progressively weaker and less detectable as one moves from linguistic level to final portfolio performance level. This attenuation can be attributed to the cumulative post-processing steps introduced at various stages of the methodology. At the classification stage, for instance, a logistic model is applied to predict returns (see Subsection 4.2). Additionally, when multiple news stories appear for the same ticker on a given day, we aggregate these predictions by selecting the mode of the forecasted directions. These aggregated predictions then serve as inputs for the subsequent portfolio construction level, which involves its own layers of transformations and procedures before yielding the final performance results. As a consequence, the initial look-ahead bias—present at the earlier, more granular levels—becomes increasingly obscured and diluted by the time the analysis concludes, making it more challenging to trace and identify. This again highlights the importance of presence of look-ahead bias in LLM even if at application level results there is no or weak traces of that. Although this study entirely eliminates look-ahead bias by employing exclusively historical language models, the findings highlight the need for more comprehensive and nuanced analyses when using language models, especially much larger ones that may contain more information than those used in this study, which could significantly impact the obtained results.

5.7 Fine-Tuned Portfolio Performance

A key question that may arise, particularly among ML experts, concerns the rationale for employing a two-step modelling approach in this study—first extracting features from LLMs and subsequently using a secondary model to predict the direction of returns—rather than directly fine-tuning the LLMs for this specific task. This concern can now be addressed by utilising FinText models, which are substantially smaller than larger models like LLaMA, making fine-tuning feasible with modest hardware resources. All results presented are based on using a single Nvidia A100 GPU to ensure consistency and comparability.

Table 13 reports the portfolio performance metrics across various batch sizes for both the base and small fine-tuned FinText models. The benchmark includes the results detailed in Subsection 5.1 for the base model and in Subsection 5.4 for the small model. We continue to employ a rolling approach, where the FinText model for year t is used for predictions in the subsequent year, $t+1$. We fine-tune this model using train data from the preceding four-year period, specifically from $t-3$ to t. This results in the creation of seven fine-tuned models, each applied to seven test years from 2017 to 2023. We also report the results for various batch sizes used during fine-tuning.[16] We allocate 90% of the data

[16] The rationale for presenting results based on batch size is that larger batch sizes require GPUs with higher memory capacities, which typically necessitates the use of more advanced, high-end GPUs. As a result, the larger batch sizes, as illustrated in Table 13, signify the need for these advanced GPUs to perform fine-tuning efficiently, directly impacting

Table 13: Portfolio Performance Comparison Across Different Batch and Model Sizes

	Base						**Small**					
	Benchmark											
	EW			VW			EW			VW		
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.309	1.124	1.449	0.098	0.256	0.371	0.384	0.981	1.381	0.151	0.242	0.409
SD	0.320	0.422	0.420	0.268	0.343	0.343	0.355	0.479	0.502	0.265	0.368	0.360
SR	0.965	2.666	3.446	0.365	0.749	1.082	1.080	2.047	2.748	0.569	0.658	1.137
DR (bps)	14.328	48.226	61.211	5.319	12.513	17.055	17.806	43.384	59.829	7.373	12.288	18.801
	Batch Size = 64											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.182	1.229	1.426	0.079	0.222	0.317	0.328	1.179	1.524	0.097	0.148	0.261
SD	0.334	0.449	0.451	0.283	0.366	0.359	0.346	0.448	0.462	0.304	0.380	0.384
SR	0.543	2.733	3.162	0.278	0.608	0.882	0.949	2.630	3.301	0.318	0.389	0.679
DR (bps)	9.453	52.881	60.841	4.705	11.484	15.153	15.450	50.895	64.936	5.671	8.735	13.278
	Batch Size = 32											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.235	1.023	1.274	0.134	0.108	0.258	0.235	1.023	1.274	0.134	0.108	0.258
SD	0.336	0.446	0.453	0.288	0.349	0.352	0.336	0.446	0.453	0.288	0.349	0.352
SR	0.701	2.294	2.813	0.467	0.310	0.734	0.701	2.294	2.813	0.467	0.310	0.734
DR (bps)	11.600	44.618	54.790	6.965	6.694	12.716	11.600	44.618	54.790	6.965	6.694	12.716
	Batch Size = 16											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.157	1.039	1.213	0.119	0.083	0.218	0.304	1.075	1.396	0.131	0.031	0.179
SD	0.329	0.444	0.435	0.279	0.363	0.362	0.354	0.449	0.467	0.296	0.378	0.382
SR	0.477	2.342	2.791	0.426	0.229	0.603	0.859	2.395	2.990	0.444	0.082	0.468
DR (bps)	8.406	45.234	52.006	6.267	5.910	11.273	14.631	46.769	59.940	6.958	4.045	9.973
	Batch Size = 8											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	0.209	1.007	1.232	0.067	0.169	0.253	0.243	1.047	1.306	0.096	0.008	0.120
SD	0.342	0.496	0.505	0.290	0.370	0.366	0.336	0.440	0.441	0.289	0.381	0.387
SR	0.611	2.030	2.442	0.232	0.457	0.690	0.723	2.379	2.958	0.333	0.022	0.311
DR (bps)	10.657	44.781	53.961	4.342	9.428	12.694	11.897	45.467	55.854	5.455	3.190	7.743

Note: This table presents the performance metrics of portfolios evaluated across different batches and model sizes. The equal-weighted portfolio, denoted as EW, and the value-weighted portfolio, denoted as VW, are considered.

for training and reserve the remaining 10% for validation. This setup allows us to implement early stopping with a patience parameter of three epochs, effectively mitigating the risk of overfitting. For the remaining parameters, we adhere to those outlined in Table 1, with two exceptions: the learning rate, which is adjusted to $1e^{-5}$,[17] and the maximum number of epochs, which is set to 10.[18] As before, the fine-tuning process was initiated with a controlled RNG set to the same state across all models, ensuring the reproducibility of the results.

