FinBERT2: A Specialized Bidirectional Encoder for Bridging the Gap in Finance-Specific Deployment of Large Language Models

Anonymous Author(s)

ABSTRACT

In natural language processing (NLP), the focus has shifted from encoder-only models like BERT to decoder-only large language models (LLMs) such as GPT-3. However, LLMs' practical application in the financial sector has reveals three limitations: (1) LLMs often perform worse than fine-tuned BERT on discriminative tasks, such as market sentiment analysis in financial reports; (2) Application on generative tasks heavily relies on retrieval augmented generation (RAG) method to provide current and specialized information, which requires high retrieval performance; (3) There are additional inadequacies in other feature-based scenarios, such as topic modeling. We introduce FinBERT2, a specialized bidirectional encoder that has been pretrained on a high-quality, financial-specific corpus of 32b tokens. This represents the largest known Chinese financial pretraining corpus for models of this parameter size. As a better backbone, FinBERT2 can bridge the gap in financial-specific deployment of LLMs through the following achievements: (1) Discriminative fine-tuned models (Fin-Labelers) outperform other (Fin)BERT variants by 0.4%-3.3% and leading LLMs by 9.7%-12.3% on average across five financial classification tasks. (2) Contrastive fine-tuned models (Fin-Retrievers) outperform both open-source (e.g., +6.8% avg improvement over BGE-base-zh) and proprietary (e.g., +4.2% avg improvement over OpenAI's text-embedding-3-large) embedders across five financial retrieval tasks; (3) Building on FinBERT2 variants, we construct the Fin-TopicModel, which enables superior clustering and topic representation for financial titles, yielding +0.07 and +0.04 improvements in coherence and informativeness scores compared to BGE-base-zh. Our work highlights the unique value of encoder-only LMs in the era dominated by decoder-only LLMs, particularly for specialized financial applications. [1]

KEYWORDS

FinBERT, Pretraining, Dense Retriever, Topic Modeling, Financial NLP, Domain-Specific LM

ACM Reference Format:

Anonymous Author(s). 2025. FinBERT2: A Specialized Bidirectional Encoder for Bridging the Gap in Finance-Specific Deployment of Large Language Models. In *Proceedings of* . ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Early large language models (LLMs) primarily relied on encoder-only architectures with masked language modeling (MLM), such as BERT [8], RoBERTa [21], and XLM [18]. However, between 2018 and 2021, the field shifted from single-task fine-tuning to large-scale multi-task learning. GPT-3 [4] demonstrated that scaling [42] could significantly narrow the performance gap between autoregressive and other architectures. Moreover, autoregressive models offer advantages such as greater task adaptability, a unified modeling paradigm [36], and reduced engineering complexity. Consequently, decoder-only models have become the dominant paradigm in LLM development.

Similarly, the financial domain has witnessed a shift from early FinBERT ([2, 7, 12] to large-scale FinLLMs ranging from billions to trillions of parameters, such as FinGPT [35], BloombergGPT [39], and FinLlama [17]. These models leverage domain-specific data and undergo post-training on financial text corpora, enhancing their understanding and generation capabilities in financial applications. However, LLMs, including fin-adapted versions, do not fully replace BERT. They still face limitations in real-world deployment.[2]

Firstly, while LLMs exhibit strong generalization and robustness as large-scale multi-task models, they are not always optimal for specific NLU tasks. In certain single-task scenarios, fine-tuned BERT-base models often outperform them. For instance, latex Kocoń et al. [16] evaluated ChatGPT on 25 analytical NLP tasks and found that, compared to state-of-the-art (SOTA) methods, its zero-shot and few-shot performance dropped by approximately 25% on average. The decline was even more pronounced for complex tasks. Similarly, Hu et al. [11] reported that in fake news detection, GPT-3.5 underperformed compared to specialized smaller models like BERT. Moreover, LLMs are costly and slow for data-intensive tasks like labeling financial reports, while smaller BERT-like models (0.1B parameters) are more efficient.

Secondly, in generative scenarios requiring external financial knowledge, such as real-time question answering (QA) on financial reports, LLMs rely on embedding-based retrieval [13] to ensure accuracy and timeliness. This approach, known as retrieval-augmented generation (RAG) [20], necessitates efficient offline retrieval. Consequently, dense retrievers (DRs) built on dual-encoder BERT architectures, such as M3E [37], BGE [41], and BCE [28], have become mainstream. These DRs achieve high retrieval accuracy on general benchmarks [25] through in-batch negative learning on large-scale weakly supervised sentence pairs and contrastive fine-tuning with mined hard negatives. However, despite extensive training, they often underperform in specialized domains such as finance and law.

Thirdly, beyond retrieval, LLMs are less effective in feature-based tasks such as clustering and topic modeling [1], text-to-image guidance [30], and measuring earnings surprises and market reactions [24]. Generative models are often impractical for these applications, as they require compact and efficient feature encoding [26] and flexible fine-tuning for task-specific adaptations. For instance, topic modeling prioritizes industry-specific features, while stock

[1]Our code, models and datasets will be accessible at https://github.com/valuesimplex/FinBERT2

[2]Classical BERT was once considered an LLM, but BERT less than 1 billion parameters is now seen as a Tiny LM and no longer meets the criteria of an LLM

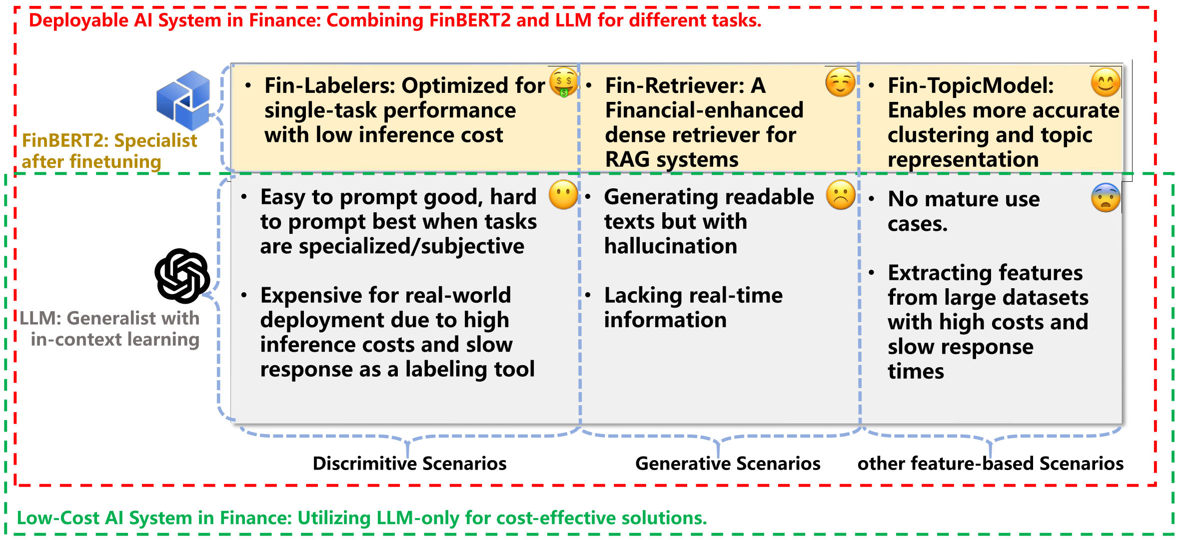

Figure 1: Illustration of how FinBERT2 bridges the gap in finance-specific deployment. The comparison presents two solution: (1) A deployable AI system combining FinBERT2 (serve as domain specialist after fine-tuning) with LLM (serve as general-purpose model with in-context learning capabilities), and (2) A conventional (Fin)LLM-only system.

return prediction relies on sentiment features. However, decoder-only LLMs struggle to meet these requirements due to their inherent architectural constraints.

