Bank of England

Improving text classification: logistic regression makes small LLMs strong and explainable ‘tens-of-shot’ classifiers

Staff Working Paper No. 1,127
May 2025

Marcus Buckmann and Ed Hill

Bank of England

Staff Working Paper No. 1,127

Improving text classification: logistic regression makes small LLMs strong and explainable 'tens-of-shot' classifiers

Marcus Buckmann[(1)] and Ed Hill[(2)]

Abstract

Text classification tasks such as sentiment analysis are common in economics and finance. We demonstrate that smaller, local generative language models can be effectively used for these tasks. Compared to large commercial models, they offer key advantages in privacy, availability, cost, and explainability. We use 17 sentence classification tasks (each with 2 to 4 classes) to show that penalised logistic regression on embeddings from a small language model often matches or exceeds the performance of a large model, even when trained on just dozens of labelled examples per class – the same amount typically needed to validate a large model's performance. Moreover, this embedding-based approach yields stable and interpretable explanations for classification decisions.

Key words: Text classification, large language models, machine learning, embeddings, explainability.

JEL classification: C38, C45, C80.

---

(1) Bank of England. Email: marcus.buckmann@bankofengland.co.uk

(2) Bank of England. Email: ed.hill@bankofengland.co.uk

We thank Max Bartolo, Moritz Pfeifer, Paul Robinson, Eryk Walczak and the participants of the CEBRA Annual Meeting 2024.

The Bank's working paper series can be found at www.bankofengland.co.uk/working-paper/staff-working-papers

Bank of England, Threadneedle Street, London, EC2R 8AH
Email: enquiries@bankofengland.co.uk

ISSN 1749-9135 (on-line)

1 Introduction

Text classification – assigning passages to categories such as sentiments or topics – is an important problem in economics, finance, and central banking that enables modellers to treat text as quantitative data. Applications are, among many others, the measurement of sentiment and monetary policy stance of central bank communication (Chen et al., 2023; Bertsch et al., 2025; Pfeifer and Marohl, 2023) and the detection of changes in topics and sentiment across news articles, board minutes, and earnings calls to understand the financial conditions of individual firms or the broader economy (Loughran and McDonald, 2011; Thorsrud, 2020; Cook et al., 2023). Text classification models can also serve as building blocks for macroeconomic forecasting methods that incorporate textual sources (Ardia et al., 2019; Ellingsen et al., 2022).

Traditionally, text classification has relied on dictionary-based approaches, where words are manually mapped to categories (Hansen and McMahon, 2016; Hubert and Fabien, 2017). However, these methods often lack robustness, as they struggle to capture context, nuance, and linguistic features such as negation, and they may not generalise well across domains.

With the proliferation of language models, they are increasingly being adopted for text classification in economics (Gössi et al., 2023; Cook et al., 2023; Kim et al., 2024). Generative models such as GPT-4 (Achiam et al., 2023), Claude 3 (Anthropic, 2024) and Gemini (Gemini Team, Google, 2023) have demonstrated impressive zero-shot performance across a wide range of tasks including text classification (Kim et al., 2024; Kostina et al., 2025).[1] However, these proprietary models share the typical disadvantages of cloud-based Software-as-a-Service, such as concerns about privacy, dependence on internet connectivity, and financial cost, as well as limited explainability and consistency, since changes to or deprecation of the model are beyond the end user's control.

Alternatively, non-generative transformer models, such as fine-tuned versions of encoder-only models, can achieve good performance in text classification (e.g. Gössi et al., 2023; Li et al., 2023; Pfeifer and Marohl, 2023). But fine-tuning such models comes at a significant cost in time, expertise, and computation.

In this study we show that we can realise the advantages of using small *locally hosted* generative models for sentence classification, without incurring trade-offs in performance or significant other costs. We do this by using their hidden states (also referred to as *embeddings* in what follows). While the text output of local generative models has been used in the economic literature for text classification (Cook et al., 2023; Konstantinidis et al., 2024), their hidden states are usually not considered.

[1] In this work we compare against GPT-4 because in late 2023, when the work was performed, GPT-4 was a strong and standard benchmark (and remains so across many tasks (Anthropic, 2024, table 1), also Section 1.1). Explicitly, we are not specifically discussing the use, advantages or disadvantages of GPT-4 versus other flagship large language models (LLMs), rather using it as a standard baseline against which to test our methods.

We present three key results, supported by a quantitative and qualitative discussion. First, we find that penalised logistic regression (PLR) on the embeddings produced by a small local model (we use a quantised Llama2 7B model as a baseline (Touvron et al., 2023)) can equal or exceed the performance of GPT-4 on sentiment analysis and classification tasks. By contrast, the text output of the local models cannot compete with that of GPT-4 in most datasets. Second, we observe that in the majority of datasets, only 60–75 training samples per class are required to train a PLR model that beats GPT-4. By contrast, we require substantially more instances to obtain small enough confidence bounds to state that GPT-4 has a *statistically* better performance than PLR and vice versa. Finally, we show that the PLR model provides stable word-level explanations in this "tens-of-shot" regime. We validate that these explanations are sensible against human annotations.

While we consider a few datasets relating to economics and finance, this paper looks at text classification tasks more broadly and thus draws on standard datasets from the machine learning literature. Similarly, the literature review in the next section focuses on the methodological text classification literature in machine learning and does not specifically cover the economics and finance literature.

The paper is structured as follows: Section 1.1 reviews the literature and Section 1.2 describes our methodology. Section 2 presents our results, beginning with a standard learning curve analysis, where models trained with varying amounts of data are assessed based on their out-of-sample performance. We show how performance is affected by prompting strategies and the choice of the language model. Section 3 then moves to the setting which mimics the case where only limited labelled data is available for both training and testing, like when a dataset or task is being approached for the first time. In this setting the quantity of labelled data influences both the performance of PLR models and the uncertainty of the performance estimates of both the PLR model and of the large LLM. Section 4 discusses the structure and characteristics of the embeddings from the local model and Section 5 considers the stability and accuracy of feature importance explainability methods. Section 6 concludes.

1.1 Literature review

Recent studies have shown that large general-purpose LLMs such as GPT-4 are competitive text classifiers (Rathje et al., 2024; Kim et al., 2024; Kostina et al., 2025), including on tasks that require specialised domain expertise (Savelka et al., 2023). Zhang et al. (2023) conducted a large scale empirical assessment across 13 sentiment analysis tasks on 26 datasets and conclude that "Even in a zero-shot setting, [LLMs'] performance can match or surpass fine-tuned smaller language models, and with little sensitivity to different prompt designs." However, other studies found that human annotators or specialised models calibrated on human annotations outperform large LLMs (Liyanage et al., 2023; Li et al., 2023; Toney-Wails et al., 2024; Vajjala and Shimangaud, 2025).

