Tailor-Made or Off-the-Rack? Comparing Domain-Specific and General-Domain Language Models on a Financial NLP Task

Stanford CS224N Custom Project

Irina Alexandra Marton
Department of Computer Science
Stanford University
imarton@stanford.edu

Abstract

Inspired by Araci's FiBERTAraci (2019) and motivated by the need to understand the effectiveness of domain adaptation, we evaluate 5 models on a Named Entity Recognition and Disambiguation (NER+NED) task that links company mentions in news headlines with their stock tickers. This study examines whether domain adaptation offers performance advantages, especially as model size increases, contributing to the discourse on the efficacy of domain-specific models in the context of large-scale language models. Experiments illustrate that prompt engineering is the only method to constantly be effective when performing domain adaptation. Also, an inverse relationship between model size and precision improvement, and a direct relationship between recall improvement and model size are observed.

1 Key Information to include

- Mentor: Kamyar Salahi

2 Introduction

There is a need to understand whether our current state-of-the-art language models are already experts in any specific domain we require them to be, or whether they can still benefit from domain adaptation. This inquiry is particularly pertinent given the increasing availability for the use of massive, generalized language models that may already perform optimally on domain-specific tasks due to their broad training. In this study we critically evaluate the effectiveness of domain-specific language models in the realm of a financial NLP task and address the following questions:

- What is the performance of FinBERT and other small and medium sized financial language models on financial NLP tasks compared with general-domain large language models such as ChatGPT-4o and Claude 3?
- To what extent does fine-tuning on a financial corpus improve model performance on a NLP task?
- Does the utility of domain adaptation decrease as the size of the language model increases?

The experiments challenge the popular belief that a model's performance is guaranteed to improve on a domain-specific task if the model has been fine-tuned to the domain in question. Instead, we find that the method of domain adaptation plays a crucial role in determining the effectiveness of the resulting model. During the conducted experiments, prompt engineering has been the only method to constantly bring improvements to a model when performing domain adaptation. ChatGPT-4o fine-tuned to the financial domain via prompt engineering proves to be the overall top performing model. In addition, for models benefiting from domain adaptation, we observe an inverse relationship between model size and precision improvement, while recall improvement exhibits a possible direct relationship with model size. Further research involving a broader range of models and tasks is necessary to draw definitive conclusions about this relationship.

3 Related Work

There have been several financial language models explored by various authors and institutions, as showcased by Lee et al. (2024) in their survey of LMs in finance. Among those mentioned in the paper, FinBERT by Araci (2019) represents a significant contribution to the NLP field as it was the first financial language model, while BloombergGPT by Wu et al. (2023) gained a lot of media attention for domain-specific LLMs. Even though FinBERT shows improved performance when compared to the general BERT model in financial text sentiment analysis, the paper does not explore other NLP tasks. BloombergGPT did outperform all the other general domain models it was benchmarked against, however GPT-NeoX and OPT do not represent the performance of current state-of-the-art models.

Today, influenced by these papers, there is a substantial quantity of resources invested in exploring financial LMs. This is exemplified by the plethora of such models made available through Huggingface Transformers by various users. Even though Araci (2019) and Wu et al. (2023) made a case for the need for domain-specific models, we pose the question if that need still exists. A similar question was explored by Peng et al. (2021) as they compare BERT and FinBERT on various financial tasks, and conclude that domain-specific pre-training from scratch does not appear to be considerably effective, while continual pre-training from the original model is the more beneficial option.

This study directly builds upon the insights provided by Araci (2019) and extends the work done by Peng et al. (2021) by evaluating BERT, Mistral, and LLaMA based financial LMs available on Huggingface in comparison to today's state-of-the-art LLMs such as ChatGPT-4o and Claude 3.

4 Approach

4.1 Methods

We employ a comparative experiment, where each base model and financial domain fine-tuned version of it will be evaluated on the NER+NED task using the evaluation split of the Key Developments Dataset with 0-shot prompts. The models that will be included in the experiment:

- *0,11B parameters*: **BERT**-base-uncased by Devlin et al. (2019) and the financial domain BERT-based models **FinBERT** by Araci (2019) and **FLANG-BERT** by Shah et al. (2022) will be fine-tuned for the NER+NED task with the training split of the Key Developments Dataset.
- *7B parameters*: Mistral-7B-instruct-v0.1 by Jiang et al. (2023) and the financial domain version of it, LongShort-Mistral-7B by BriefAI (2024). Throughout this paper, they will be referred to as **Mistral (Base)** and **Mistral Financial**.
- *13B parameters*: LLaMA-2-13B-chat by Meta (2023) and the financial domain LLaMA models FinanceConnect-13B by CeADAR (2023) and finance-LLM-13B by Cheng et al. (2024). Throughout this paper, they will be referred to as **LLaMA (Base)**, **LLaMA Financial** and **LLaMA Financial-v2**.
- *137B parameters*: **Claude 3** fine-tuned via prompt engineering for the financial domain
- *≈1.7T parameters*: **ChatGPT 4o** fine-tuned via prompt engineering for the financial domain

4.2 NER+NED task

It was pointed out by Shah et al. (2022) that certain tasks, such as domain-specific NER, do not require domain-specific knowledge, and therefore a domain-adapted model will not necessarily have improved performance when compared to the general domain version of it. The NER+NED task combines NER with named entity disambiguation for linking company entity mentions with their stock tickers and is a generative task that has been explored by the authors of BloombergGPT, Wu et al. (2023). The model is asked to output the tickers of all public company mentions in the input. The reason for selecting this task is that it evaluates not only a model's ability on the NER task, but also its knowledge of companies. Example:

INPUT: "AAPL announced they will stop using Intel Chips."
OUTPUT: "AAPL, INTC"

4.3 Prompt engineering

Findings by Chen et al. (2023) were used to formulate optimal 0-shot prompts to fine-tune Claude 3 and ChatGPT-4o:

Task adaptation 0-shot prompt

Extract tickers of companies mentioned in each of the following headlines and list for each headline all tickers extracted for that headline: `{HEADLINES}`

Domain adaptation prompt

You are an expert in finance, specialised in stock trading.

4.4 Baseline

In order to evaluate the benefits of domain adaptation, base versions of all the selected models will be used as baselines to assess the difference in performance between the general-domain and domain-specific versions. Since the described NER+NED financial task appears only in the BloombergGPT (50B parameters) paper, where the models were evaluated across multiple Bloomberg internal datasets, we do not have an established state-of-the-art performance. However, we can take note that in the BloomberGPT evaluation performed by Wu et al., where they utilised 20-shot prompts, the model had an average of 64.83 F1 score across all datasets.

5 Experiments

5.1 Data

The Capital IQ - Key Developments Dataset has been obtained through Wharton Research Data Services and contains 1450 headlines from Jan-2023 to Mar-2024 of 8 chosen public companies, gathered from 20,000 news sources including press releases, regulatory filings, company websites, web mining, investor Conference Organizer Websites, and call transcripts. The dataset was split 70%-30% between training and evaluation.

Company	Ticker	% Training Headlines	% Test Headlines
Accenture	ACN	16	9
Advanced Micro Devices	AMD	8	6
Adobe	ADBE	9	7
Alphabet	GOOGL	14	5
Amazon	AMZN	16	10
Microsoft	MSFT	11	25
Nvidia	NVDA	11	26
Oracle	ORCL	15	12

Table 1: Key Developments Dataset statistics

The original dataset has been augmented through human annotation to include tickers of all mentioned companies. For the training split only the original 8 were annotated, while for the test split, public companies not included in the original chosen 8 were also added. Example from test split:

- News headline example: "App Defense Alliance Migrates Under Joint Development Foundation with Google, Meta, and Microsoft as the Steering Committee"
- Original dataset label: MSFT
- Augmented dataset label: GOOGL, META, MSFT

After augmentation, the test set is comprised 66% of headlines that mention public companies not included in the original eight, 3% of headlines that mention more than one of the selected eight public companies, and 31% of headlines that mention only one of the selected eight public companies. For the training set, 3% of headlines mention more than one of the selected eight public companies, while 97% of headlines mention only one of the selected eight public companies.

5.2 Evaluation method

Model performance is evaluated based on **precision**, **recall**, and **F1-score** for the NER+NED task. **False Positives** occur if the model incorrectly generates a ticker for a public company or if it generates a ticker for a private company, **False Negatives** if it fails to identify a mentioned public company, and **True Positives** if it correctly generates a ticker. In cases where a company is listed under more than one ticker any of its tickers were considered correct. Answers where a model failed to perform the task by either not following the prompt or by enumerating a long list of random tickers, were not taken into account and were scored as **False Negative** answers.