The above challenges hinder the applications in small and medium-sized financial enterprises, where scenarios are often specialized and diverse. To address this, we re-evaluate the value of lightweight, localized, and customizable FinBERT models and propose a hybrid architecture that integrates FinBERTs (as domain experts mitigating these limitations) with LLMs (as general-purpose generative models with in-context learning capabilities). Specifically, we pre-train FinBERT2, an enhanced version of its predecessor FinBERT1 [10]. FinBERT2 is trained on a carefully curated Chinese financial corpus comprising 32B tokens and further optimized for downstream tasks such as labeling, retrieval, and topic representation. In these tasks, FinBERT2 can effectively replace, assist, or complement LLMs, respectively, offering a more efficient and deployable NLP system for financial applications.Our contributions can be summarized as follows:

1) We pre-trained FinBERT2 on a 32B token Chinese financial corpus to inject domain knowledge. To the best of our knowledge, this is the largest pre-training corpus for a BERT-like language model in the Chinese finance domain, and it is trained with a finance-customized tokenizer.

2) As a more efficient and high-performance alternative to labeling, Fin-Labelers outperform other (Fin)BERT variants by 0.4%–3.3% and leading LLMs (e.g., GPT-4-turbo, Claude 3.5 Sonnet, Qwen2) by 9.7%–12.3% on average across five financial classification tasks.[3]

3) As an enhanced RAG assistant, Fin-Retrievers surpass both open-source and proprietary embedding models. They achieve an average improvement of +6.8% over BGE-base-zh and +4.2% over OpenAI's text-embedding-3-large across five financial retrieval tasks.

4) In feature-based applications such as topic modeling, Fin-TopicModel, built on FinBERT2 variants, enables superior clustering and topic representation for financial titles. It achieves +0.07 and +0.04 improvements in coherence and informativeness scores, respectively, compared to BGE-base-zh.

To promote community research and the wider application of FinBERT2, we open-source the datasets and the weights of the FinBERT2 series of models.

2 METHODS

2.1 Overview of FinBERT2

As shown in Figure 2,our work starts from the data layer consisting of Fin-Corpus and Fin-downstream datasets, through the foundation layer where produces Fin-Tokenizer and FinBERT2-base/large , to the downstream application layer featuring three main components: (1) Fin-Labeler, fine-tuned for five downstream financial tasks including market sentiment classification, industry classification, and named entity recognition (NER); (2) Fin-Retriever, trained via contrastive learning for five financial retrieval tasks; (3) Fin-TopicModel, an enhanced version of TopicModel, integrating key components and improvements derived from other FinBERT2 variants.

2.2 Pre-training of FinBERT2

2.2.1 Fin-Corpus for Pre-training. The FinBERT2 model has seen a significant expansion in its pre-training corpus, with the total token count increasing to 32B (99 G).

[3] The term "Labeler" highlights its role in the LLM era—fine-tuning BERT for task-specific, large-scale, real-time text processing.

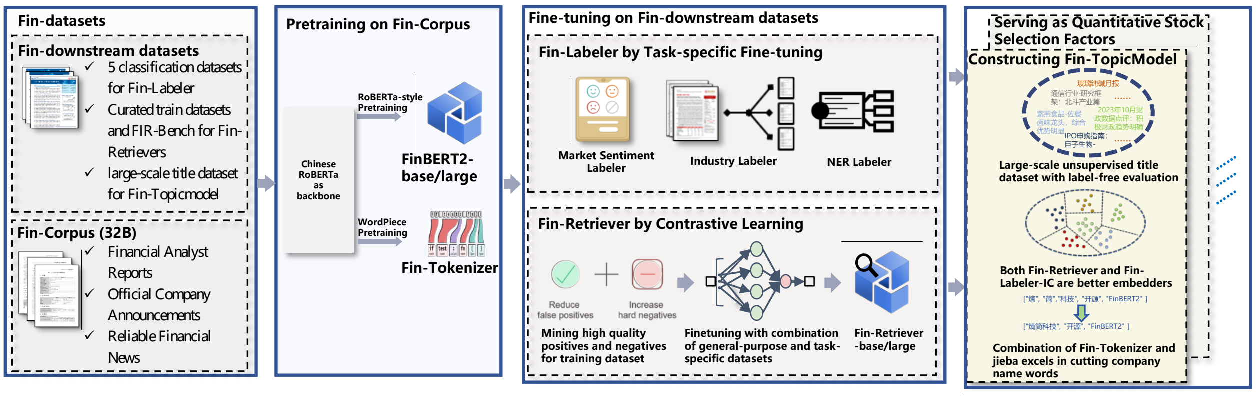

Figure 2: Overview of our FinBERT2 work.

1) Analyst Reports Corpus (16B tokens, 53G): We have compiled a collection of 2.6 million financial analysis reports, encompassing over twenty types of reports such as stock information, futures research, industry analysis and institutional commentaries. The dataset spans from the past 15 years and has undergone detailed data cleansing.

2) Company Announcements Corpus (6.4B tokens, 19G): Sourced from web-scraped announcements on the official websites of domestic listed companies, this dataset includes a wide array of corporate disclosures from various industries, such as financial reports, significant event statements, notices of shareholders' meetings, stock repurchase plans, and executive changes. It spans a period of 20 years and has been standardized in format to align with the report data format.

3) Duxiaoman [43] Open-Source News FinCorpus (9.6B tokens, 27G): Comprising articles and information aggregated from multiple sources including major financial news websites and social media, this dataset offers a comprehensive collection of financial news and insights.It spans a period of 20 years and has been standardized in format to align with the report data format.

2.2.2 Filtering Low-quality Fin-Corpus. Our 32B financial corpus contains noisy data, including URLs, verbose content, and incoherent or repetitive text. Due to the high cost of using LLMs for full dataset filter, we distill LLM's ability to judge quality into a lightweight BERT. We use Qwen2.5-72B-Instruct to rate a 100K-token subset on a 1–10 scale, labeling data above 8 as high-quality and below 4 as low-quality. This yields a 4K-instance training set (2K per class, 90%–10% train-test split). Fine-tuning RoBERTa-wwm-Chinese on this dataset produces a classifier with over 99% accuracy, which we use to filter the full corpus, removing 15% of low-quality data.

2.2.3 Pre-training Details. chinese-roberta-wwm-ext [6] was used as the initial backbone for FinBERT2 pre-training. Specifically, the training corpus is sliced into the longest contiguous segments of no more than 512 tokens. Strategies [21] such as Dynamic masking, pre-training using Whole Word Masking without Next Sentence Prediction (NSP) are also employed, which help minimize training loss and achieve lower bits-per-character (BPC) on the held-out financial corpus . We train with mixed-precision and AdamW [22] weight decay optimizer on 8 Nvidia A100 40G GPU machines, with the training code implementated by huggingface's transformer library.

2.2.4 Expanded Vocabulary for Fin-Tokenizer. Using the WordPiece algorithm [31], we extract domain-specific vocabulary from our 32B-token financial corpus and the C4 dataset, the latter being part of the original pre-training data for Chinese-RoBERTa. This process expands the model's vocabulary by 14,000 words, incorporating a substantial number of high-frequency financial terms and company names (e.g., BYD). Building on Fin-Tokenizer, we conduct post-pre-training on our financial corpus, yielding FinBERT2, a model better adapted to domain-specific tasks.

2.3 Task-specific Fine-tuning of Fin-Laberers

2.3.1 Five Downsteam Datasets for Fin-Labelers. Due to the disparity between existing public Chinese financial datasets and real-world business practices, we constructed five financial classification datasets by directly extracting and annotating data from financial terminal systems. This dataset encompasses three financial application scenarios, including:

1) Report-related industry classification (IC): Classify Report-related passage according to the China International Trust and Investment Corporation (CITIC) Level 1 industry classification, covering 28 industry categories.

2) Market Sentiment Classification (MSC): This task aims to classify the sentiment of textual commentary related to financial events or assets, facilitating market sentiment analysis and stock correlation studies. The first type of sentiment classification, applied to reports, includes four categories that represent varying levels of sentiment polarity and intensity. The second type, focused on news sentiment classification, consists of two categories: positive and negative.

3) Named Entity Recognition (NER) in Finance: Recognize and extract entities (e.g., company or personal names) appearing in the financial reports.