Our approach – learning a linear model on the hidden states of large neural networks

– is known in the literature as linear probing (Belinkov, 2022; Alain and Bengio, 2016). This technique has mostly been used to understand what information hidden states of language models represent (Jawahar et al., 2019; Zhu et al., 2024; Gurnee and Tegmark, 2023; Chen et al., 2023; Campbell et al., 2023). Recent work has shown that linear probing can improve the accuracy of generative language model predictions: Cho et al. (2023) used linear probing on 12 classification tasks and showed that it outperforms in-context learning for both GPT-J and GPT-2. As in our work, the best performance was obtained when augmenting the text to classify with a prompt stating the classification task. Jiang et al. (2023) and Zhang et al. (2024) also use a surrounding prompt to improve the quality of embeddings of generative models for downstream tasks. Vajjala and Shimangaud (2025) report strong text classification performance of a linear model trained on embeddings. However, they obtain the embeddings from a small sentence embedding model, as we do in Section 2.2, where we find that PLR-E on these embeddings performs worse than PLR-E trained on the embeddings of larger *generative* models.

Instead of learning a linear model on the embeddings, Abbas et al. (2024) used linear probing to calibrate the token probability, an approach we use as a baseline in our paper as well (PLR-L below).

Fine-tuning local generative models to enhance the quality of their embeddings has shown that large models trained on broad corpora can outperform smaller, dedicated sentence embedding models on benchmark evaluations (Wang et al., 2023; Rui Meng, 2024).

Our paper also relates to the literature on edge computing, and to the cost-effective use and 'democratisation' of AI. Despite recent advances in the efficiency of fine-tuning (Hu et al., 2021; Liu et al., 2024), fine-tuning BERT-type models to a bespoke classification problem requires the collection of appropriate data and access to sufficient computational resources and expertise. By contrast, learning a simple linear model on top of the embedding of an open-source generative model is computationally cheap, and penalised logistic regression is amongst the best known and widely understood prediction methodologies across disciplines. While inference with a several billion parameter model as in this paper is substantially slower than using sub-billion parameter previous-generation models, we note that the model used is already a year old, and improvements in model quantisation and pruning (Lin et al., 2024; Sun et al., 2023; Dettmers et al., 2023; Ma et al., 2024) and the development of more efficient model architectures (Gu and Dao, 2023; Peng et al., 2023) will continue to decrease the memory and computational footprint of models with equivalent performance.

Furthermore, using proprietary LLMs such as GPT-4 or Gemini for text classification has several disadvantages including the exposure of private data (both to the provider of the service and, possibly, the communication system), payment for the service, the requirement of (stable) internet access, and the lack of model consistency against deprecation or updating. Not having access to the model's parameters removes flexibility, limiting options around fine-tuning and model adaptation, and also means that tasks which cannot be easily expressed as a prompt (for example classifying text according to an individual's personal preferences) cannot be performed.[2] And while many approaches exist to explain various aspects of the behaviour and limitations of LLMs (Zhao et al., 2023), these usually cannot be applied without having access to the full model. Explainability is important both to inform their practical commercial use, and for diagnosing and avoiding pathological behaviours for robust performance (e.g. Du et al., 2023) and legal compliance (European Union, 2024).

1.2 Methodology

Our method has three steps: prompt construction, text embedding, and penalised logistic regression (PLR). We will describe these steps with signposts to robustness and other checks.

1.2.1 Prompt construction

We add a contextualising prefix and a suffix indicating the classification task to be performed around the text to be classified. An example from the Financial Phrases dataset is

I am extremely delighted with this project and the continuation of cooperation with Viking Line.

This is extended to

The following sentence contains financial news: I am extremely delighted with this project and the continuation of cooperation with Viking Line. Does the sentence have (a) positive, (b) negative, (c) neutral sentiment? Answer: (

Similarly, for a classification problem with two classes, such as the irony dataset, we extend the text

Today is going to be a great day.

to

Consider the following tweet: Today is going to be a great day . #not. Is this tweet ironic? Answer with Yes or No. Answer:

We show that adding this surrounding text substantially improves performance relative to using just the text and that the results are robust to the precise wording and direction of the surrounding prompt (Section 2.3).

[2]The relative importance of these issues will depend on the user's situation. Regarding cases where the user wants to limit, but need not eliminate, external LLM usage, this work complements model cascade and selection methods (Chen et al., 2023; Šakota et al., 2023), possibly superseding them in the simple classification case.

1.2.2 Embedding

The prompt is fed to the LLM, which returns an embedding used for the classification task. The embedding is the final layer activation before the prediction head, which is 4096 dimensional for our baseline model. We show that our results are robust to the size and quantisation of the model in Section 2.4.

1.2.3 Text prediction

We also assess the quality of the text output. The LLMs usually provide an answer where the first token is one of the candidate tokens specified in the prompt (e.g. "Yes", or "No" for binary classification tasks or "a", "b", or "c" for a three-class task). To handle rare cases where the token with the highest overall logit is not among the candidates, we always select the answer with the highest logit within the candidate set.

1.2.4 Penalised logistic regression

Finally, we perform penalised logistic regression (PLR) on the embeddings. Specifically, we use ridge regression, that is, l_2 regularisation. In the binary case this minimises the log-likelihood

$$\ell = \sum_{k=1}^{K} y_k \ln(p(\boldsymbol{e_k})) + \sum_{k=1}^{K} (1 - y_k) \ln(1 - p(\boldsymbol{e_k})) + \lambda \|\boldsymbol{a}\|^2 \tag{1}$$

for the K instances with class $y_k \in \{0, 1\}$, embedding vector $\boldsymbol{e}_k$, and regularisation parameter λ to find a_0 and $\boldsymbol{a}$ in the function

$$p(\boldsymbol{e}) = (1 + e^{-(a_0 + \boldsymbol{a}.\boldsymbol{e})})^{-1}. \tag{2}$$

$\boldsymbol{a}$ is the normal to the classification surface and $a_0 + \boldsymbol{a}.\boldsymbol{e} = \ln(p/(1-p))$ denotes the log-odds.

1.2.5 Implementation

Language models. We use Llama 2 7B (Touvron et al., 2023) as our baseline model, which, at the time of conducting the experiments, was a leading small open-source model. To test the robustness of our results, we also consider Llama2 13B and the smaller, but more recent, Stable LM Zephyr 3B model (Stability AI, 2024).