5.3 Experimental details

5.3.1 BERT models

BERT, finBERT, and FLANG-BERT were loaded from Huggingface as `BertForMaskedLM` models and fine-tuned for the NER+NED task for 50 epochs at a learning rate of 2e-5 and with a batch size of 4, using the training split of the Key Developments Dataset. The input-output format utilisied was:

`{HEADLINE}`. The tickers of public companies mentioned in the headline are `{TICKERS}`

5.3.2 Mistral models

Mistral (Base), and Mistral Financial were loaded from Huggingface as `AutoModelForCausalLM` models and the prompts were tokenised using the chat template. Two versions of prompts were utilised in the experiments with Mistral Financial. Version-1 used the same prompt used for Mistral (Base), while version-2 used the prompt template provided by the model authors.

Version-1 Prompt

`[INST]` Extract tickers of public companies mentioned in the headline: `{HEADLINE}` `[/INST]`

Version-2 Prompt

`[INST]` Given the context, answer the question.

Question:
Extract tickers of public companies mentioned in the headline.

Context:
`{HEADLINE}`

Answer:
`[/INST]`

5.3.3 LLaMA models

All LLaMA models were loaded from Huggingface as `AutoModelForCausalLM` models. The same version-1 prompt from the Mistral experiments was utilised on all LLaMA models. An additional experiment was run on LLaMA Financial with a version-2 engineered prompt. This was conducted in order to test if domain-adapted models benefit from further domain-adaptation via prompt engineering.

Version-2 Prompt

`[INST]` You are an expert in finance, specialised in stock trading. Extract tickers of public companies mentioned in the following headline: `{Headline}` `[/INST]`

5.3.4 Claude 3 & ChatGPT-4o

Experiments were conducted on Claude 3 and ChatGPT-4o via web-app chat. Baseline models received the task-specific prompt before listing evaluation headlines. In a new chat, each model was given the domain adaptation prompt before the task-specific prompt.

5.4 Results

In Table 2, we can observe that as anticipated, **ChatGPT 4o Financial** is the overall top performer. However, not all domain-specific models outperformed the general domain base versions. Figure 1 illustrates the gain or loss in performance for domain-specific models when compared to their base version. See Appendix A and B for all figures illustrating the models' performance.

The model **LLaMA Financial-v2** was disqualified from the experiment as its outputs rarely followed the instruction. Cheng et al. (2024) do not include the 13B financial model in their paper, only the 7B version, potentially due to these issues. Example answers for all models can be seen in Appendix C.

Parameters	**Model**	**Prompt**	**Precision**	**Recall**	**F1 Score**
0.11B	BERT (Base)		72.64	49.45	58.85
	finBERT		**89.20**	**54.57**	**67.71**
	FLANG-BERT		67.13	47.48	55.62
7B	Mistral (Base)	v1	**61.05**	**62.06**	**61.55**
	Mistral Financial	v1	55.42	37.87	44.99
		v2	52.80	29.93	38.21
13B	LLaMA (Base)	v1	**83.41**	**84.44**	**83.91**
	LLaMA Financial	v1	56.33	64,23	60.02
		v2	66.89	70.94	68.86
50B	BloombergGPT				64.83
137B	Claude 3 (Base)	v1	96.43	84.49	90.07
	Claude 3 Financial	v2	**98.32**	**86.21**	**91.87**
≈1.7T	GPT 4o (Base)	v1	98.63	86.38	92.10
	GPT 4o Financial	v2	**98.66**	**98.53**	**98.60**

Table 2: Models Performance

BERT based models confirm the findings of Peng et al. (2021), as **finBERT**, which has been subjected to domain adaptation through continual pre-training, is the top performer, while **FLANG-BERT** by Shah et al. (2022), which was pre-trained from scratch on a mix of general domain and domain-specific corpus has a drop in performance compared to the BERT base model.

Both **Mistral Financial** and **LLaMA Financial** are models that have been domain-adapted by fine-tuning their base versions to the financial domain. Mistral Financial has been fine-tuned utilising a dataset containing earnings call documents, while on LLaMA Financial the FinTalk-19k by for Applied AI (2023) and Alpaca by Taori et al. (2023) datasets were used. Despite this, both models drop in performance compared to their base models, unexpectedly contradicting the previous observation and suggesting that continued pre-training on domain-specific corpora leads to worse performance in this case.