The initial key observation reveals that, across both EW and VW portfolios, regardless of whether the base or small models are used, the Sharpe ratio and average daily returns are generally lower compared to the corresponding benchmark portfolios. Moreover, among all the batch sizes tested in

the costs associated with fine-tuning.

[17] Lowering the learning rate is a common practice for fine-tuning, as it improves model stability by allowing the model to refine its weights more precisely. A similar or close learning rate is generally employed when fine-tuning models for widely-used benchmarks such as RACE, SQuAD, and GLUE.

[18] Our results show that in all runs, the fine-tuning process consistently halted before reaching the maximum number of epochs. This early termination was driven by the early stopping mechanism, which prevents overfitting.

this study, the largest batch size of 64 yielded the best performance relative to models fine-tuned with smaller batch sizes. Using a larger batch size during fine-tuning may help stabilise the training process by reducing the variance in gradient estimates, which, in turn, leads to more consistent updates and potentially improves model convergence. A crucial question is why fine-tuning might lead to lower performance than benchmark portfolios. This could be due to the signal-to-noise ratio inherent in this task, which is lower than typical fine-tuning tasks in NLP. Another possible explanation is that the hyperparameters used during fine-tuning may require further optimisation to better suit the specific demands of this task. The results, which account for transaction costs, are also presented in Table A5, and they align with the findings discussed here.

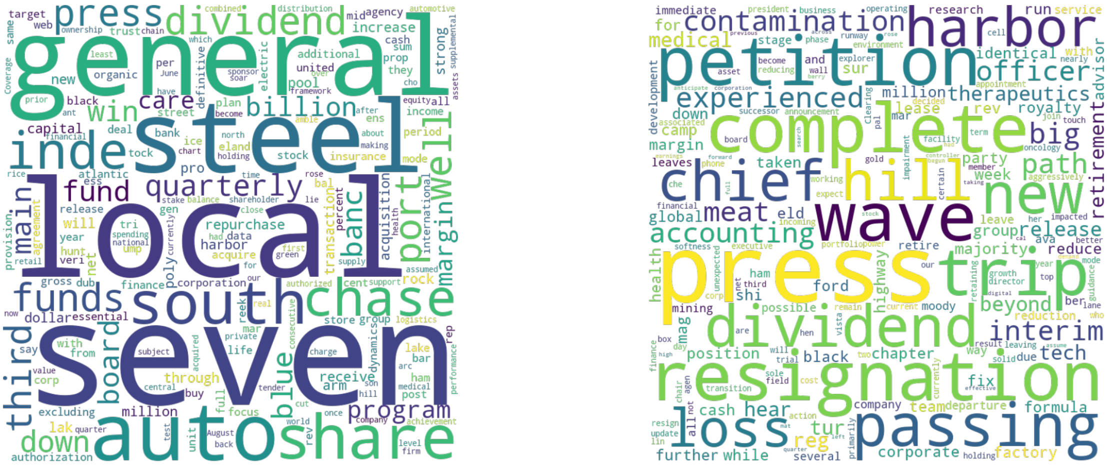
Finally, it is essential to recognise that fine-tuning requires significant computational resources, even when working with small-scale models like FinText. These demands span both the fine-tuning phase and the subsequent deployment of the fine-tuned models on test data. The time required for fine-tuning and deploying each model is notably greater than the durations reported in Subsection 5.1.[19] This extended time commitment poses significant challenges, especially when experimenting with different hyperparameters. Consequently, despite its limitations, adopting a two-step modelling approach proves more efficient, yielding better portfolio performance in our tests.

5.8 Explainable AI

A major limitation in the field of ML, particularly regarding LLMs, is their intrinsic nature as 'black-box' models. This limitation stems from the challenge of interpreting their decision-making processes, which operate with a level of complexity that is difficult to understand. As LLMs, such as the LLaMA model, scale up to billions of parameters, this issue becomes even more pronounced. The vast number of parameters and the intricate architectures of these models introduce substantial challenges when applying XAI methods. However, FinText LLMs present a unique opportunity: their more manageable size makes it feasible to apply XAI methods more effectively, offering greater insights into the inner workings of these models. Lundberg and Lee (2017) introduced SHapley Additive exPlanations (SHAP), which is grounded in coalition game theory. Shapley values, denoted as ϕ_i, quantify the contribution of an individual model input, S (in this context, a token within news stories), to the model's output, $f(S)$. In this case, the output refers to the probability of return direction and

[19] On average, the fine-tuning phase for the base model takes approximately 4 hours per run, with an additional 25 minutes needed for deployment per run. Therefore, the total time required to achieve each portfolio performance presented in Table 13 is around 28 hours for fine-tuning and an additional 3 hours for deployment. For smaller models, these times are reduced by nearly half.