To test the usefulness of LLMs for classifications on standard consumer hardware with 16GB RAM and no GPU, we use quantisations of all LLMs, provided by `TheBloke` on Hugging Face. For the Llama models we use the *ggml* quantisation, for the Zephyr model, which we tested at a later point in time, we use the more recent *gguf* quantisation. Additionally we evaluated the performance of two sentence embedding models. First, we used `bge-large-en-v1.5` (Xiao et al., 2024), a 326-million-parameter 1024-dimensional embedding model that performed best on the MTEB benchmark (Muennighoff et al., 2022) both across all tasks and on classification tasks specifically, as of

September 9, 2023. The model can be downloaded from Hugging Face. Second, we tested `text-embedding-ada-002` – OpenAI's best embedding model as of September 9, 2023 – which is accessible via the OpenAI API.

Penalised logistic regression. We estimate ridge regression using the R package `glmnet`. Unless stated otherwise, we use the default regularisation path (100 different values for λ) and choose the model with the lowest degree of regularisation instead of conducting a hyperparameter search. Section 4 shows that our model is surprisingly robust to the exact choice of λ: fixing its value makes our approach computationally cheaper and less complex, increasing its value for practitioners.

1.2.6 Datasets and performance metric

We evaluate the performance on 17 classification tasks drawn from diverse domains, including movie reviews, news headlines, YouTube comments, tweets, and Reddit posts. While these datasets can all be considered simple classification tasks, where the sentences are homogeneous and classes are clearly defined (e.g. positive vs. negative sentiment or whether a comment is spam or not), the central banking dataset constitutes a more complex classification problem. The annotators were given a detailed guide that defined hawkishness and dovishness for eight different categories (economic status, dollar value change, energy/house prices, foreign nations, Fed expectations/actions/assets, money supply, keywords/phrases, labour). Both annotators were *"researchers who have taken finance-related coursework and understood macroeconomics"*.

We do not make use of all instances of the larger datasets but instead rely on sub-sampling. For the binary datasets, sub-sampling is performed to ensure class balance. Our main performance metric is accuracy but we also report F1 scores for our main results and find that these are qualitatively highly similar. The datasets and the sizes of the samples we draw are described in detail in Appendix A.

2 Performance comparison: Small training set, large test set

We begin with a learning curve analysis. Specifically, we sample 20% of the observations of a dataset as the test set and sample training samples of increasing size from the remaining 80% of observations. To obtain stable performance estimates we repeat this procedure 50 times. Confidence intervals show ± 1.96 standard errors around the mean performance estimate and are estimated using bootstrapping.

Figure 1 shows the learning curves for 12 classification problems (the remaining learning curves being shown in Figure C.1 in the Appendix). In each case we show the accuracies of zero-shot next token prediction for both GPT-4 and the baseline (Llama2 7B q4.0) model, along with penalised logistic regression models trained on the baseline model's next token logits (PLR-L) and on the baseline model's embedding (PLR-E).

Despite the zero-shot next token accuracy of the baseline model significantly underper-

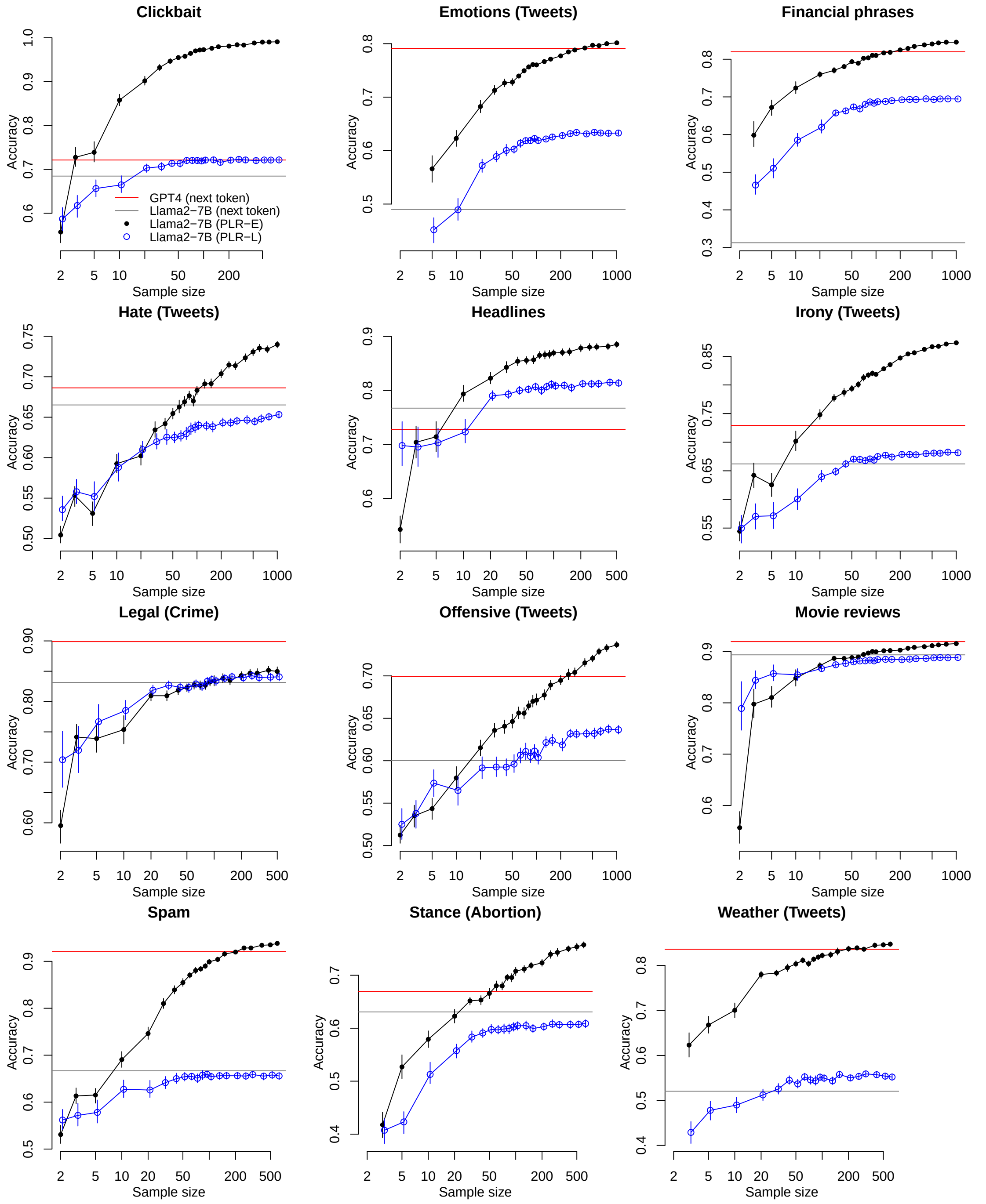

Figure 1: The accuracies of the zero-shot next token text predictions from GPT-4 and Llama2-7B, along with the learning curves for the PLR-L and PLR-E methods applied to our baseline model (Llama2-7B q4.0).

forming relative to GPT-4, the accuracy of the PLR-E method becomes comparable to or exceeds that of GPT-4 as the training sample size is increased. PLR-L is inferior to PLR-E but exceeds the performance of the baseline model's next token prediction in several datasets.