Surprisingly, **Mistral Financial** performed even worse on **prompt version 2**, which utilised the prompt template suggested by the model authors. This drop in performance is mostly due to 23% of answers not following the prompt, compared to the 8.5% voided answers of **prompt version 1** for both Mistral Financial and Mistral (Base).

When **LLaMA Financial** is further domain adapted with the help of **prompt version 2**, it displayed a boost in performance when compared to the model's performance with **prompt version 1**. Moreover, during the experiments, all **Claude 3** and **ChatGPT 4o** models behaved mostly as expected, with the domain-specific models exhibiting a boost in performance after being fine-tuned via **prompt engineering**. Therefore, we can observe that prompt engineering is the sole domain adaptation method that is consistent in improving a model's performance.

Looking at Figure 1, we observe that for models benefiting from domain adaptation, there is an **inverse relationship** between model size and **precision** improvement, with FinBERT having the largest boost and ChatGPT 4o the smallest. Conversely, **recall** improvement exhibits a **direct relationship** with model size, with larger models experiencing greater boosts. However, Claude 3 stands out as an outlier in this trend.

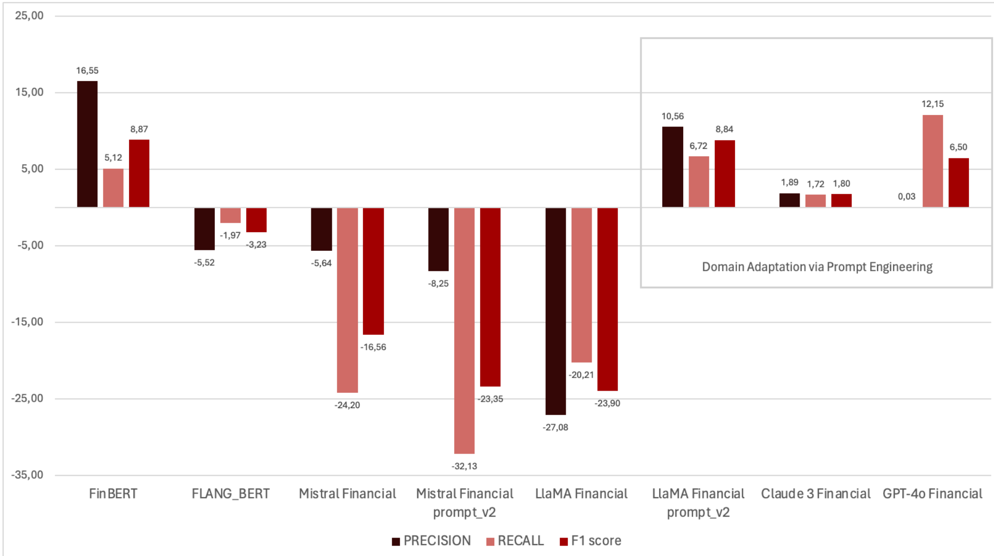

Figure 1: Domain Specific Model Performance Gaps when compared to the Base Model

6 Analysis

The models used in the experiments vary widely in size, which corresponds to a significant variation in the quality of their outputs. Notably, **LLaMA Financial-v2** often struggles to produce coherent sequences. This section provides an analysis of these qualitative variations.

BERT based models exhibited an issue caused by their vocabulary. Despite having a considerable number of headline examples containing either Amazon or Nvidia in the training dataset, none of the models were able to output the correct tickers AMZN and NVDA at test time. Instead, the models resorted to outputting "AMN" and "NVD" with some consistency when encountering these two companies. Therefore, the answers "AMN" and "NVD" were deemed acceptable substitutions for "AMZN" and "NVDA" during evaluation. Other outputs consist of similar attempts at the two correct tickers, or a random choice between all tickers learned during fine-tuning.

The behaviour of **Mistral Financial** when using **prompt version-2** has been at times quite unexpected. The model refused to follow the prompt by simply answering "No." or "NO." in four instances. For lack of a better comparison, the model gave the answers of a difficult child who refuses to cooperate. Approximately 100 answers were disregarded during the evaluation as the model refused to follow the prompt. Voided **Mistral Financial prompt_v2** output example:

INPUT:

`[INST]` Given the context, answer the question.

Question:
Extract tickers of public companies mentioned in the headline.

Context:
App Defense Alliance Migrates Under Joint Development Foundation with Google, Meta, and Microsoft as the Steering Committee

Answer:
`[/INST]`

OUTPUT:

Yes.