2.3.2 Fine-tuning Details. Fine-tuning in BERT usually optimizes all parameters to maximize classification probability. For

Task	Labels	Train/Test Samples	Lr(e-5)	Bs	Epochs	Metric
IC	28	4000/400	5	5	1	weighted-f1
MSC(4 labels)	4	1280/400	5	5	1	weighted-f1
MSC(2 labels)	2	4000/400	5	5	1	weighted-f1
NER(person)	3	664/140	5	4	5	recall
NER(company)	3	2883/300	5	4	5	recall

Table 1: Details of five classification tasks for Fin-Labelers .

Test Dataset Name	Query count	Doc count	Avg Query Words	Avg Doc Words
Sin-Doc-FinQA	114	1626	29.2	1656.3
Multi-Docs-FinQA	54	9384	16.6	589.8
Research Reports	228	1975	28.7	1405.7
Announcements	67	77429	40.2	1311.3
Indicators	200	182695	27.9	31.8

Table 2: statistics of FIR-Bench used to evaluate Fin-Retriever.

sequence classification tasks like MSC and IC, the [CLS] token's embedding (h) is fed into a fully connected layer to predict class label probabilities. For token classification tasks like NER, each token's corresponding vector is used for prediction. In our experiments, we employed a fine-tuning configuration as detailed in Table 1. We fine-tuned on 5 Downsteam Datasets for Fin-Labelers with an AdamW weight decay optimizer on a NVIDIA A100 40GB GPU to ensure consistency and comparability of the experiments. For the sequence classification task, we set the following parameters: epoch of 1, batch size of 5 , learning rate of 5e-5. For the token classification task, the same learning rate are maintained, the epoch is increased to 5.

2.4 Contrastive Learning for Fin-Retriever

We fine-tune FinBERT2 using contrastive learning on 64k financial QA samples and 150k retrieval data, carefully curated with balanced positive and negative examples for optimal performance. Evaluation covers five financial datasets and the C-MTEB benchmark, assessed using recall and nDCG@10.

2.4.1 Constructing a High-Quality Training Dataset. To enhance both financial and general capability of our model, we fine-tune the pre-trained FinBERT2 on a combination of general-purpose and finance-specific datasets. The training data consists of 64k financial QA data, 100k T2Retrieval data, 40k MMarcoRetrieval data, and 10k DuRetrieval data.Additionally, We have also optimized the setup and enhanced the quality of positive and negative examples in the training dataset.

1) Generating High-Quality Negative Samples: Since the original datasets lack negative samples, we employ the pre-trained model to mine 50 hard-negative samples for each query. To ensure these negative samples do not contain information that could potentially answer the query, we utilize the LLM (Qwen2.5-72b-Instruct) for further filtering. Additionally, financial data is incorporated into the negative sample pool to enhance the model's generalization ability and enrich dataset diversity.

2) Balancing Positive and Negative Samples: Achieving a balance between positive and negative samples is critical for optimal performance. After extensive experimentation, we found that a ratio of 2 positive instances to 8 negative instances delivers the best results. Increasing the number of negatives to 15 maintains comparable performance, while reducing it to 5 results in a performance decline. For the RoBERTa-large model, which has greater capacity than the RoBERTa-base model, the negative sample size per query is increased from 10 to 15 to better utilize its capacity.

3) Improving Positive Sample Quality and Dataset Integration: In the T2Retrieval, MMarcoRetrieval, and DuRetrieval datasets, positive instances may not always directly answer the query. To address this, we use the LLM (Qwen2.5-72b-Instruct) to filter and ensure that positive instances contain sufficient information to adequately respond to the query. This step improves the quality of positive samples. By combining high-quality positive and negative samples across datasets, we create a robust and effective training dataset.

2.4.2 Contrastive Learning Details. We train Fin-Retriever based on FinBERT2 using the widely adopted InfoNCE loss [29]. We set the following fine-tuning parameters: epoch of 3, batch size of 512, learning rate of 5e-5, temperature of 0.1, warm up ratio of 0.1 and weight decay of 0.01. We train the model on 4 NVIDIA A100 40GB GPU. For long document retrieval, we utilize a sliding window approach, where each window produces an embedding vector per sliding step, with a slight overlap between adjacent windows to maintain semantic coherence. A window size of approximately 400 words and an overlap of 20 words deliver optimal performance.

2.4.3 Financial Information Retrieval Benchmark (FIR-Bench). To comprehensively assess its domain-specific ability, we curated five financial retrieval test datasets (FIR-Bench) derived from our business data, ensuring a thorough evaluation of its performance in the financial domain. The statistics of these five test datasets are presented in Table 2.

1) Single-Document Financial Question-Answer Test Dataset (Sin-Doc FinQA): This dataset consists of queries linked to both positive and negative document instances. For each query, the candidate documents originate from the same article, with the number of positive documents ranging from a minimum of 1 to a maximum of 10. On average, each query is associated with 8.4 documents, of which 2.6 are positive.

2) Multi-Documents Financial Question-Answer Test Dataset (Multi-Docs FinQA): Unlike Sin-Doc FinQA, the Multi-Docs FinQA has document instances that come from different articles, thus it contains a much larger number of both positive and negative documents. The maximum number of positive documents is limited to 50. In the corpus, each question is associated with an average of 9,384 documents, of which 14 are labeled as positive on average.

3) Financial Retrieval Datasets from Three Sources: This includes three datasets—financial research reports, indicators, and announcements. We use the cosine similarity metric to calculate the recall rate of positive instances for each dataset.

2.4.4 General-domain Retrieval Test. Besides domain-specific retrieval evaluation, We also evaluate general-purpose retrieval capabilities of the models. We utilized the subset of C-TMEB [41] to evaluate the model's capability in general-domain. Specifically, our models are evaluated on the following representative and influential datasets: T2Retrieval, CovidRetrieval, MMarcoRetrieval and DuRetrieval datasets. We also used nDCG@10 as the evaluation metric.

	Backbone	IC	MSC(4 labels)	MSC(2 labels)	NER(person)	NER(company)	Avg
LLMs	**Qwen2-72b-Instruct**	0.9250	0.4880	0.8850	0.9669	**0.8995**	0.8329
	GPT-4-turbo	0.8600	0.4750	0.8880	0.9315	0.8787	0.8066
	Claude-3.5-Sonnet	0.9030	0.5230	0.8650	**0.9957**	0.8683	0.8310
General BERTs	**BERT-base-chinese**	0.9166	0.8676	0.8840	0.9901	0.8269	0.8970
	Chinese-MacBERT-base	0.9128	0.8616	0.9422	0.9854	0.8324	0.9069
	Chinese-RoBERTa-wwm-ext	0.9196	0.8841	0.9424	0.9901	0.8158	0.9104
FinBERTs	**FinBERT1-base**	0.9294	0.9147	0.9453	0.9901	0.8481	0.9255
	Mengzi-BERT-base-fin	0.9083	0.8657	0.9498	0.9902	0.8324	0.9093
	FinBERT2-base (ours)	0.9398	**0.9249**	0.9546	0.9901	0.8378	0.9295
	FinBERT2–large (ours)	**0.9432**	0.9131	**0.9573**	0.9804	0.8514	**0.9291**

Table 3: Performance of FinBERT2 and baselines on fin-classification tasks.

2.5 Constructing a Pipeline for Fin-TopicModel

2.5.1 Overview of Fin-TopicModel. We implemented the Fin-TopicModel on top of Fin-Retriever and other FinBERT2-related components. It performs unsupervised clustering on the Fin-Retriever embeddings using the HDBSCAN algorithm to obtain multiple clusters (topics). For each cluster, it uses c-TF-IDF (Class-Based Term Frequency-Inverse Document Frequency) to measure the importance of words within the cluster. By analyzing the high-frequency vocabulary in each cluster, it automatically generates topic descriptions.A visual presentation of results can be found in the Appendix F.

2.5.2 Large-scale Unsupervised Title Dataset with Label-free Evaluation. A dataset of 56,540 titles of reports was created from 59,014 articles (2022–2024) with an average length of 27 characters. It supports label-free topic modeling. A comprehensive evaluation is composed of subjective scoring (e.g., coherence, conciseness, informativeness) using LLM, clustering metrics like Silhouette Coefficient and Calinski-Harabasz Index, additional metrics, including topic diversity and outlier rate, provide further insights. This framework enables robust, unsupervised exploration of topic modeling without labeled data.