Even with sample sizes of 10 observations, PLR-E outperforms the baseline model's next token prediction in 9 of the 17 datasets. This is surprising due to the very small sample size and the high dimensionality of the embedding space.

These results also hold when using the F1 macro score as a performance metric – the analogous learning curves are shown in Figure C.3 in the appendix.

Figure 2 presents an alternative view of the learning curves across all 17 datasets. From left to right, it compares the accuracy of GPT-4 with that of our baseline model using zero-shot next-token prediction, PLR-L at a sample size of 100, and PLR-E at sample sizes of 10 and 100. On average, the next token prediction and PLR-L underperform relative to GPT-4, while PLR-E surpasses GPT-4 when trained on 100 examples. Even in datasets where PLR-E underperforms at this sample size, the margin of difference is generally small.

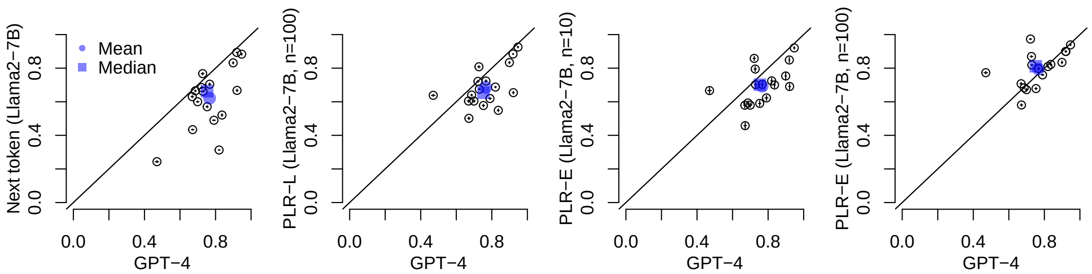

Figure 2: Comparing GPT-4 to our baseline model. From left, using the baseline model's next token prediction, learning from its logits (PLR-L), and learning from its embeddings (PLR-E).

Table 1 shows the performance of PLR-E at different sample sizes. The last two columns show at what sample size (divided by the number of classes) PLR-E outperforms GPT-4. We observe that PLR-E eventually beats GPT-4 in all but four datasets. The last column replicates this analysis but here the embeddings on which we train PLR-E are only based on the sentences, omitting the instructions (see Section 2.2).

2.1 Relating next token prediction, logits, and embeddings

Considering a two-class problem, we can decompose the performance gap between next-token prediction and the PLR-E method. We use l_+ and l_- to denote the logits associated with the responses for the first and second class (the logits of 'Yes' and 'No', or 'a' and 'b', for example), and $\boldsymbol{e}_+$ is the vector used in the prediction head to extract the logits

Dataset	Classes	Sample size (PLR-E, full prompt)					GPT-4 Token	Min. sample (per class) for PLR-E win	
		10	30	100	250	400		Full prompt	Sentence
Central banking	3	0.46	0.53	0.58	0.62	0.63	0.67	-	-
Clickbait	2	0.86	0.93	0.97	0.98	0.99	0.72	2	5
Emotions (Tweets)	4	0.62	0.71	0.76	0.78	0.79	0.79	100	-
Financial phrases	3	0.72	0.77	0.81	0.83	0.84	0.82	67	-
Hate (Tweets)	2	0.59	0.63	0.68	0.71	0.72	0.69	62	100
Headlines	2	0.79	0.84	0.87	0.88	0.88	0.73	5	10
Irony (Tweets)	2	0.70	0.78	0.82	0.85	0.86	0.73	10	15
Legal (Crime)	2	0.75	0.81	0.83	0.85	0.85	0.90	-	-
Legal (Money)	2	0.70	0.74	0.80	0.82	0.83	0.77	25	75
Legal (Work)	2	0.92	0.93	0.94	0.94	0.95	0.95	250	-
Offensive (Tweets)	2	0.58	0.64	0.67	0.70	0.72	0.70	125	-
Movie reviews	2	0.85	0.89	0.90	0.91	0.91	0.92	-	-
Spam	2	0.69	0.81	0.90	0.93	0.93	0.92	125	100
f Stance (Abortion)	3	0.58	0.65	0.71	0.74	0.75	0.67	20	83
Stance (Atheism)	3	0.67	0.72	0.77	0.81	0.82	0.47	1	2
Stance (Feminism)	3	0.59	0.64	0.68	0.71	0.72	0.75	-	-
Weather (Tweets)	3	0.70	0.78	0.82	0.84	0.84	0.84	67	-
Mean	2	0.69	0.75	0.80	0.82	0.83	0.77		
Median	2	0.70	0.77	0.81	0.83	0.84	0.75		

Table 1: The accuracy of PLR-E at different sample sizes, the accuracy of GPT-4's next-token prediction, and the minimum sample size (per class, for the full prompt and with the sentence only) where PLR-E wins against GPT-4's next token prediction.

such that $l_+ = \boldsymbol{e}.\boldsymbol{e}_+$, and similarly for l_-. The log-odds can be decomposed as:

$$\begin{aligned} a_0 + \boldsymbol{a}.\boldsymbol{e} = &(l_+ - l_-) \\ &+ [a_\pm + (a_+ - 1)l_+ + (a_- + 1)l_-] \\ &+ \left[\boldsymbol{a}.\left(\boldsymbol{e} - \sum_{i=+,-} \frac{(a_i - \boldsymbol{a}.\boldsymbol{e}_i) + \boldsymbol{a}.\boldsymbol{e}_i}{\boldsymbol{a}.\boldsymbol{e}_i}\boldsymbol{e}_i\right) + a_0 - a_\pm\right] \end{aligned}$$

The first term is the log-odds associated with next-token prediction: If $l_+ - l_- > 0$ then $+$ is predicted, and $-$ otherwise.

The first square-bracketed term still only considers the log-odds but represents the learned correction to the position of the classification surface, which may need translating and rotating in the $\{l_+, l_-\}$-space to best fit the data. Figure 1 shows that the model was well positioned in some cases (i.e. $a_\pm$, $a_+ - 1$, and $a_- - 1$ are all small) and so the zero-shot next-token prediction results in an accuracy very close to the large sample limit for PLR-L. PLR-L always eventually exceeds the performance of the zero-shot next-token case as the position of the classification surface is improved.