Figure 6: Shapley Value Representation for Top Ranked Daily Trading News Stories



(a) Long Portfolio

(b) Short Portfolio

Note: The figures on the left and right display the word clouds of Shapley values for long and short portfolios, respectively.

the relationship is expressed as follows:

$$\phi_i = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{i\}} |S|! \ \ (|N| - |S| - 1)! \ \ [f(S \cup \{i\}) - f(S)], \tag{3}$$

where $f(S \cup \{i\}) - f(S)$ captures the marginal contribution to the probability of return direction by adding token i to the set S, N represents the set of all model inputs, $|S|!$ represents the number of different ways the chosen set of tokens can be ordered, and $(|N| - |S| - 1)!$ represents the number of ways the remaining tokens can be added. The Shapley value ϕ_i represents the magnitude and sign of the average contribution of token i. As tokens are added to the set, changes in the probability of the return direction reflect their relevance.

Due to the two-step modelling approach used in this study, we combined both steps—feature extraction from the LLM and the logistic model for prediction—into a single block for applying SHAP. The Shapley values are obtained by applying the SHAP explainer to the FinText estimation model block in Subsection 5.1. To reduce computational complexity, the SHAP explainer is applied to daily news stories with the highest estimated probabilities of positive return for long portfolios and the lowest probabilities for short portfolios. This process is repeated within each rolling window, generating a set of tokens and their corresponding Shapley values for each year from 2017 to 2023. The final Shapley values are aggregated by taking the maximum value of shared tokens across these yearly groups while retaining tokens not shared across different years.

Figure 6 illustrates the results, with the long portfolio displayed on the left and the short portfolio on the right. Long portfolio yields mixed results but still features tokens like 'agreement', 'acquisition', ‘deal’, and ‘increase’. Conversely, the short portfolio comprises tokens such as ‘petition’, ‘contamination’, ‘resignation’, and ‘loss’. This indicates that FinText LLMs effectively emphasise the terms that influence long and short portfolios. However, the presence of additional tokens highlights the complexity of modelling language, suggesting that capturing linguistic nuances extends beyond analysing individual tokens in isolation. The advantage of utilising LLMs lies in their ability to process multiple tokens simultaneously, thereby capturing the meaning derived from the interaction of these tokens within the text.

6 Conclusion

This study evaluates the effectiveness of developing specialised LLMs for accounting and finance. By being pre-trained on high-quality, domain-specific historical data, these models have aimed to mitigate look-ahead bias, thereby ensuring that the results obtained are not compromised by any potential information leakage. A diverse range of textual datasets has been utilised, including news articles, regulatory filings, IP records, key corporate information, speeches from the ECB and the FED, transcripts of corporate events, board member information, and Wikipedia for general knowledge, covering the period from 2007 to 2023. Notably, a separate model has been pre-trained for each year within this timeframe. The pre-trained models are based on the RoBERTa architecture and include a base model with approximately 125 million parameters, alongside a smaller variant comprising 51 million parameters, resulting in a total of 34 pre-trained LLMs.

Despite being substantially smaller, these models, called FinText, have consistently outperformed state-of-the-art LLMs, including LLaMA 1, 2, and 3. In an asset pricing context, by constructing a zero-net-investment portfolio, FinText has achieved a Sharpe ratio of 3.45, highlighting the importance of model specialisation and data over sheer model size. The FinText models have demonstrated not only superior performance relative to general-purpose LLMs but also have consistently surpassed specialised financial models such as FarmPredict, FinBERT, and LMD. To further substantiate these findings, our analysis has been extended by constructing alpha-adjusted portfolios, accounting for transaction costs, and exploring alternative configurations of these LLMs. These configurations have included models with smaller sizes, longer pre-training periods, and varying portfolio trading sizes. In all cases, the results have reinforced our initial conclusions. Furthermore, FinText models have been fine-tuned for trading purposes, and while they have achieved satisfactory performance, the simpler two-stage modelling approach has continued to outperform them.

This study systematically examines look-ahead bias at various levels, ranging from the linguistic level to the portfolio level. The findings reveal the potential presence of look-ahead bias at the linguistic level. Additionally, the study demonstrates that while look-ahead bias can significantly distort outcomes at the classification level, its influence diminishes at the portfolio level due to the cumulative effects of post-processing steps. By employing FinText models trained exclusively on historical data, the research effectively mitigates this bias. Therefore, we emphasise the critical importance of addressing look-ahead bias in the application of LLMs in accounting and finance, particularly when working with retrospective data. Failure to account for such bias risks undermining the analytical integrity and reliability of the insights produced.

An advantage of models like FinText is their efficient design, which allows them to perform effectively even in environments with limited computational resources or restricted access to advanced GPUs. In contrast to larger models like LLaMA, FinText's more compact architecture has also facilitated the integration of XAI methods. This integration has provided deeper insights into the model's decision-making process, allowing for the identification of specific token groups that influence predictions for both long and short portfolio legs, thereby enhancing the model's overall interpretability. Our findings indicate that while there are meaningful token combinations in each leg, the inclusion of tokens that cannot be easily categorised as positive or negative sentiment underscores the complexity of modelling language. This complexity extends beyond traditional approaches that model tokens in isolation.

Specialised language models such as FinText present significant opportunities for advancing research by expanding the range of possible studies. These models, being more manageable in size compared to general-purpose LLMs, can be applied or fine-tuned to address specific research objectives, thereby improving precision and relevance in empirical work across accounting, finance, and related areas, including economics, business, marketing, and management. By enabling more targeted analysis, these models can potentially transform how domain-specific textual data is leveraged, offering new insights for advancing empirical research.