2.5.3 Encoding from FinBERT2 Variants. In conventional practice, retrieval models are commonly used to extract embedding vectors from texts to support topic modeling tasks. For this purpose, we selected FinRetriever to encode document embeddings. At the same time, we intuitively believe that FinLabeler-IC (a fine-tuned industry classification model) enables BERT to further learn semantic information related to specific industry classification tasks during training. These fine-tuned embeddings capture fine-grained semantic features in Identify the industry, making them highly suitable for topic modeling tasks. Subsequent experiments confirmed that the two chosen models indeed demonstrated significant advantages in topic modeling.

2.5.4 Precise Words Cutting. We used Fin-Labeler-NER to extract 3,290 company-related named entities from titles, which were added to Jieba's custom dictionary to enhance segmentation. To compare Fin-Tokenizer's financial vocabulary with Jieba's default segmentation, we tested Jieba on 13,804 custom financial terms from Fin-Tokenizer, revealing 1,724 segmentation inconsistencies. Analysis showed 545 were two-character terms (mostly auxiliary words or adverbs), and 1,179 were four-character terms (mainly company names).

Unlike pre-trained models like Fin-Labeler-NER (e.g., BERT), Fin-Tokenizer uses WordPiece subword segmentation based on statistical methods. To combine the strengths of both, we created a merge tokenizer, which uses a greedy strategy to select the segmentation with the longest coverage, improving NER-based tokenization accuracy and coverage.

2.5.5 Pipeline Details. we implemented the Fin-TopicModel based on the BERTopic library [9], which is an unsupervised topic modeling library for short texts. First, a pre-trained embedding model from SentenceTransformer was used to generate semantic vectors of size (56,540, 768) for the dataset titles. These embeddings were then reduced to 32 dimensions using UMAP, with parameters set to n_neighbors=15 and min_dist=0.0. Next, HDBSCAN was applied for density-based clustering, with min_cluster_size=2 and min_samples=1 to minimize outliers. A custom stopwords list based on the stopwords_cn.txt and the NERcom-Enhanced Tokenizer were used to instantiate a CountVectorizer for text vectorization. This facilitated the use of c-TF-IDF (Class-based Term Frequency-Inverse Document Frequency) to generate a list of keywords for each topic, ranked by their importance, as a descriptive representation of the topics.

3 EVALUATIONS AND ANALYSIS

3.1 Fin-Labelers

We benchmarked FinBERT2 against general-domain BERT models (e.g., BERT-base-Chinese, MacBERT, RoBERTa) and financial-domain pre-trained models (e.g., FinBERT1 and Mengzi-Fin). Table 1 provides a detailed summary of the number of labels, the sizes of training/testing samples, fine-tuning hyperparameters (learning rate, batch size, and number of epochs), and evaluation metrics for each task. A portion of downstream task datasets was reserved for evaluating Fin-Labelers, with testing conducted during training, and the best test results were recorded. All BERT models are same fine-tuned.

We also benchmark leading Large Language Model (LLM) APIs, including Qwen2-72b-Instruct, GPT-4-turbo, and Claude-3.5-Sonnet, against our suite of financial classification tasks. For each task, we meticulously crafted prompts, employing popular techniques such

DR Model	Sin-Doc-FinQA			Multi-Docs-FinQA		Research Reports		Announcements		Indicators	
	R@1	R@3	R@5	R@20	R@50	R@10	R@20	R@10	R@20	R@5	R@10
BGE-base-zh	0.479	0.815	0.906	0.238	0.318	0.921	0.960	0.387	0.482	0.910	0.930
BCE-embedding-base	0.513	0.824	0.902	0.227	0.309	0.967	0.978	0.318	0.421	0.803	0.915
text-embedding-3-small	0.511	0.823	0.906	0.197	0.234	0.864	0.872	0.473	0.509	0.863	0.928
text-embedding-3-large	**0.560**	0.845	0.920	0.215	0.257	0.951	0.960	0.492	0.526	0.940	0.965
Fin-Retriever-base (Ours)	0.520	0.846	0.916	0.307	0.398	**0.987**	**0.991**	0.566	0.642	0.950	**0.975**
Fin-Retriever-large (Ours)	0.554	**0.867**	**0.937**	**0.315**	**0.402**	0.983	0.987	**0.571**	**0.664**	**0.960**	0.970

Table 4: Performance comparison of Fin-Retriever with other dense retrievers (DRs) on 5 financial retrieval tasks.

DRmodel	T2Retrieval	CovidRetrieval	MMarcoRetrieval	DuRetrieval	Avg
BGE-base-zh	0.832	**0.799**	**0.634**	**0.829**	0.774
BCE-embedding-base	0.846	0.756	0.634	0.810	0.762
text-embedding-3-small	0.825	0.706	0.603	0.769	0.726
text-embedding-3-large	**0.906**	0.760	0.636	0.822	**0.781**
Fin-Retriever-base	0.847	0.779	0.603	0.776	0.751
Fin-Retriever-large	0.848	0.772	0.603	0.783	0.752

Table 5: Performance comparison of Fin-Retriever with other dense retrievers (DRs) on 4 general retrieval benchmark datasets from C-MTEB We use NDCG@10 as the evaluation metric.

as role specification, few-shot learning, and chain-of-thought to optimize LLM performance. To ensure fairness and accuracy in the evaluation, we applied identical prompts across the three powerful LLMs and performed three tests for each sample to obtain three responses. The final predicted label was determined by applying a majority voting rule to the acquired labels. The specific prompts are shown in the Appendix B.

3.1.1 Compared with LLMs. As shown in Table 2, the average F1 scores of Qwen2-72b-Instruct (0.8329), GPT-4-turbo (0.8066), and Claude-3.5-Sonnet (0.8310) are significantly lower than FinBERT2-base (0.9295) and FinBERT2-large (0.9291). On the challenging market sentiment classification task, LLMs scored below 0.523, far behind FinBERT2-base (0.9249). For NER tasks, LLMs showed superiority—e.g., Claude-3.5-Sonnet excelled in company names (0.868). These findings highlight the limitations of LLMs on domain-specific tasks without fine-tuning and reinforce FinBERT2's superiority in both effectiveness and efficiency for financial applications.

3.1.2 Compared with Other BERTs. General-purpose BERT models achieve average scores of 0.8970–0.9104 across five tasks, while FinBERT2-base and FinBERT2-large outperform them with scores of 0.9295 and 0.9291. For the complex four-class market sentiment classification (MSC) task, FinBERT2-base achieves 0.9249, compared to 0.8841 from general-purpose models, highlighting its ability to capture domain-specific nuances. Compared to financial-domain models like FinBERT-Chinese and Mengzi-BERT-base-fin, FinBERT2 also performs better. For example, in the IC task, FinBERT2-base scores 0.9398, surpassing FinBERT-Chinese (0.9294) and Mengzi-BERT-base-fin (0.9083). While simpler tasks show comparable performance, FinBERT2 demonstrates clear superiority in complex tasks such as four-class MSC, underscoring its domain-specific advantages.

3.1.3 Analysis about Tasks and Architecture. The performance of LLMs and BERT-based models varies significantly depending on the complexity of the NER task. In the context of simple NER tasks, such as person name recognition, some large language models (LLMs) may exhibit entity omission issues, resulting in performance inferior to that of fine-tuned BERT models. However, in complex NER tasks, such as company name recognition, LLMs are better able to demonstrate stronger generalization capabilities. Furthermore, BERT-Large models outperform BERT-Base models on complex NER tasks. This indicates that increased model scale contributes to improved generalization for complex NER tasks, potentially due to the capture of richer contextual information and semantic relationships.

Sentiment classification presents a distinct challenge for LLMs due to its reliance on industry-specific annotation criteria. The labeling criteria are typically determined by professional analysts and exhibit significant industry-specific characteristics. For instance, in traditional industries, 10% growth might be considered highly positive, whereas in another industry, it might only be viewed as moderately positive. The in-context learning capabilities of LLMs are often limited to surface-level semantics and numerical values, which hinders their adaptation to tasks with such nuanced, industry-specific criteria,which is difficult to be expressed by prompts.