The second square-bracketed term results from the PLR-E model being able to discriminate based on directions outside of the plane spanned by $\boldsymbol{e}_+$ and $\boldsymbol{e}_-$. This allows it to bring in features which discriminate between training instances but which were not projected into the logit directions, or were, but their contribution was swamped by other variance. Examining the learning curves in Figures 1 and C.1 suggests that despite not being in the logit directions these features are generally well separated, leading to the model learning rapidly with new training instances.

2.2 Robustness: Omitting instructions and sentence embedding models

To investigate the role of the instructions for the accuracy of PLR-E, we replicate the analysis, but now train PLR-E on embeddings found when omitting the surrounding instructions from the prompt. Additionally, we train PLR-E models (both on the full prompt and the sentence only) on the embeddings from two standard non-generative sentence embedding models, `bge-large-en-v1.5` and `ada-002`. Neither embedding model is instruction-tuned but both have different sizes and lineages.

Figure 3 compares the different models and prompts in learning curves for 12 datasets with the remaining datasets shown in Figure C.2 in the appendix. First, we observe that in most datasets PLR-E on Llama2 embeddings is more accurate when including the instructions in the prompt. Second, we find that the sentence embedding models `bge` and `ada2` are mostly inferior to PLR-E, particularly at small sample sizes. For both sentence embedding models adding the contextualising prompt does not improve, and often hurts, the performance.

Figure 4 summarises the findings from the learning curves. It compares our baseline approach (horizontal axis) to PLR-E trained on Llama-2 embeddings without instructions (first row) and to PLR-E trained on `bge` (second row) and `ada2` (third row) embeddings.

2.3 Robustness: Choice of prefix and suffix

At small sample sizes, adding instructions boosts the performance of PLR-E. But are the models also sensitive to the wording of the instructions? To test this, we attach different prefixes and suffixes to the sentence, as shown in Table 3 in the appendix.

Figure C.4 in the appendix shows learning curves of PLR-E model for the different prompts in two datasets. In both datasets, we observe that those prompts with minimal or no instructions perform worse, with accuracies between 0.1 and 0.2 lower for small training set sizes. For other prompt configurations we do not observe substantial differences in their performance (with a spread in accuracies of around 0.05), suggesting that PLR-E is robust to the exact prompt specification as long as the instructions are complete.

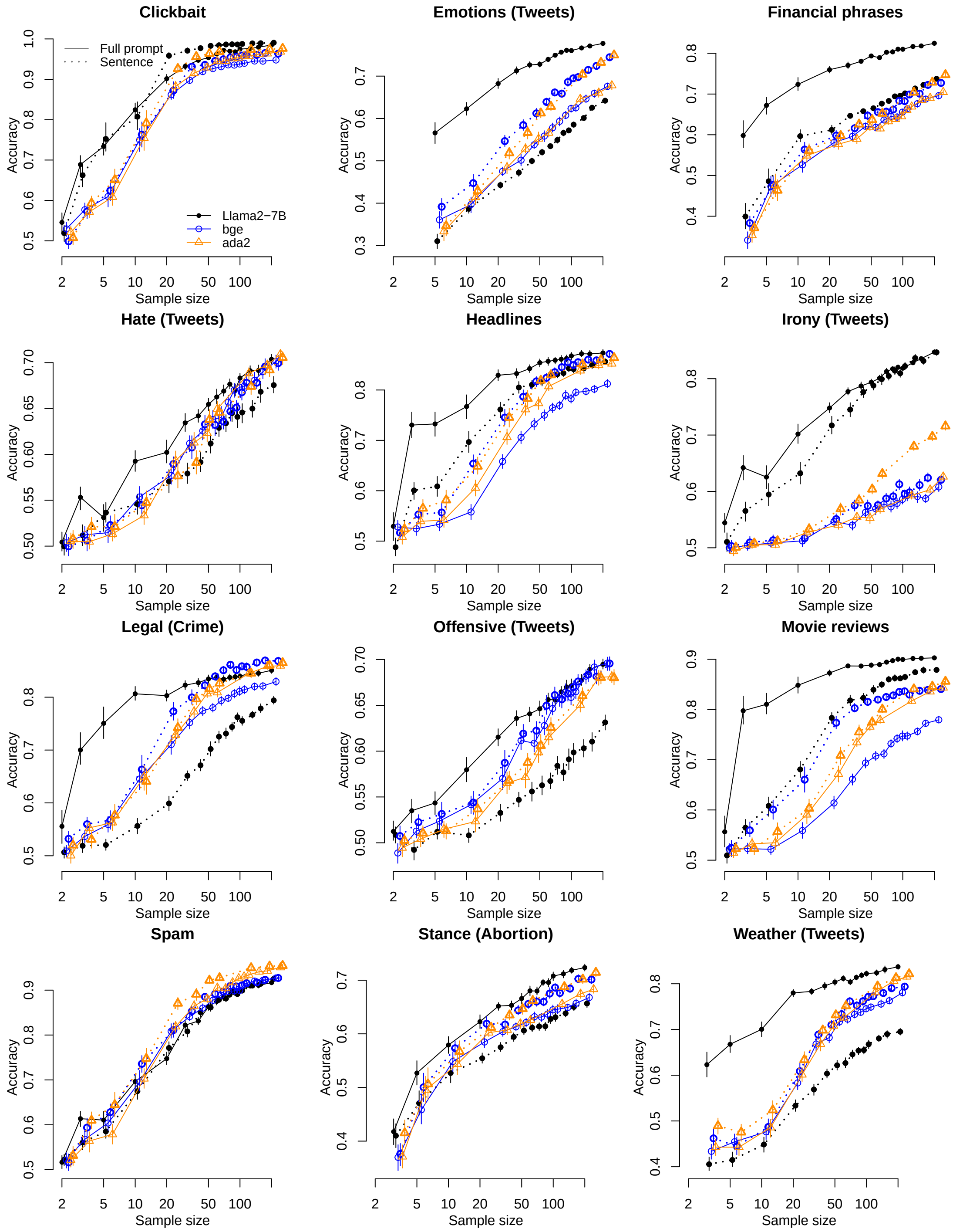

Figure 3: The accuracy of PLR-E when trained on embeddings from different models and prompts, c.f. Figure 1.