References

Antweiler, W. and M. Z. Frank (2004). Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards . *The Journal of Finance 59*(3), 1259–1294.

Brown, T. B. (2020). Language Models are Few-Shot Learners. *arXiv preprint ArXiv:2005.14165.*

Carhart, M. M. (1997). On Persistence in Mutual Fund Performance. *The Journal of Finance 52*(1), 57–82.

Chen, J., G. Tang, G. Zhou, and W. Zhu (2023). ChatGPT, Stock Market Predictability and Links to the Macroeconomy. *Available at SSRN 4660148.*

Chen, Y., B. T. Kelly, and D. Xiu (2022). Expected Returns and Large Language Models. *Available at SSRN 4416687.*

Devlin, J. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805.*

Dubey, A., A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. (2024). The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783.*

Fan, J., J. Guo, and S. Zheng (2022). Estimating Number of Factors by Adjusted Eigenvalues Thresholding. *Journal of the American Statistical Association 117*(538), 852–861.

Frazzini, A., R. Israel, and T. J. Moskowitz (2012). Trading Costs of Asset Pricing Anomalies. *Fama-Miller Working Paper, Chicago Booth Research Paper* (14-05).

Gentzkow, M., B. Kelly, and M. Taddy (2019). Text as Data. *Journal of Economic Literature 57*(3), 535–74.

Hoffmann, J., S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. (2022). Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556.*

Huang, A. H., H. Wang, and Y. Yang (2023). FinBERT: A Large Language Model for Extracting Information from Financial Text. *Contemporary Accounting Research 40*(2), 806–841.

Ke, Z. T., B. T. Kelly, and D. Xiu (2019). Predicting Returns With Text Data. Technical report, National Bureau of Economic Research.

Kim, A., M. Muhn, and V. Nikolaev (2024a). Financial Statement Analysis with Large Language Models. *arXiv preprint arXiv:2407.17866.*

Kim, A., M. Muhn, and V. V. Nikolaev (2024b). Bloated Disclosures: Can ChatGPT Help Investors Process Information? *Chicago Booth Research Paper* (23-07), 2023–59.

Kingma, D. P. and J. Ba (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980.*

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692.*

Lopez-Lira, A. and Y. Tang (2023). Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models. *arXiv preprint arXiv:2304.07619.*

Loshchilov, I. and F. Hutter (2017). Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101.*

Loughran, T. and B. McDonald (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance 66*(1), 35–65.

Loughran, T. and B. McDonald (2016). Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research 54*(4), 1187–1230.

Lundberg, S. and S.-I. Lee (2017). A Unified Approach to Interpreting Model Predictions. *arXiv preprint arXiv:1705.07874.*

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781.*

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018, June). Deep Contextualized Word Representations. In M. Walker, H. Ji, and A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, pp. 2227–2237. Association for Computational Linguistics.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI blog 1*(8), 9.

Rahimikia, E., S. Zohren, and S.-H. Poon (2021). Realised Volatility Forecasting: Machine Learning via Financial Word Embedding. *arXiv preprint arXiv:2108.00480.*

Rajbhandari, S., J. Rasley, O. Ruwase, and Y. He (2020). ZeRO: Memory optimizations Toward Training Trillion Parameter Models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE.

Sarkar, S. K. and K. Vafa (2024). Lookahead Bias in Pretrained Language Models. *Available at SSRN.*

Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance 62*(3), 1139–1168.

Touvron, H., T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. (2023). LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971.*

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288.*

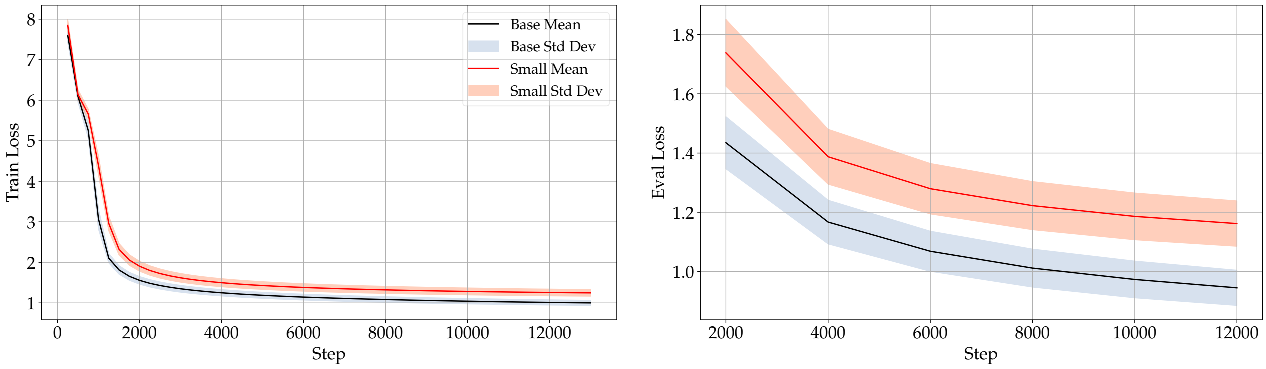
Turc, I., M.-W. Chang, K. Lee, and K. Toutanova (2019). Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962.*