3.2 Fin-Retriever

We compare our Fin-Retriever with popular open-source models of similar parameter sizes (BAAI General Embedding (BGE), Bidirectional Contrastive Embedding (BCE)) as well as OpenAI's proprietary models[4].

3.2.1 Performance on FIR-Bench. As shown in Table 4 both Fin-Retriever-base and Fin-Retriever-large outperform general-purpose

[4]https://platform.openai.com/docs/guides/embeddings

Embedder in pipeline	LLM-score(0-3)			Cluster Qulity Metrics		
	Coherence	Conciseness	Informativity	Silhouette Coefficient	Calinski Harabasz Score	Davies Bouldin Score
BGE-base-zh	1.765	1.550	1.865	0.141	11.394	1.254
BCE-embedding-base	1.790	1.460	1.870	0.171	12.483	1.158
text-embedding-3-small	1.744	1.533	1.809	0.106	11.204	1.274
text-embedding-3-large	1.795	1.445	1.825	0.182	12.934	1.155
Fin-Labeler-IC	1.830	1.500	1.845	0.170	11.108	1.108
Fin-Retriever-base	**1.835**	1.515	**1.905**	**0.192**	**13.296**	1.077
Fin-Retriever-large	1.760	**1.570**	1.860	0.174	12.690	**1.070**

Table 6: Performance comparison of different embedding models on the topic modeling task. The LLM scores reflect the semantic quality of the topics, while the clustering quality metrics measure the compactness and separation of the topics.

Embedder in pipeline	TD	Outliers Rate	Topic Count	Avg Docs/ Topic	SD Docs/Topic
BGE-base-zh	0.211	0.230	11872	3.563	3.125
BCE-embedding-base	0.217	0.253	12129	3.667	3.281
text-embedding-3-small	0.209	0.246	12260	3.477	2.962
text-embedding-3-large	**0.218**	0.224	11820	3.712	3.353
Fin-labeler-IC	0.196	**0.222**	**13417**	**3.280**	**2.144**
Fin-Retriever-base	**0.218**	**0.222**	11796	3.728	3.211
Fin-Retriever-large	0.213	0.227	12008	3.637	3.178

Table 7: Summarization of the statistical properties of the topics generated in the topic modeling task.

retrievers in FIR-Bench, demonstrating their strong domain specialization. Fin-Retriever-large consistently achieves the highest recall scores, with an average R@k of 0.746, while Fin-Retriever-base also surpasses most baselines, achieving 0.730 on average. Compared to the best-performing general model text-embedding-ada-002-large (0.723 avg), both Fin-Retriever variants exhibit superior financial retrieval capabilities. Particularly in Research Reports (R@10 = 0.987 for Fin-Retriever-large vs. 0.951 for text-embedding-ada-002-large) and Announcements (R@10 = 0.571 for Fin-Retriever-large vs. 0.492 for text-embedding-ada-002-large), the Fin-Retriever models demonstrate a significant advantage. This suggests that in financial domains where precision is critical, Fin-Retriever provides a substantial improvement over general-purpose dense retrievers.

3.2.2 Performance on General Retrieval Datasets. Despite being optimized for financial retrieval, Fin-Retriever-base and Fin-Retriever-large remain competitive in general retrieval tasks, as reflected in Table 5. Fin-Retriever-large achieves an average NDCG@10 score of 0.752, while Fin-Retriever-base follows closely with 0.751, both outperforming text-embedding-3-small (0.726) and approaching the performance of text-embedding-3-large (0.781), an advanced model from OpenAI's text-embedding series, with a parameter count significantly exceeding that of BERT.

3.3 Fin-TopicModel

3.3.1 Metrics. Evaluating topic models is a complex and evolving challenge. We introduce a suite of unsupervised evaluation metrics for Fin-TopicModel, thus circumventing the need for labeled data. The metrics fall into three broad categories. First, subjective scoring leverages LLMs to evaluate aspects such as coherence, conciseness, and informativeness. Second, clustering-based measures, including the Silhouette Coefficient and Calinski-Harabasz Index, are employed. Third, supplementary metrics like topic diversity and outlier rate are incorporated to provide a more nuanced understanding. For a comprehensive description of the metrics, refer to Appendix D.

3.3.2 Analysis of Fin-TopicModel based on Fin-Retrievers. Topic model based on Fin-Retriever-base demonstrates all-around superior performance across both subjective scoring (coherence, conciseness, and informativeness) and clustering quality metrics. It achieves the highest coherence score (1.835) and excels in clustering compactness and separation, as reflected by the highest Calinski-Harabasz Index (13.296) and a low Davies-Bouldin Score (1.077). These results highlight the robust and balanced capabilities of Fin-Retriever-base in topic modeling tasks, where it effectively captures semantic quality while maintaining well-formed clusters. In addition, it achieves a high topic diversity score (0.218) while maintaining the lowest outlier rate (0.222), By contrast, Fin-Retriever-large also performs well but falls slightly short of Fin-Retriever-base. This performance gap may be attributed to the increased model complexity of Fin-Retriever-large, which could require more hyper-parameter optimization to fully leverage its potential.

3.3.3 Analysis of TopicModel based on Fin-Labeler-IC. Topic model based on Fin-Labeler-IC showcases unique strengths and trade-offs in topic modeling. It generates the highest number of topics (13,417), significantly more than other models, which suggests its potential to capture granular distinctions in the data. Additionally, it achieves competitive semantic quality and clustering quality, with coherence (1.830) and informativeness (1.845) scores. However, this granularity comes at the cost of lower topic diversity (0.196), indicating potential redundancy or over-segmentation of the topics. Despite its simpler fine-tuning approach, Fin-Labeler-IC is competent for Topic modeling by leveraging its industry-specific training objective, which is inherently aligned with topic relevance. This trade-off highlights the value of task-specific embeddings for applications where topic relevance is paramount.

3.4 Ablations and Discussions

We conducted a detailed ablation study to verify the impact of pre-training corpus size and vocabulary expansion on the final results. All ablation experiment results are presented in the Appendix C.

3.4.1 Ablations of Domain Pre-trained Data Volume. To demonstrate that increasing the volume of the domain pre-training corpus can enhance model capability, we trained the 7.3B/10.3B Finbert models with the same configuration on a subset of the corpus. We then evaluate these models on both classification and retrieval tasks.

Table 8 demonstrates the impact of pre-training data volume on classification tasks. The results indicate a clear performance improvement as the pre-training data size increases. Without domain-specific pre-training, the average F1 score is 0.9104. With 7.3B tokens, the score increases to 0.9189, and with 32B tokens (FinBERT2-base), it reaches 0.9295, achieving the best results. Tasks like MSC and IC benefit significantly from larger pre-training datasets, showing notable performance gains. This highlights the crucial role of extensive pre-training in capturing domain-specific features and improving generalization in classification tasks.

Table 9 presents the effect of pre-training data size on retrieval tasks, evaluated using Recall@k. The results show that larger pre-training datasets lead to substantial performance improvements. Without domain-specific pre-training, the average Recall is 0.621. Pre-training on 16B and 32B tokens improves the average Recall to 0.659 and 0.686, respectively. Tasks such as "Announcements" and "Research Reports" benefit the most, with Recall increasing from 0.325 to 0.566 and 0.943 to 0.987, respectively. These results demonstrate that larger pre-training datasets enable the model to better understand domain-specific information and enhance retrieval performance.

3.4.2 Abalation of Vocabulary Expansion on Retrieval Performance. Table 10 presents a comparison of pre-training FinBERT2-base with and without vocabulary expansion. The results show that adding a domain-specific vocabulary (Fin-Tokenizer) significantly improves performance across retrieval tasks. For instance, the model with Fin-Tokenizer achieves R@50 of 0.397 on Multi-Docs-FinQA and R@20 of 0.642 on Announcements, representing increases of 0.034 and 0.155 compared to the model without Fin-Tokenizer. This highlights that vocabulary expansion effectively enhances the model's ability to understand and retrieve financial domain-specific terminology.