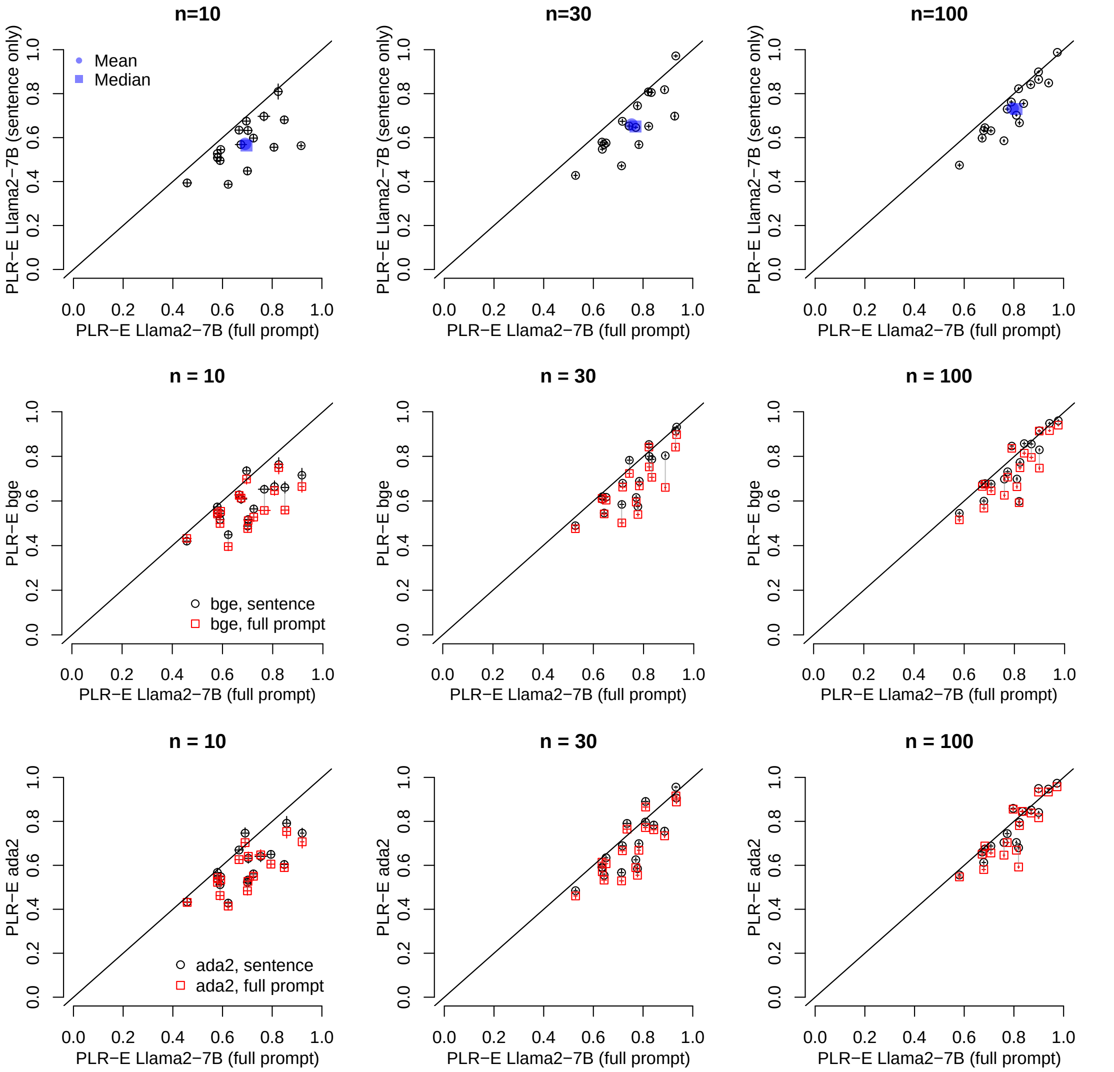

Figure 4: Instruction prompting. Top panel: Comparing the performance when using PLR-E on our baseline model with and without surrounding instructions. Middle and bottom panels: Comparing our baseline PLR-E (with instructions) against applying PLR-E to the embeddings from two sentence embedding models with (red arrows) and without instructions (black crosses).

2.4 Robustness: Model size and quantisation

We test whether our results hold for different generative language models. Specifically, we compare the larger Llama2 13B-chat (q4.0) and the smaller, but more recent, Stable LM Zephyr 3B (q5.0) to our baseline model. The results are shown in the top two rows of Figure C.6 in the appendix. Considering next token prediction, Llama2 13B outperforms the baseline model in most datasets. However, when evaluating the performance of

Datasets	GPT-4	Llama2 7B		Llama2 13B		Zephyr 3B		ada2	bge
	Token	PLR-E	Token	PLR-E	Token	PLR-E	Token	PLR-E	PLR-E
Central banking	**0.67**	0.58	0.43	0.58	0.44	0.60	0.48	0.55	0.52
Clickbait	0.72	**0.97**	0.68	**0.97**	0.76	0.96	0.53	0.96	0.94
Headlines	0.73	**0.87**	0.77	0.86	0.63	0.86	0.53	0.84	0.80
Spam	0.92	0.90	0.67	0.89	0.64	0.92	0.56	**0.93**	0.91
Financial phrases	**0.82**	0.81	0.31	**0.82**	0.63	**0.82**	0.66	0.67	0.66
Weather (Tweets)	**0.84**	0.82	0.52	0.81	0.68	0.83	0.77	0.78	0.75
Irony (Tweets)	0.73	**0.82**	0.66	0.81	0.56	0.66	0.53	0.59	0.59
Emotions (Tweets)	**0.79**	0.76	0.49	0.76	0.70	0.78	0.67	0.65	0.63
Offensive (Tweets)	**0.70**	0.67	0.60	0.67	0.63	**0.70**	**0.70**	0.65	0.66
Hate (Tweets)	0.69	0.68	0.67	0.61	0.65	**0.72**	0.67	0.69	0.67
Stance (Feminism)	**0.75**	0.68	0.57	0.65	0.55	0.65	0.36	0.58	0.57
Stance (Abortion)	0.67	0.71	0.63	**0.73**	0.57	0.67	0.41	0.66	0.65
Stance (Atheism)	0.47	0.77	0.24	**0.81**	0.28	0.78	0.60	0.70	0.71
Movie reviews	**0.92**	0.90	0.89	0.89	0.87	0.90	0.87	0.82	0.75
Legal (Money)	0.77	0.80	0.70	0.78	0.73	0.78	0.69	**0.86**	0.84
Legal (Work)	0.95	0.94	0.88	0.96	**0.97**	0.94	0.89	0.93	0.92
Legal (Crime)	**0.90**	0.83	0.83	0.86	0.86	0.86	0.78	0.84	0.81
Mean	0.77	**0.80**	0.62	0.79	0.66	0.79	0.63	0.75	0.73
Median	0.75	**0.81**	0.66	**0.81**	0.64	0.78	0.66	0.70	0.71

Table 2: Comparison of accuracy of different models. PLR-E methods are trained on 100 samples.