Vaswani, A. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems.*

Zhou, Y., J. Fan, and L. Xue (2024). How Much Can Machines Learn Finance from Chinese Text Data? *Management Science.*

Appendix A

Figure A1: Training and Evaluation Loss Trends of FinText Models (Base and Small)

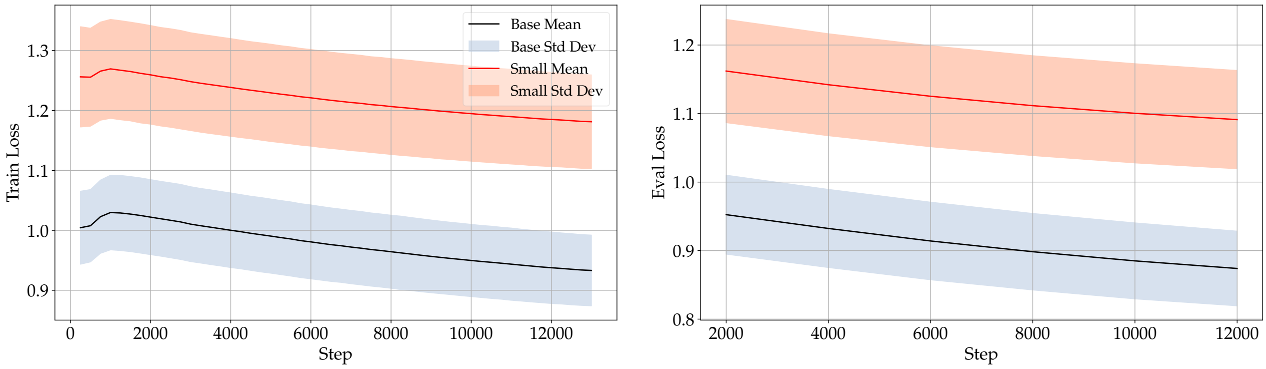


(a) Train

(b) Evaluation

Note: The figures on the left and right display the average and standard deviation of the training and evaluation losses for the FinText models across various steps. Loss values are reported for both the base and small models. These statistics are derived from the performance outcomes of 17 FinText models covering the period from 2007 to 2023. Training losses are recorded every 250 steps, while evaluation losses are recorded every 2000 steps. In both the training and evaluation figures, some pre-trained models exhibit a higher number of steps than the maximum displayed here due to the different number of pre-training samples and reporting intervals defined during the pre-training process. To ensure consistency across different figures, these additional steps were excluded from the figures.

Figure A2: Training and Evaluation Loss Trends of FinText Models (Further Pre-Training)



(a) Train

(b) Evaluation

Note: The figures on the left and right display the average and standard deviation of the training and evaluation losses for the FinText models across various steps. Loss values are reported for both the base and small models, illustrating the results after an additional 5 epochs of pre-training following the initial 5 epochs for all models. These statistics are derived from the performance outcomes of 17 FinText models covering the period from 2007 to 2023. Training losses are recorded every 250 steps, while evaluation losses are recorded every 2000 steps. In both the training and evaluation figures, some pre-trained models exhibit a higher number of steps than the maximum displayed here due to the different number of pre-training samples and reporting intervals defined during the pre-training process. To ensure consistency across different figures, these additional steps were excluded from the figures.

Table A1: Pre-Training Duration (Hours) for FinText LLMs

Year	Base	Base (Further)	Small	Small (Further)
2007	46.43	92.86	23.60	49.89
2008	45.70	91.40	25.86	51.74
2009	40.76	81.53	25.35	50.69
2010	40.12	80.15	24.95	49.90
2011	43.42	86.82	24.61	49.22
2012	39.00	78.10	22.05	46.32
2013	42.20	84.40	23.88	47.74
2014	38.18	80.10	23.73	47.45
2015	37.98	76.00	21.09	44.72
2016	37.96	75.61	20.99	44.57
2017	37.71	75.44	20.91	44.32
2018	37.57	74.83	20.83	44.18
2019	37.53	78.84	20.87	44.24
2020	37.50	78.71	20.85	44.18
2021	40.36	80.71	20.35	43.19
2022	40.22	76.91	22.78	45.56
2023	36.62	73.25	22.73	45.44
Average Time	39.96	80.33	22.67	46.67
Total Time	679.26	1365.66	385.43	793.36

Note: This table details the number of hours allocated to pre-training the FinText models, categorised by specific years. It also provides the average and cumulative pre-training durations required for each group of models. The FinText base and small models consist of approximately 125 million and 51 million parameters, respectively. For the 'Further' versions, the models underwent an additional 5 epochs of pre-training. The reported time for the 'Further' versions represents the total duration, including the base variant pre-training time and the additional time spent on further pre-training. The pre-training was conducted using various combinations of GPUs, including configurations of up to four NVIDIA A100 GPUs (80GB) and two NVIDIA Grace-Hopper GPUs (GH200 480). Despite the variability in hardware configurations, all model parameters—such as the training batch size per device, evaluation batch size per device, and the number of gradient accumulation steps—were optimised to utilise the maximum available memory, ensuring that the total training batch size remained consistent across all models.