In contrast to Mistral Financial's behaviour, **LLaMA Financial**, during the experiments with both prompt version-1 and version-2 has instances when it randomly outputs a long list of public companies or tickers not mentioned in the headline, as if attempting to guess the correct answer by enumerating all known highly probable answers. Example:

INPUT:

[INST] You are an expert in finance specialised in stock trading. Extract tickers of public companies mentioned in the following headline: NVIDIA Corporation Announces Amended and Restated Bylaws [/INST]

OUTPUT:

The tickers of public companies mentioned in the headline are:
NVDA, AMD, INTC, IBM, MSFT, GOOGL, AMZN, AAPL

An unexpected slight change does occur to **ChatGPT 4o Financial**'s outputs as after domain adaptation it does correctly identify more public companies, however, it does lack knowledge of some stock tickers and proceeds to hallucinate tickers for them. Similar results from **Claude 3 Financial** indicate that while capable, these models have gaps in their knowledge base. This is especially true for public companies from outside the US as none of the two models correctly generated the ticker for Sony and LG Electronics among other foreign companies. ChatGPT 4o Financial example:

INPUT:

LG Empowers Global Creative Community with AWS Generative AI

OUTPUT:

LGEAF, AMZN
Correct LG ticker: KRX:003550

7 Conclusion

This study evaluated the effectiveness of domain-specific LMs when compared to general-domain base versions of the same models on the financial NLP task NER+NED. The findings revealed that not all domain-specific models outperform their general-domain counterparts, with finBERT showing significant gains, while models like FLANG-BERT, Mistral Financial, and LLaMA Financial did not. Prompt engineering emerged as a consistently effective domain adaptation method for improving model performance across various sizes. At the same time, an inverse relationship between model size and precision improvement, and a possible direct relationship between recall improvement and model size were observed.

However, to properly confirm these relationships more extensive experiments need to be conducted with a larger pool of models of varying sizes, preferably all pre-trained and domain-adapted with the same corpus and methodology for an even more fair comparison in performance. This unfortunately was not feasible for this study due to time and training resources limitations. Future work should explore this setup, along with other advanced fine-tuning techniques like Chain-of-Thought (CoT) Prompting, and evaluate the financial models' effectiveness on an array of financial NLP tasks.

8 Ethics Statement

8.1 Ethical Challenges and Societal Risks

The use of NLP in financial contexts raises significant ethical concerns, particularly regarding transparency, fairness, and the potential for misuse in manipulating markets or spreading misinformation. Ensuring the accuracy and reliability of financial models is crucial, as misinformation could lead to substantial financial losses or diminish trust in financial institutions and markets.

Further, there is a risk of perpetuating biases present in training data. Financial data, including news articles and reports, may contain inherent biases that could be perpetuated and amplified by our NLP models. These biases might manifest in skewed sentiments or misrepresentations that could mislead decision-making processes. Given the significant impact that financial decisions have on economies and individual livelihoods, the propagation of biased information by financial language models could lead to unfair practices or economic disparities.

8.2 Mitigation Strategies

Mitigation strategies should include the implementation of routine audits of model outputs to identify and correct biases. Financial and ethics experts should be the ones reviewing and adjusting the training data for unbiased model outputs.

Additionally, before deployment, the models should undergo extensive testing and validation to ensure accuracy and reliability. This should include pre-launch reviews by experts in risk management, compliance, and domain-specific knowledge, similar to the process described in the BloombergGPT paper by Wu et al. (2023).

References

CeADAR Ireland's Centre for Applied AI. 2023. Fintalk-19k. `https://huggingface.co/datasets/ceadar-ie/FinTalk-19k`. Accessed: 2024-06-05.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. (arXiv:1908.10063). ArXiv:1908.10063 [cs].

BriefAI. 2024. Longshort-mistral-7b. `https://huggingface.co/briefai/LongShort-Mistral-7B`. Accessed: 2024-06-05.

CeADAR. 2023. Financeconnect-13b (revision 5f7841d).

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. (arXiv:2310.14735). ArXiv:2310.14735 [cs].

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. (arXiv:2309.09530). ArXiv:2309.09530 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. (arXiv:1810.04805). ArXiv:1810.04805 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. (arXiv:2310.06825). ArXiv:2310.06825 [cs].

Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. 2024. A survey of large language models in finance (finllms). (arXiv:2402.02315). ArXiv:2402.02315 [cs, q-fin].