PLR-E, the larger model is not superior and the performance difference decreases with increasing sample size. Comparing Zephyr 3B and Llama2 7B, we do not see a clear winner in next token prediction but also observe a decrease in the performance difference when using PLR-E.

Table 2 shows the performance of PLR-E and next token prediction on the different models and compares it to the next token prediction of GPT-4. The PLR-E models are trained on 100 instances only. If a classification approach performs as well as or better than GPT-4, the cell is underlined. The best performing model on a dataset is highlighted in bold. Table 4 in the appendix replicates this table using the macro F1 score as a performance metric.

Our baseline model uses 4-bit quantisation, next we test whether our results hold for 2-bit and 8-bit quantisations on our key datasets. Figure C.6 (bottom panel) in the appendix shows that for the majority of the classification tasks we observe a substantial improvement in the zero-shot next token predictive accuracy when using the 8-bit model and a corresponding decline in performance when using the 2-bit model. However, when using PLR-E the performance differences between models are less pronounced, particularly at larger sample sizes. Note that the degree of quantisation influences processing speed as well as the memory footprint: going from the 4 to the 8 bit model also increases the average token generation time by 30%.

In summary, having compared models of different lineages, sizes, and degrees of quantisation, we conclude that while these differences affect the accuracy of next token predictions, they make relatively little difference to the performance of the PLR-E method.

2.5 Robustness: In-context few-shot learning

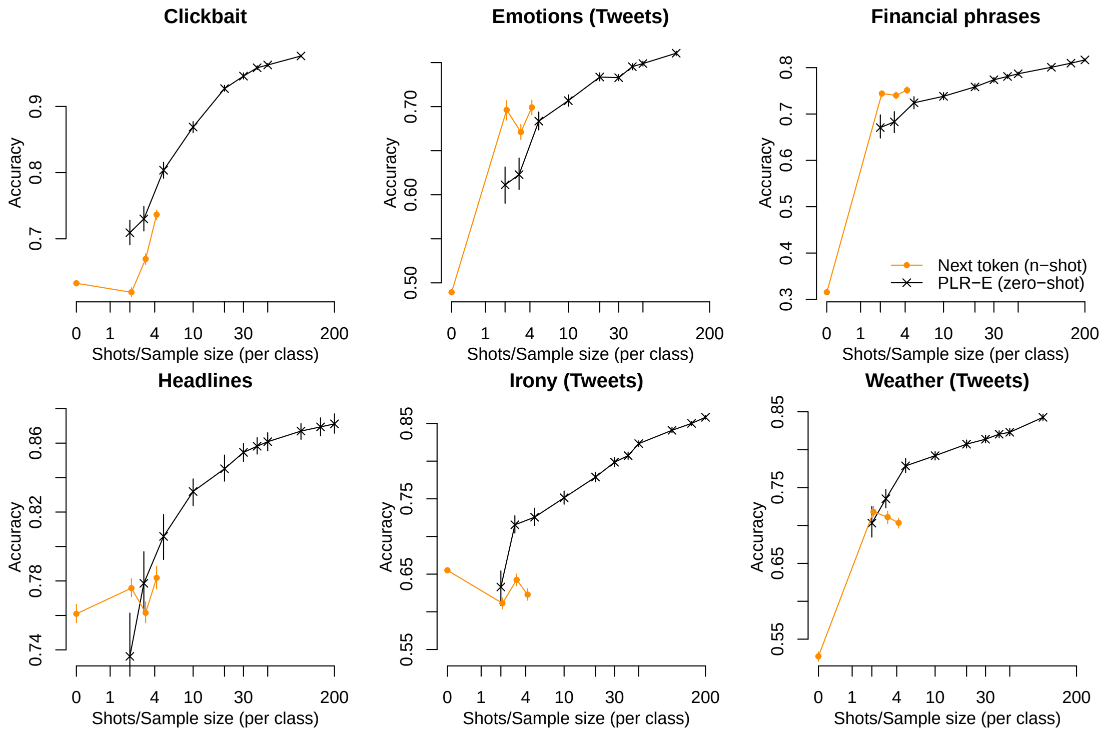

Figure 5: Comparison of zero-shot and few-shot next token prediction of the baseline model and the PLR-E (zero-shot prompts) when calibrated on the same number of examples.

In-context few-shot learning (Brown et al., 2020) works by showing the generative model examples of prompts and responses. We test this calibration approach using $m \in \{2, 3, 5\}$ shots *per class* and compare its performance against PLR trained on exactly the same number of zero-shot text embeddings per class.[3] Due to the high computational cost of few-shot learning, we only conduct the experiments on six of our datasets. Figure 5 shows that few-shot learning leads to substantial performance improvements over the zero-shot case in four of the six datasets. Comparing few-shot learning against PLR-E zero-shot, we only observe a consistently better performance of few-shot learning in two datasets. In both datasets, the performance of few-shot learning seems to have saturated at five shots, suggesting PLR-E will perform better with more training samples.

[3]Note that this differs from our other analyses, where we randomly sample observations across classes.

In Figure C.5 in the appendix, we combine the in-context and PLR-E approaches by calibrating PLR-E on the embeddings produced from few-shot prompts. While few-shot prompts rarely lead to better performance at any point along the learning curves, they introduce a substantial computational overhead at inference time, since in a $k-$class, m-shot-per-class case, the prompt will typically be km times longer, making tens-of-shot cases very costly.

3 Performance comparison: "tens-of-shot" labelled data – small training and test sets

The previous section showed that the PLR-E method can match or exceed the performance of a flagship language model (GPT-4) in the "tens-of-shot" limit, whereas the zero-shot next-token performance of the baseline model is, on average, significantly worse than GPT-4's zero-shot performance.

For ad hoc analysis, simply using GPT-4 will usually be an accurate and low-effort approach. But in any setting where the performance of the model has some value attached to it, the user needs a number of labelled instances to show empirically that their chosen model is indeed a good classifier.

In this section we show that the number of labelled instances needed as a test set to validate the performance of GPT-4 (or any other zero-shot model) is large enough that, if those instances are used instead to train and test the PLR-E method, the methods are then in "competition" (where we define competition as there being no statistically significant difference in their performance). By the time we have provided enough labelled instances to resolve that competition, the number of instances is large enough that the PLR-E method equals or betters GPT-4's performance in all but one dataset.