Table A2: Hourly Electricity Usage and Cumulative Cost for Pre-Training FinText LLMs

Hour	Cum. kWh Per Model	Cum. Cost Per Model	Cum. Cost All Model	Cum. CO_2 (kg) Per Model	Cum. CO_2 (kg) All Model
Base Models					
1	0.4	0.1	1.3	0.1	1.4
5	8.9	1.8	30.2	1.8	31.3
9	17.7	3.5	60.0	3.7	62.1
13	26.2	5.2	89.2	5.4	92.3
17	35.0	7.0	119.0	7.2	123.2
21	43.6	8.7	148.4	9.0	153.6
25	52.4	10.5	178.3	10.9	184.5
29	61.5	12.3	209.0	12.7	216.3
33	70.5	14.1	239.8	14.6	248.2
37	79.4	15.9	270.1	16.4	279.5
41	87.6	17.5	298.0	18.1	308.4
45	90.8	18.2	308.9	18.8	319.7
47	**92.4**	**18.5**	**314.3**	**19.1**	**325.3**
Small Models					
1	0.4	0.1	1.3	0.1	1.3
5	7.4	1.5	25.3	1.5	26.2
9	14.9	3.0	50.6	3.1	52.4
13	22.6	4.5	76.8	4.7	79.5
17	30.3	6.1	102.9	6.3	106.5
21	37.9	7.6	128.9	7.8	133.4
25	**44.1**	**8.8**	**150.0**	**9.1**	**155.2**

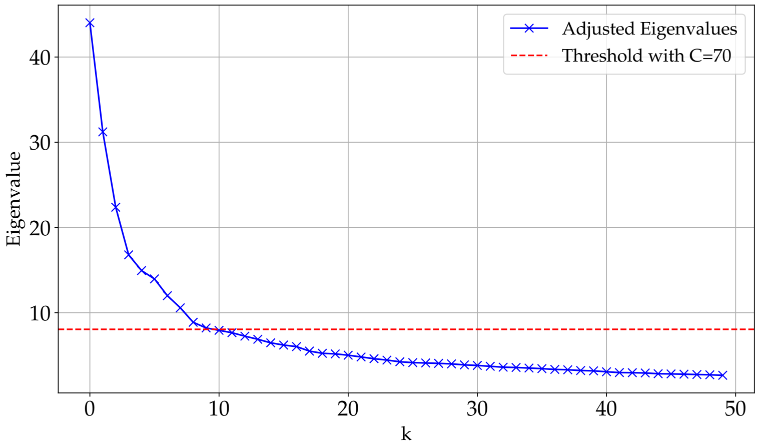
Note: This table provides the estimated cumulative electricity consumption in kWh and the corresponding cumulative costs (in pounds) for the pre-training of each individual model and the total across all 17 models during the period from 2007 to 2023. To streamline the presentation, data is reported at four-hour intervals, with intermediate results omitted. The calculations for electricity costs are based on an average electricity rate of £0.20 per kWh. It is important to note that these values are approximate due to variations in the GPU configurations used during pre-training, and actual electricity consumption may vary. The entire electricity used is also fully traceable and sourced exclusively from renewable energy. Therefore, we convert kWh to kg of CO_2 using the greenhouse gas reporting conversion factors from the UK Department for Business, Energy and Industrial Strategy. The conversion factor is 0.20707 kg CO_2 equivalent per kWh saved when the energy is sourced from renewable energy. This factor reflects the carbon emissions produced by UK power stations per kWh generated and includes other greenhouse gases, such as methane and nitrous oxide, converted to their CO_2 equivalents. Bold values highlight the estimated final electricity consumption, cost, and CO_2 emissions for each group of models. Results for the further pre-trained models are omitted from this table, as they closely resemble those of the base and small models. Additionally, these values account solely for the electricity consumption of the GPUs, leaving out other related costs and emissions from server components like the CPU, RAM, and other hardware. Our servers operate with a power usage effectiveness (PUE) of 1.4, and this value has been incorporated into our calculations.

Table A3: Yearly Sample Size Breakdown of Textual Data Sources

Year	Key Info	News	IP	Filling	ECB	FED	Transcript	Wikipedia	Board Info	Total
2007	2.319	14.324	0.052	9.780	0.009	0.006	0.085	0.244	0.011	26.831
2008	2.882	16.352	0.064	10.898	0.010	0.007	0.276	0.268	0.019	30.776
2009	3.418	18.251	0.076	12.096	0.011	0.008	0.499	0.290	0.034	34.684
2010	4.104	20.205	0.088	13.470	0.012	0.008	0.819	0.313	0.041	39.059
2011	4.768	22.136	0.100	15.213	0.014	0.009	1.248	0.337	0.056	43.879
2012	5.394	23.880	0.111	17.761	0.015	0.009	1.684	0.361	0.069	49.283
2013	6.001	25.206	0.121	20.501	0.016	0.009	2.188	0.385	0.068	54.496
2014	6.604	26.577	0.130	22.490	0.017	0.010	2.713	0.411	0.076	59.026
2015	7.250	27.908	0.137	24.508	0.017	0.010	3.227	0.437	0.078	63.573
2016	7.877	29.286	0.142	26.566	0.018	0.011	3.728	0.463	0.082	68.172
2017	8.584	30.870	0.147	28.910	0.020	0.012	4.330	0.490	0.091	73.453
2018	9.307	32.149	0.149	31.061	0.020	0.012	4.973	0.513	0.087	78.271
2019	9.985	33.459	0.150	33.289	0.021	0.013	5.618	0.533	0.090	83.158
2020	10.663	34.942	0.151	35.893	0.022	0.013	6.404	0.555	0.125	88.768
2021	11.832	38.843	0.152	43.599	0.026	0.015	8.293	0.583	0.105	103.449
2022	12.529	40.840	0.152	47.078	0.027	0.016	9.397	0.597	0.132	110.767
2023	13.052	42.462	0.152	50.001	0.028	0.017	10.356	0.597	0.146	116.810

Note: This table provides a breakdown of the yearly sample sizes (in millions) for all textual data sources from 2007 to 2023. The columns represent data sources used for pre-training our LLMs, including Key Information, News, intellectual property (IP), Filing, European Central Bank (ECB) and Federal Reserve (FED) speech, Transcript, Wikipedia, and Board Information. The total sample size for each year is given in the last column.