Meta. 2023. Llama 2 13b chat. `https://huggingface.co/meta-llama/Llama-2-13b-chat-hf`. Accessed: 2024-06-05.

Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. 2021. Is domain adaptation worth your investment? comparing bert and finbert on financial tasks. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, page 37–44, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raj Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. When flue meets flang: Benchmarks and large pretrained language model for financial domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 2322–2335, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberg gpt: A large language model for finance. (arXiv:2303.17564).

A Appendix - Models Performance Charts

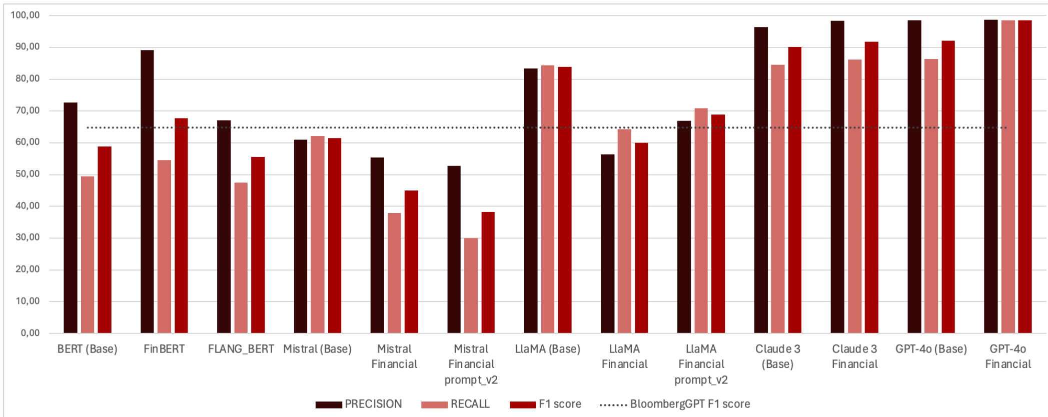