3.4.3 Is Advantage in Retrieval Tasks Attributable to Fine-Tuning Rather than the FinBERT2 Backbone? Although our model outperforms others on multiple financial retrieval tasks, this advantage may be partly attributed to the lack of fine-tuning of other models on the same financial data. To ensure a fair comparison, we fine-tuned the BGE-base-zh using the same dataset. As shown in Table 11, BGE-base-zh, which shares the same architecture as our Fin-Retriever-base, was fine-tuned with the same data and hyperparameters employed in the second stage of contrastive learning for Fin-Retriever. We subsequently evaluated both models on FIR-Bench. These results suggest that FinBERT2-base is well-suited for retrieval tasks, especially the Research Reports and Announcements retrieval tasks.

3.4.4 Is BERT-large a Better Backbone Than BERT-base? We conducted scale-based fine-tuning experiments on the RoBERTa model, utilizing both the RoBERTa-chinese-base and RoBERTa-chinese-large variants, and systematically compared their performance. While the results demonstrate that the RoBERTa-chinese-large model generally outperforms its base counterpart in both label tasks and retrieval tasks, the performance gains are not significant. Furthermore, training and fine-tuning the large model require extensive hyperparameter optimization and computational resources, posing a considerable challenge. Due to these factors, we did not dedicate substantial effort to further optimizing the large model. Despite the slight advantage shown by the large variant, the base model remains highly competitive and efficient, making it a strong choice for practical applications.

4 FUTURE WORK

The future-oriented work is as follows:

1) In pre-training task, we will perform domain adaptation based on optimized architecture BERT such as ModernBERT [38], or larger parameter BERT such as MegatronBert-1.3B [14].

2) In retrieval tasks, our training remains insufficient, as we have only utilized 200,000 data points so far. If more data were used for training, the model's performance could be further improved.

3) In topic modeling tasks, we have not explored many additional optimization techniques to enhance the modeling performance, such as strategies for reducing outliers and employing more diverse representations. If further optimizations were applied, the model might achieve better results.

4) In addition to topic modeling task, we aim to apply FinBERT2 in quantitative investment.The Fin-Labeler-MSC, fine-tuned on the market sentiment dataset, demonstrates exceptional proficiency in understanding complex financial market sentiment, making it a positively correlated factor with excess stock returns. The factor can be integrated into softprob multi-classification XGBoost models to establish a robust mapping between embedded vectors and real market behaviors.The approach has potential for selecting the topK stocks as a effective quantitative investment strategy.

5 CONCLUSION

In this work, we introduced FinBERT2, a specialized bidirectional encoder pre-trained on the largest known Chinese financial corpus (32B tokens). Our results demonstrate that encoder-only models still play a crucial role in financial NLP, complementing the strengths of decoder-only LLMs. Specifically, FinBERT2 achieves: (1) superior performance on discriminative financial classification tasks, outperforming existing (Fin)BERT variants and leading LLMs; (2) enhanced retrieval capabilities through Fin-Retrievers, surpassing both open-source and proprietary embedding models; and (3) improved topic modeling with Fin-TopicModel, yielding better clustering and topic representation. These findings highlight the continued relevance of encoder-based architectures in financial AI, particularly in scenarios requiring high precision and domain-specific understanding.

REFERENCES

[1] Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470* (2020).

[2] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models. https://arxiv.org/abs/1908.10063

[3] Gagan Bhatia, El Moatez Billah Nagoudi, Hasan Cavusoglu, and Muhammad Abdul-Mageed. 2024. FinTral: A Family of GPT-4 Level Multimodal Financial Large Language Models. https://arxiv.org/abs/2402.10986

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[5] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932* (2020).

[6] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3504–3514.

[7] Desola. 2019. *FinBERT: Pre-Trained Model on SEC Filings for Financial Natural Language Tasks*. https://doi.org/10.13140/RG.2.2.19153.89442

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[9] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794 [cs.CL] https://arxiv.org/abs/2203.05794

[10] Panpan Hou, Mengchao Zhang, Zhibing Fu, and Yu Li. 2020. FinBERT. https://github.com/valuesimplex/FinBERT.

[11] Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad Actor, Good Advisor: Exploring the Role of Large Language Models in Fake News Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 20 (2024), 22105–22113. https://doi.org/10.1609/aaai.v38i20.30214 arXiv:2309.12247 [cs]

[12] Huang. 2023. FinBERT: A Large Language Model for Extracting Information from Financial Text*. *Contemporary Accounting Research* 40, 2 (2023), 806–841. https://doi.org/10.1111/1911-3846.12832

[13] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[14] IDEA-CCNL. 2021. Fengshenbang-LM. https://github.com/IDEA-CCNL/Fengshenbang-LM.

[15] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[16] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. ChatGPT: Jack of All Trades, Master of None. *Information Fusion* 99 (2023), 101861. https://doi.org/10.1016/j.inffus.2023.101861 arXiv:2302.10724 [cs]

[17] Thanos Konstantinidis, Giorgos Iacovides, Mingxue Xu, Tony G. Constantinides, and Danilo Mandic. 2024. FinLlama: Financial Sentiment Classification for Algorithmic Trading Applications. https://arxiv.org/abs/2403.12285

[18] G Lample. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* (2019).

[19] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* (2019).

[22] Ilya Loshchilov, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101* (2017).

[23] Guangyuan Ma, Xing Wu, Zijia Lin, and Songlin Hu. 2024. Drop your Decoder: Pre-training with Bag-of-Word Prediction for Dense Passage Retrieval.. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1818–1827.

[24] Vitaly Meursault, Pierre Jinghong Liang, Bryan R Routledge, and Madeline Marco Scanlon. 2023. PEAD. txt: Post-Earnings-Announcement Drift Using Text. *Journal of Financial and Quantitative Analysis* 58, 6 (2023), 2299–2326.

[25] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2014–2037. https://aclanthology.org/2023.eacl-main.148

[26] Pandu Nayak. 2019. Understanding searches better than ever before. https://blog.google/products/search/search-language-understanding-bert/ Accessed: 2025-01-05.

[27] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005* (2022).

[28] Inc. NetEase Youdao. 2023. BCEmbedding: Bilingual and Crosslingual Embedding for RAG. https://github.com/netease-youdao/BCEmbedding.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.

[31] Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5149–5152.

[32] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. https://arxiv.org/abs/2212.09741

[33] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[34] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-Training. https://arxiv.org/abs/2212.03533

[35] Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. FinGPT: Instruction Tuning Benchmark for Open-Source Large Language Models in Financial Datasets. https://arxiv.org/abs/2310.04793

[36] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective works best for zero-shot generalization?. In *International Conference on Machine Learning*. PMLR, 22964–22984.

[37] Yuxin Wang, Qingxuan Sun, and Sicheng He. 2023. *M3E: Moka Massive Mixed Embedding Model*. Moka Massive Mixed Embedding.

[38] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663* (2024).

[39] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. https://arxiv.org/abs/2303.17564

[40] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 538–548. https://aclanthology.org/2022.emnlp-main.35

[41] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. https://arxiv.org/abs/2309.07597

[42] Biao Zhang, Behrooz Ghorbani, Ankur Bapna, Yong Cheng, Xavier Garcia, Jonathan Shen, and Orhan Firat. 2022. Examining scaling and transfer of language model architectures for machine translation. In *International Conference on Machine Learning*. PMLR, 26176–26192.

[43] Xuanyu Zhang and Qing Yang. 2023. Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 4435–4439.

[44] Zhuosheng Zhang, Hanqing Zhang, Keming Chen, Yuhang Guo, Jingyun Hua, Yulong Wang, and Ming Zhou. 2021. Mengzi: Towards Lightweight yet Ingenious Pre-Trained Models for Chinese. https://arxiv.org/abs/2110.06696

A RELATED WORKS

A.1 FinLMs

Since 2019, specialised models have emerged to tackle the complexities of financial text. FinBERTs (Araci [2], Desola [7], and Huang [12]), have demonstrated proficiency respectively in financial sentiment analysis and document processing and information extraction from financial texts. In the Chinese community, Hou et al. [10]'s FinBERT was open-sourced, and its F1 scores in different financial tasks were significantly improved compared to the vanilla BERT. Another influential work Mengzi-fin (Zhang et al. [44]) further enriching the field of Chinese FinBERTs. In parallel, Larger LMs have ventured into the financial domain, tailoring their capabilities to specific financial tasks. BloombergGPT (Wu et al. [39]) are characterised by 50B-parameter model trained on a 363B token financial corpus The FinTral suite by Bhatia et al. [3], based on Mistral-7b and multimodal data, outperformed GPT-4 on several tasks including Sentiment Analysis (SA), Named Entity Recognition (NER) etc. FinLlama (Konstantinidis et al. [17]), derived from Llama-2-7b, improved sentiment classification accuracy and quantified sentiment strength. Both FinBERTs and FinLLMs highlight the advantages of in-domain pre-training. In these works, although FinLLMs has greater potential and application scenarios, relatively small specialized model trained on more financial corpus are more parameter-efficient that either match or outperform much larger language models.