Figure A3: Adjusted Eigenvalues



Note: This figure shows the top adjusted eigenvalues, revealing the presence of four major factors alongside five weaker factors, based on the adjusted eigenvalue thresholding with $C = 70$.

Table A4: Cumulative Log Returns Across Hyperparameter Configurations in FarmPredict Model

κ	$\lvert\hat{S}\rvert$		
	500	1000	2000
438	1.4487	1.4274	1.4317
514	0.8615	1.3014	1.3541
604	0.6141	1.3395	1.1866
734	0.0953	-0.0781	-0.1339
922	0.0767	-0.2328	-0.3086
1216	-0.2647	-0.2856	-0.5043
1615	-0.2576	-0.5562	-0.5028
2433	-0.5228	-0.3293	-0.3302
3931	-0.4696	-0.4935	-0.5170

Note: This table presents the cumulative log returns obtained from various combinations of κ and $|\hat{S}|$ for the year 2016. The κ values are set based on data used for training from 2013 to 2015. The maximum cumulative log return of 1.4487 is achieved with the combination $\kappa = 438$ and $|\hat{S}| = 500$.

Table A5: Portfolio Performance Comparison Across Different Batch and Model Sizes (Including Transaction Costs)

	Base						**Small**					
	Benchmark											
	EW			VW			EW			VW		
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.072	0.685	0.629	-0.189	-0.048	-0.220	0.384	0.981	1.381	0.151	0.242	0.409
SD	0.320	0.422	0.421	0.269	0.343	0.343	0.355	0.479	0.502	0.265	0.368	0.360
SR	-0.225	1.622	1.493	-0.703	-0.139	-0.643	1.080	2.047	2.748	0.569	0.658	1.137
DR (bps)	-0.786	30.732	28.531	-6.067	0.444	-6.407	17.806	43.384	59.829	7.373	12.288	18.801
	Batch Size = 64											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.200	0.795	0.611	-0.209	-0.078	-0.270	-0.055	0.745	0.706	-0.191	-0.152	-0.327
SD	0.335	0.450	0.452	0.283	0.366	0.360	0.346	0.449	0.462	0.304	0.381	0.385
SR	-0.597	1.766	1.353	-0.737	-0.212	-0.751	-0.159	1.659	1.529	-0.627	-0.400	-0.850
DR (bps)	-5.680	35.605	28.359	-6.698	-0.422	-8.162	0.249	33.601	32.361	-5.731	-3.180	-10.043
	Batch Size = 32											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.146	0.592	0.462	-0.153	-0.191	-0.327	-0.090	0.652	0.578	-0.153	-0.209	-0.346
SD	0.336	0.447	0.454	0.288	0.349	0.353	0.327	0.462	0.456	0.299	0.383	0.386
SR	-0.435	1.325	1.017	-0.530	-0.545	-0.927	-0.275	1.411	1.268	-0.512	-0.546	-0.898
DR (bps)	-3.544	27.463	22.427	-4.408	-5.151	-10.506	-1.425	30.098	27.087	-4.306	-5.407	-10.801
	Batch Size = 16											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.225	0.610	0.401	-0.168	-0.215	-0.366	-0.079	0.644	0.581	-0.156	-0.267	-0.407
SD	0.330	0.444	0.435	0.279	0.363	0.363	0.355	0.450	0.467	0.297	0.378	0.382
SR	-0.684	1.372	0.921	-0.601	-0.591	-1.010	-0.224	1.432	1.243	-0.526	-0.706	-1.064
DR (bps)	-6.766	28.147	19.686	-5.112	-5.905	-11.923	-0.591	29.586	27.463	-4.447	-7.775	-13.253
	Batch Size = 8											
	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short	Long	Short	Long-Short
AR	-0.175	0.577	0.419	-0.220	-0.129	-0.333	-0.140	0.617	0.494	-0.191	-0.290	-0.465
SD	0.343	0.497	0.505	0.291	0.371	0.367	0.336	0.441	0.442	0.289	0.381	0.388
SR	-0.509	1.161	0.828	-0.756	-0.348	-0.907	-0.416	1.401	1.117	-0.661	-0.760	-1.197
DR (bps)	-4.563	27.665	21.561	-7.043	-2.408	-10.532	-3.283	28.370	23.510	-5.931	-8.625	-15.456

Note: This table presents the performance metrics of portfolios, including transaction costs, that are evaluated across different batch and model sizes. The equal-weighted portfolio, denoted as EW, and the value-weighted portfolio, denoted as VW, are considered.