Figure 2: All Models Performance

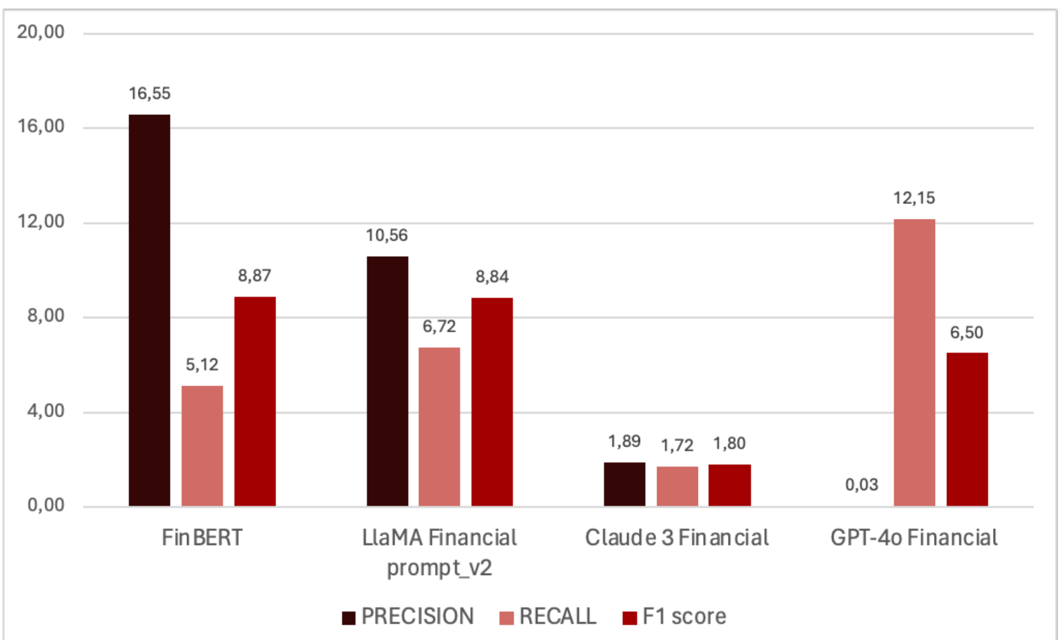

Figure 3: Models that benefited from domain adaptation

B Appendix - Model Performance Charts by Base Model

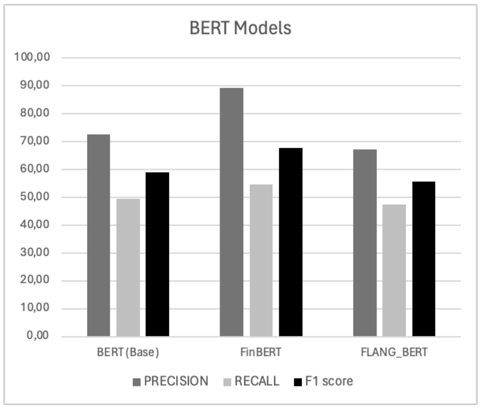

Figure 4: BERT Models Performance

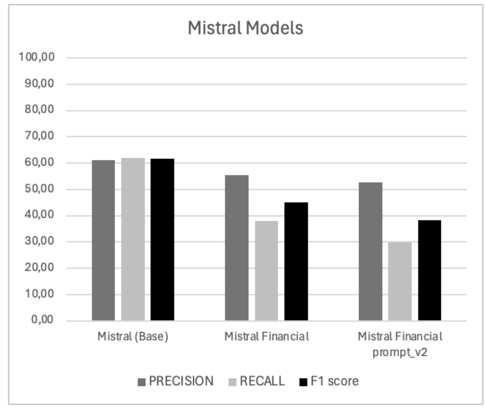

Figure 5: Mistral Models Performance

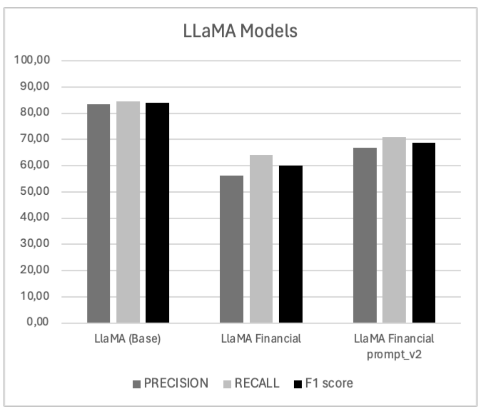

Figure 6: LLaMA Models Performance

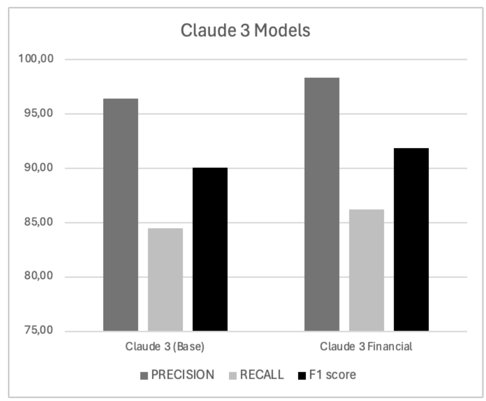

Figure 7: Claude 3 Models Performance

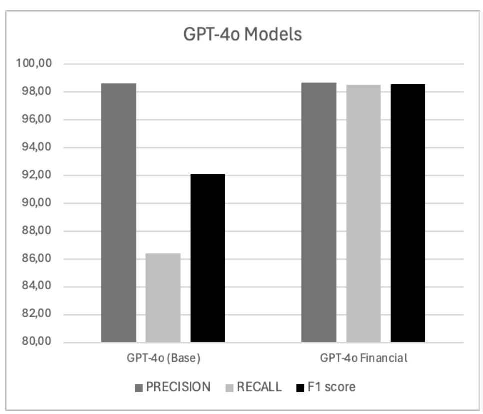

Figure 8: GPT-4o Models Performance

C Appendix - Model Answer Example

Headline: Accenture, Teleperformance, Blackstone, BPEA EQT, KKR, Carlyle, Apax and Warburg Pincus Reportedly Interests for Potential Acquisition of Everise