A.2 Dense Retrievers

Dense retrievers (DRs), also called text embedders typically use a BERT-based dual encoder architecture that learns by minimizing the similarity between documents and queries. This independence of document representation from query representation accommodates the offline computational demands of large corpora. pre-training tasks designed specifically for retrieval (passage ranking), such as ICT [19], BFS [5] and RetroMAE [40] are proved effective, but it has also been shown that carefully fine-tuning a vanilla BERT model can also outperforms these approaches [15, 23] Recent studies have explored the construction of generalized text representation models through large-scale contrastive learning [27, 32, 34]. These works mostly follow a multi-stage training approach: i.e., pre-training on large-scale weakly supervised text pairs on a in-batch-negative manner, and supervised fine-tuning on triplets with hard hegatives to further fit to popular benchmarks [25, 33, 41]. Despite extensive training, out-of-domain generalization remains limited. These DRs even fail to reach BM25 level without further fine-tuning on labeled datasets. As far as we know, no one tried to customize an Fin-DR based on a FinBERT pre-trained sufficiently on large scale financial corpus.

B PROMPTS OF DOWNSTREAM TASKS USING LLMS(CHINESE TO ENGLISH ALREADY)

B.1 IC Prompt

Role: Senior Industry Researcher. **Task:** According to the CITIC industry classification definition, classify short financial texts into primary CITIC industry categories.

The industry list and classification descriptions are as follows: Building Materials: Involves the production and sale of construction materials such as cement, glass, ceramics, and new building materials. Food and Beverages: Includes food manufacturing and beverage production, such as liquor, dairy products, and meat processing. Media: Covers industries such as broadcasting, television, publishing, internet media, and advertising. Non-Ferrous Metals: Includes mining, smelting, and processing of non-ferrous metals such as aluminum, copper, lead, and zinc. Computers: Involves the development, production, and sale of computer hardware and software. Non-Banking Finance: Includes non-banking financial institutions such as securities, insurance, and trusts. Pharmaceuticals: Covers pharmaceutical manufacturing, biotechnology, and medical devices. Commercial and Retail: Includes retail formats such as department stores, supermarkets, and specialty retail. Electricity and Utilities: Involves electricity production and supply, as well as public utilities such as water and gas. Steel: Includes steel smelting and rolling. Real Estate: Involves real estate development, sales, and rental. Machinery: Includes the manufacturing of various machinery and equipment, such as engineering machinery and general equipment. Agriculture, Forestry, Animal Husbandry, and Fishery: Covers industries such as agricultural planting, forestry, animal husbandry, and fishery. Basic Chemicals: Includes petrochemicals, fertilizers, pesticides, and coatings. National Defense and Military Industry: Covers the R&D and production of defense technology and military equipment. Banking: Includes financial institutions such as commercial banks and policy banks. Transportation: Includes transportation services such as road, rail, aviation, and waterway transport. Home Appliances: Involves the production and sale of household appliances such as refrigerators, washing machines, and air conditioners. Catering and Tourism: Includes catering services and tourism services. Construction: Involves construction, renovation, and decoration. Light Industry Manufacturing: Includes light industry products such as papermaking, packaging, and furniture manufacturing. Automobiles: Covers the production and sale of vehicles and auto parts. Textiles and Apparel: Includes the production and sale of textiles and clothing. Electronic Components: Involves electronic components, semiconductors, and optoelectronic products. Telecommunications: Includes telecommunications equipment and service provisioning. Power Equipment: Covers power generation equipment and transmission and transformation equipment. Petroleum and Petrochemicals: Includes oil exploration, refining, and petrochemical products. Coal: Covers coal mining, processing, and sales.

Input: text: Financial news brief

Result Format: Rationale:{CoT, no more than 100 words} Result:{Industry Name}

B.2 MSC(2 labels) Prompt

Role: Senior Financial Data Analyst, Senior Industry Researcher. **Task:** Financial sentiment classification: Aim to classify evaluative texts on financial events or items into sentiments to observe market sentiment.

The sentiment classification task includes two categories: 0: Negative 1: Positive

Result Format: Rationale:{CoT, no more than 100 words} Result:{Sentiment label, 0 or 1}

B.3 MSC(4 labels) Prompt

Role: Senior Financial Data Analyst, Senior Industry Researcher. **Task:** Financial sentiment classification: Aim to classify evaluative texts on financial events or items into sentiments to observe market sentiment.

The sentiment classification task includes four categories: Positive Sentiment (Label: 3): Texts typically include positive evaluations of company performance, stock recommendations, or optimistic industry outlooks. Examples include terms such as "recommend," "exceeds expectations," "significant growth," "strongly recommend," "high-quality," "leader," and "performance surge," indicating positive evaluations and optimistic expectations of companies or industries. Neutral-Positive Sentiment (Label: 2): Texts may include affirmations of company performance but also concerns or uncertainties about certain factors. Such texts often use terms like "in line with expectations," "stable," "neutral-positive," "slight growth," and "maintain," reflecting confidence in companies or industries but with some reservations. Neutral Sentiment (Label: 1): Texts provide objective descriptions of companies or industries without obvious positive or negative sentiment tendencies. Terms like "neutral," "stable," "flat," and "basically in line with expectations" indicate neutral views on companies or industries. Negative Sentiment (Label: 0): Texts typically contain concerns about company performance, non-recommendations of stocks, or pessimistic industry outlooks. Examples include terms like "decline," "losses," "risks," "below expectations," "reduction," and "negative growth," reflecting negative evaluations and pessimistic expectations of companies or industries.

Result Format: Rationale:{CoT, no more than 100 words} Result:{Sentiment label, one of 0-3}

B.4 NER(company) Prompt

Role: Senior Data Annotation Engineer. **Task:** NER (Named Entity Recognition): Identify and extract company entities mentioned in financial texts.

Example: Question: Text to extract company entities from: Everbright Pharmaceutical and Fuxiang Co., Ltd. Analysis: Antibiotic Upgrades by “Water Sellers,” Simultaneous Growth of Volume and Profit. The company significantly benefits from stable demand growth and industry supply contraction, with performance approaching a turning point. We predict earnings for 1921 to be 1.20, 1.52, and 1.86 yuan, respectively, with year-on-year growth rates of 67%, 27%, and 22%. The current price corresponds to 15x, 12x, and 10x P/E for 1921. Health is essential for the heart and mind. Answer: Rationale: ... Result:['Everbright Pharmaceutical','Fuxiang Co., Ltd.']

Result Format: Rationale:{CoT, no more than 100 words} Result:{List of company entities}

B.5 NER(person) Prompt

Role: Senior Data Annotation Engineer, Entity Extraction Engineer. **Task:** NER (Named Entity Recognition): Identify and extract name entities mentioned in financial texts.

Example: Question: Text to extract name entities from: 2019 Q3 Report Analysis: Overall Slight Improvement, Significant Recovery in Small and Mid-Cap Stocks. Marginal improvement in profitability for electronics, media, and telecommunications, as well as enhanced profitability in finance, home appliances, construction, and public utilities. Food and beverages remain stable. Wang Yang is a key contributor. Dongwu Strategy, Dongwu Machinery: Chen Xianfan 18616532999. Answer: Rationale: ... Result:['Wang Yang','Chen Xianfan']

Result Format: Rationale:{CoT, no more than 100 words} Result:{List of name entities}

C ABLATIONS TABLES

Backbone for task-specific fine-tuning	IC	MSC(4 labels)	MSC(2 labels)	NER(person)	NER(company)	Avg
w/o domain-pre-trained (Chinese-RoBERTa)	0.9196	0.8841	0.9424	0.9901	0.8158	0.9104
w/ 7.3B domain-pre-trained	0.9241	0.9044	0.9424	**0.9902**	0.8333	0.9189
w/ 10.3B domain-pre-trained	0.9252	0.8975	0.9499	**0.9902**	0.8278	0.9181
w/ 16B domain-pre-trained	**0.9437**	0.9176	0.9525	**0.9902**	0.8333	0.9275
w/ 32B domain-pre-trained (our FinBERT2-base)	0.9398	**0.9249**	**0.9546**	0.9901	**0.8378**	**0.9295**

Table 8: Ablation experiments about pre-training strategy and data volume on different backbones following the same task-specific fine-tuning procedure. w * means with *,w/o * means without *.