Appendix B

Preprocessing Steps for Fillings

The preprocessing steps implemented for filings were outlined as follows. The goal of these steps was to extract, clean, and preprocess text data while preserving relevant content and removing unnecessary information such as HTML tags, metadata, and repetitive punctuation. The preprocessing pipeline involved the following steps:

1. **Initial Setup:** Libraries such as `BeautifulSoup`, `pandas`, `numpy`, `argparse`, and `nltk` were imported for parsing, data manipulation, and text processing. Command-line arguments were defined to allow custom inputs, such as the year, month, quarter, minimum sentence length, and alphanumeric proportion thresholds.
2. **HTML Parsing and Cleaning:** The `BeautifulSoup` library was used to parse HTML content. Specific sections such as 'ITEM 1' or headers like `ims-header` were targeted for processing. Unwanted tags, tables, and metadata were removed using extraction methods and regular expressions.
3. **Text Preprocessing:** Non-alphanumeric characters, excessive white-space, and unwanted substrings (e.g., 'BEGIN PRIVACY-ENHANCED MESSAGE') were removed. Strings were optimised by filtering out repeated punctuation and irrelevant sections such as page numbers and placeholders.
4. **Advanced Filtering:** Text was tokenised into sentences using `nltk`. Filtering criteria were applied to remove sentences containing excessive links, colons, or other undesired patterns. A robust filtering mechanism was used to retain meaningful sentences, requiring a minimum of 2 words and at least 70% alphanumeric characters. This excluded incomplete, overly short, or symbol-heavy sentences, ensuring the processed text focused on linguistically relevant content.

Preprocessing Steps for BoardEx

The following steps outlined the methodology employed to preprocess and convert BoardEx data into text format for analysis:

1. **Data Loading:**
   - Data files for employment, achievements, education, details, and activities were loaded from the WRDS.
2. **Data Cleaning and Formatting:**
   - Duplicates were removed from each dataset based on relevant columns, such as `DirectorID`, `DirectorName`, `CompanyName`, etc.
   - Missing or erroneous data entries were filtered out, e.g., rows with invalid `DirectorName` or `DirectorID`.
3. **Data Integration:**
   - Each dataset was indexed by `DirectorID` to facilitate merging.
   - Gender information was standardised and mapped to pronouns (`he/she/they`).
   - Multiple datasets were combined to form a unified structure for generating text narratives.

4. **Narrative Generation:**
   - Templates were defined for each category (e.g., education, employment, achievements) to generate sentences programmatically.
   - Generated text was adjusted for grammatical correctness (e.g., usage of 'a/an' based on vowels).
   - Randomised weights, ranging between 0.3 and 0.7, were assigned to formal names and pronouns in narratives. At each decision point where a formal name or pronoun could be used, the system referenced the assigned weights to probabilistically select one option. For instance, if the weight for pronouns was set to 0.4, there was a 40% chance that a pronoun would be chosen and a 60% chance that a formal name would be used. This approach ensured a dynamic and balanced distribution across the narrative while avoiding predictable patterns.
5. **Data Shuffling and Randomisation:**
   - Narrative sentences were shuffled within each director's record to avoid sequential patterns.
   - Inner dictionary structures were randomly shuffled to preserve variability in data presentation.

The templates used for creating text descriptions were outlined below. These templates were categorised into different types based on the type of information being processed (e.g., education, achievements, employment, etc.). The templates were parameterised with variables, including `\name` (Director's name), `\role` (Employment or activity role), `\company` (Company or organisation name), `\university` (University name), `\qualification` (Degree or qualification name), `\start_date` and `\end_date` (Dates), `\activity` (Activity name), and `\achievement` (Achievement description).

1. **Education Templates:**
   - `In \year, \name received a \qualification.`
   - `\name graduated in \year with a \qualification.`
   - `\name holds a \qualification, having completed it in \year.`
   - `\name graduated from \university in \year with a \qualification.`
   - `\name is an alumnus of \university, having completed a \qualification in \year.`
2. **Achievements Templates:**
   - `\name was given the \achievement in \month \year.`
   - `In \month \year, \name was recognised as \achievement.`
   - `\name received the \achievement award in \year.`
   - `\name was inducted into \organisation in \year.`
   - `\name was honoured by \organisation in \year.`
3. **Employment Templates (Static for Describing Past Employment):**
   - `\name worked as a \role at \company from \start_date to \end_date.`
   - `\name held the role of \role at \company from \start_date to \end_date.`
   - `From \start_date to \end_date, \name served as \role for \company.`
   - `\name held various positions at \company from \start_date to \end_date.`

4. **Employment Templates (Dynamic for Describing Ongoing Roles):**
   - `\name has been working as a \role at \company since \start_date.`
   - `\name is currently employed as \role at \company.`
   - `\name has held the position of \role at \company since \start_date.`
5. **Activity Templates:**
   - `\name participated as a \role in \activity from \start_date to \end_date.`
   - `Since \start_date, \name has been involved in \activity as a \role.`
   - `\name was an active \role in \organisation until \end_date.`
6. **Biographical Details Templates:**
   - `\name was born on \day \month \year.`
   - `On \day \month \year, \name was born.`
   - `\name passed away on \day \month \year.`
   - `In \year, \name passed away.`

Templates were selected randomly from the respective category for each record to enhance variety in generated descriptions. This randomisation ensured that repetitive patterns were minimised in the output. Additionally, shuffling was applied to intermediate lists of text entries to prevent deterministic outputs. Dynamic formatting rules, such as proper capitalisation and corrections for articles, were also applied.