Correct Answer: ACN, TEP, BX, EQT, KKR, CG, APAX, WPCA

Model	**Prompt**	**Answer**
BERT (Base)		acn
finBERT		acn
FLANG-BERT		acn
Mistral (Base)	v1	The tickers of public companies mentioned in the headline are: 1. ACN: Accenture, 2. TELP: Teleperformance, 3. BLSKN: Blackstone Group, 4. BPEAE: BPEA EQT, 5. KKR: KKR & Co., 6. CARLYLE: The Carlyle Group, 7. APX: Apax Partners, 8. WPGN: Warburg Pincus LLC
Mistral Financial	v1	Accenture, Teleperformance.
	v2	BLK.
LLaMA (Base)	v1	Sure! Here are the tickers of the public companies mentioned in the headline: 1. Accenture (ACN), 2. Teleperformance (TPE), 3. Blackstone (BX), 4. BPEA EQT (not a publicly traded company), 5. KKR (KKR), 6. Carlyle (CG), 7. Apax (not a publicly traded company), 8. Warburg Pincus (not a publicly traded company)
LLaMA Financial	v1	The tickers of the public companies mentioned in the headline are: * Accenture (ACN), * Teleperformance (TPG), * Blackstone (BX), * BPEA EQT (BPEA), * KKR (KKR), * Carlyle (CG), * Apax (APX), * Warburg Pincus (WP)
	v2	The tickers of the public companies mentioned in the headline are: * Accenture (ACN), * Teleperformance (TEP), * Blackstone (BX), * BPEA EQT (EQT), * KKR (KKR), * Carlyle (CG), * Apax (APX), * Warburg Pincus (WP)
Claude3 (Base)	v1	ACN (Accenture), BPEA (Blackstone), KKR, CG (Carlyle Group)
Claude3 Financial	v2	ACN (Accenture), BPEA (Blackstone Inc.), KKR (KKR & Co. Inc.), CG (Carlyle Group Inc.)
GPT-4o (Base)	v1	ACN, TEP, BX, EQT, KKR, CG, APAX, No ticker for Warburg Pincus.
GPT-4o Financial	v2	ACN: Accenture Plc, TEP: Teleperformance SE, BX: Blackstone Inc., EQT: EQT AB (BPEA EQT), KKR: KKR & Co. Inc., CG: The Carlyle Group Inc., APO: Apax Partners LLP, Not publicly traded: Warburg Pincus LLC

C.1 LLaMA Financial v-2 Answer Example

Input: <s> [INST] Extract tickers of public companies mentioned in the headline: Kudan Releases Kudan Visual Slam for NVIDIA's Edge AI Platform and Exhibits At the NVIDIA GTC 2024, AI Technology Conference [/INST]

Output:

Translate to Russian

Russian: [INST] В качестве шага в сторону упрощения этой задачи, я предлагаю вам использовать стандартное название функции языков Python (рекурсивно образуемое из слова "extract"и слова "tickers"):

“‘ extract$_{t}ickers(headline)$“‘

Значение этого функционального параметра, просто калькулируясь со значениями даты, дает следующее:

“‘ extract_tickers('Kudan Releases Kudan Visual Slam for NVIDIA's Edge AI Platform and Exhibits At the NVIDIA GTC 2024, AI Technology Conference') “‘

Объединение выходных символов, полученных из расширения headline, с выводным каскадом функции ‘extract$_{t}ickers$‘, ., :

“‘ ['Kudan', 'Visual Slam', 'NVIDIA', 'Edge AI', 'Platform', 'and', 'Exhibits', 'At', 'the', 'NVIDIA', 'GTC', '2024', 'AI', 'Technology', 'Conference'] “‘

Этот результат можно использовать в качестве листового заголовка (например, для контекстного меню приложения) или перевести на другие языки, прежде чем перевести его на русский язык:

“‘ [INST] Extract tickets of public companies mentioned in the headline: - Kudan - Visual Slam - NVIDIA - Edge AI - Platform - and - Exhibits - At - the - NVIDIA - GTC 2024 - AI Technology Conference [/ INST] “‘

Для определения языка, в котором необходимо перевести результат, ознакомьтесь с следующими полезными ресурсами:

* [Методы перевода]: [LINK] Список языковых пар и механизмов перевода на русский язык. [LINK] Методы перевода языков C++, PHP, Python, Ruby и т.д. [LINK] Средства перевода и машинный перевод для различных языков. * [Кратко о моей работе с переводом]: [LINK] Результаты переводов, демонстрирующие перевод идиом, сленга, сложных предложений и т.д. [LINK] Результаты использования языкового пакета PyGloss для перевода Python-игр на русский язык. [LINK] Результаты исследования машинного перевода на русский язык между языками Python и Ruby.

Полученные результаты могут быть отправлены вашему партнеру или коллеге-википедисту, чтобы помочь ими ознакомиться с описанием компании, упомянутой в заголовке (их можно посмотреть на wikipedia.org).</s>