Backbone for contrast fine-tuning	Sin-Doc-FinQA R@1	Sin-Doc-FinQA R@3	Sin-Doc-FinQA R@5	Multi-Docs-FinQA R@20	Multi-Docs-FinQA R@50	Research Reports R@10	Research Reports R@20	Announcements R@10	Announcements R@20	Avg
w/o domain-pre-trained (Chinese-RoBERTa)	**0.534**	0.837	0.911	0.291	0.376	0.943	0.956	0.325	0.417	0.621
w/ 7.3B domain-pre-trained	0.507	0.838	0.915	0.292	0.384	0.934	0.947	0.412	0.445	0.630
w/ 10.3B domain-pre-trained	0.522	0.819	**0.920**	0.296	0.388	0.965	0.969	0.390	0.448	0.635
w/ 16B domain-pre-trained	0.514	0.844	0.916	0.305	0.392	0.965	0.982	0.462	0.549	0.659
w/ 32B domain-pre-trained (our FinBERT2-base)	0.520	**0.846**	0.916	**0.307**	**0.398**	**0.987**	**0.991**	**0.566**	**0.642**	**0.686**

Table 9: Ablation experiments about data volume of domain-pre-train on different backbones after contrast fine-tuning. w * means with *,w/o * means without *. We use Recall@k as the metric.

FinBERT2-base pre-trained Configuration	Sin-Doc-FinQA			Multi-Docs-FinQA		Research Reports		Announcements	
	R@1	R@3	R@5	R@20	R@50	R@10	R@20	R@10	R@20
w/o vocabulary expansion	**0.521**	0.843	**0.917**	**0.279**	0.363	0.896	0.938	0.408	0.487
w/ vocabulary expansion (Fin-Tokenizer)	0.520	**0.846**	0.916	0.307	**0.398**	**0.987**	**0.991**	**0.566**	**0.642**

Table 10: Ablation experiments about tokenizer w/wo vocabulary expansion

Backbone for contrast fine-tuning	Sin-Doc-FinQA			Multi-Docs-FinQA		Research Reports		Announcements	
	R@1	R@3	R@5	R@20	R@50	R@10	R@20	R@10	R@20
BGE-base-zh	0.519	0.843	**0.920**	**0.320**	0.389	0.965	0.978	0.378	0.442
FinBERT2-base	**0.520**	**0.846**	0.916	0.307	**0.398**	**0.987**	**0.991**	**0.566**	**0.642**

Table 11: Performance of BGE-base-zh and FinBERT2-base after fin-retrieval fine-tuned by the same training dataset and fine-tuning processes. We use Recall@k as the metric.

D METRICS OF FIN-TOPICMODEL

- **Subjective Evaluation:** We leverage a large language model (Qwen-max) to subjectively evaluate the topic descriptor lists. First, 200 topic descriptor lists are randomly sampled, and Qwen-max is prompted to score them on three aspects: **Coherence**, **Conciseness**, and **Informativity** (on a scale of 1 to 3). The average score across these three dimensions is calculated and used as the final metric.
- **Clustering Quality Evaluation:** We adopt three widely used metrics—**Silhouette Coefficient**, **Calinski-Harabasz Index**, and **Davies-Bouldin Index**—to evaluate the quality of the embeddings used for clustering. These metrics measure the compactness of data within clusters and the separation between clusters, providing a quantitative assessment of the clustering algorithm's effectiveness.
- **Topic Diversity (TD):** This metric calculates the proportion of unique words across topics. TD values range from 0 to 1, where higher values indicate greater topic diversity and more varied topics generated by the model.
- **Outlier Rate:** We also calculate the proportion of outliers (documents assigned a label of -1) relative to the total dataset size as an additional measure of embedding quality. By the way, the number of outliers can be reduced through post-prediction strategies or parameter adjustments.
- **Other statistics:** In addition, we summarize other statistical properties of the topics generated by different embedding models in the topic modeling task. These include the **outlier rate**, **total number of topics**, **average document count per topic**, and the **standard deviation of document counts per topic**, which provide further insights into the characteristics of the generated topics.

E PROMPTS FOR ASSESSING TOPIC DESCRIPTIVE WORDS (TRANSLATED FROM CHINESE)

Please evaluate the given topic keyword list based on the following standards for topic quality assessment. For each criterion, provide a score ranging from 1 to 3, along with a brief explanation of the score.

Topic Quality Assessment Criteria: 1. Coherence Definition: The keywords within a topic should be semantically related and collectively describe a topic or multiple closely related topics.

2. Conciseness Definition: A topic should not contain irrelevant or meaningless words, such as noise words or semantically redundant terms.

3. Informativity Definition: A topic should provide sufficient, specific, meaningful, or valuable information, covering different aspects of the same topic.

Evaluation Instructions: For the provided topic keyword list, rate each criterion on a scale of 1 to 3: 1 point: Poor performance, does not meet the standard. 2 points: Average performance, partially meets the standard. 3 points: Excellent performance, fully meets the standard. For each rating, provide a brief explanation to justify the score. Input:{Topic Keywords List}

Example Response Format: { "Topic Keyword List": ["strategy", "market", "investment", "risk", "return"], "Evaluation": {

"Coherence": { "Score": 3, "Explanation": "Keywords are closely related, all relevant to the field of financial investment, and collectively describe the theme of investment strategies." },

"Conciseness": { "Score": 3, "Explanation": "Keywords are clear, with no stopwords or meaningless terms, and no redundancy detected." },

"Informativity": { "Score": 2, "Explanation": "The topic only covers the main aspects of financial investment but lacks detailed descriptions of specific markets or investment tools." } } }

F VISUALIZATION OF FIN-TOPICMODEL

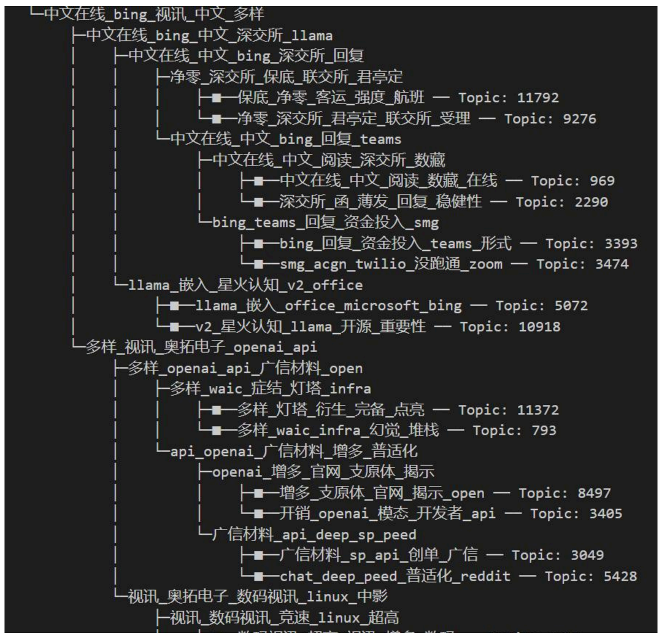

Figure 3: This figure illustrates the hierarchical structure of topic clustering in a tree diagram, where each node represents a topic or subtopic, and the branches reflect the hierarchical relationships between topics. The top-level nodes represent broader topics, while the child nodes provide further subdivisions of the parent topics, progressively narrowing down into more specific semantic domains. Each leaf node in the diagram corresponds to a unique topic identifier (Topic ID), which is used to identify specific topics.