3.1 Validating zero-shot performance

Even if no actual competitor model exists, there is a natural limiting classification behaviour against which we can evaluate a model: the random baseline accuracy a_r, which is 0.5 in a balanced, two-class problem. Depending on the accuracy of our model, we need some number of labelled data points to statistically show that the model performs better than random. To illustrate this, we model the true accuracy of the classifier $\hat{a}$ as a binomial variable with an unknown success probability $\hat{a}$. Our point estimate for $\hat{a}$ is the observed accuracy in the labelled sample. By estimating the binomial proportion confidence interval using the Wilson score method with continuity correction (Newcombe, 1998), we estimate how many labelled observations are needed so that the confidence interval of the estimate $\hat{a}$ does not overlap with the random classifier. For example, at a sample size 10, we need to observe an accuracy of at least 0.8 to reject the hypothesis that the classifier is not better than random (at $\alpha = 0.1$). At a sample size of 20, the minimum accuracy required to reject the null hypothesis is 0.67.

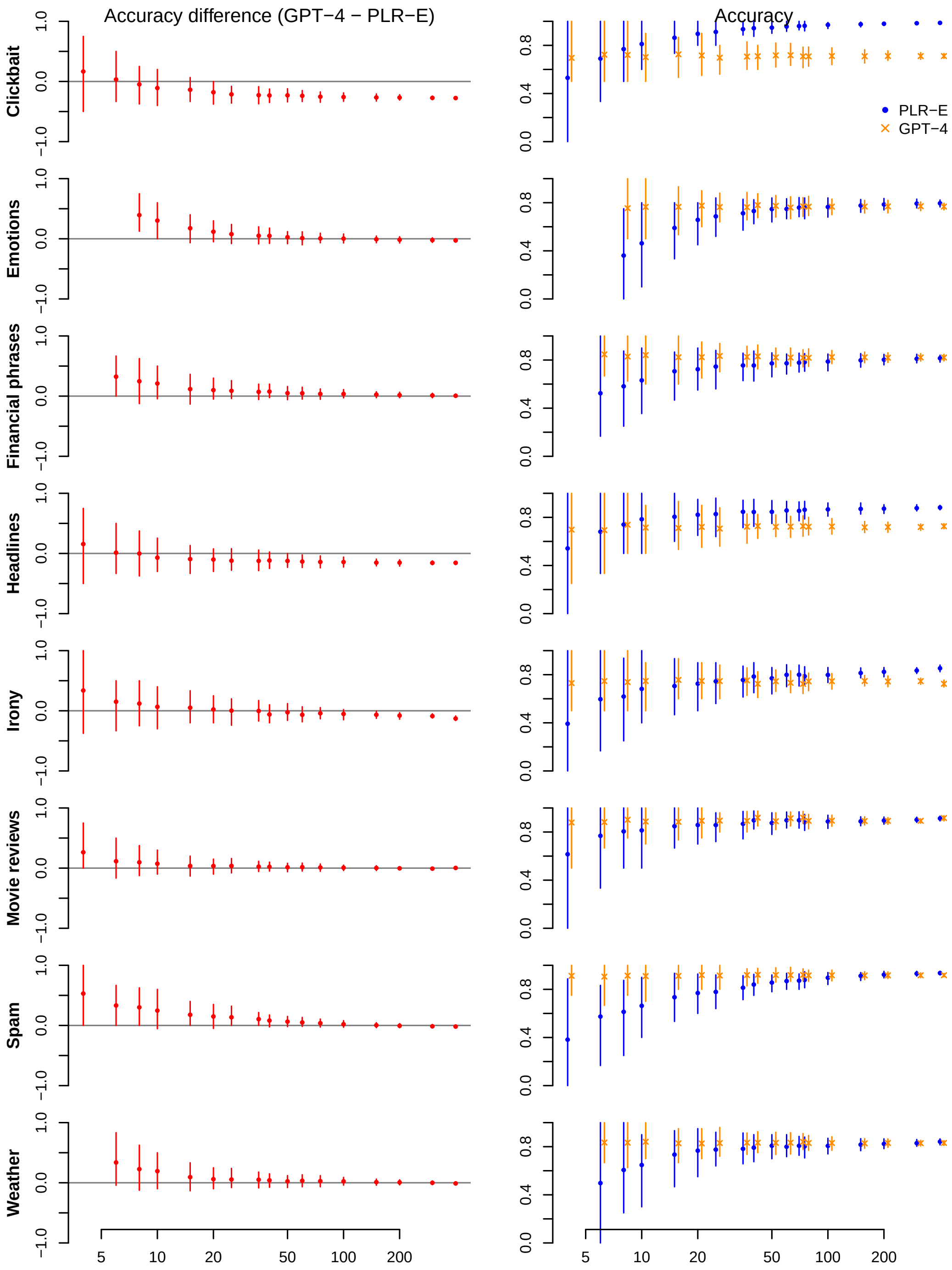

Figure 6: Uncertainty of accuracy estimates for GPT-4 and PLR-E (right panel) and their difference (left panel) as a function of the training set size. We show the 5^{th} to 95^{th} percentile range based on 250 randomly drawn samples.

In practice, the user might have some further operational or regulatory reasons for imposing a tight bound on the maximum uncertainty around the observed model performance, which requires a sizeable set of labelled instances. For example, with 25, 50, 100, and 250 labelled data points, the width of the 95% confidence interval around an observed accuracy of 0.75 is 0.36, 0.25, 0.18, and 0.11, respectively.

When comparing the performance against a random baseline, we only need to estimate $\hat{a}$ and its uncertainty, but if we compare two classifiers the sample size requirements increase due to two unknown parameters. Additionally, a classifier (PLR-E) being trained on the sample introduces a further, and not estimable, variance in its accuracy across samples. In the following, we measure the empirically observed uncertainty of the point estimates of the performance of GPT-4 and PLR-E by repeated sampling.

3.2 Statistical comparison of baseline model and GPT-4

For a given training sample of size n, we use k-fold cross-validation to obtain a point estimate of the performance of PLR-E, where $k = \min(20, n)$. For GPT-4, we just measure the accuracy across all n observations. To estimate the variance of these point estimates, we replicate this procedure on 250 randomly drawn training sets. We reject training samples that do not have at least two data points of each class in the sample, ensuring at least one observation per class in each training set.

The left-hand side of Figure 6 shows the mean and the 5^{th} and 95^{th} percentiles of the *difference* in accuracy of GPT-4 and PLR-E. The right-hand side shows the same statistics for the *levels* of accuracy of the two models. The maximum sample size tested is 400. While GPT-4 generally performs better in the mean than PLR-E at the smallest sample size, that mean cannot be observed in practice. The 5^{th} quantile of the difference in performance is lower or equal to zero in all datasets, telling us that we cannot reliably state in practice that GPT-4 outperforms PLR-E.[4] With increasing sample size, the uncertainty interval around the performance difference shrinks, but so does the GPT-4's advantage. GPT-4 is only the "statistical winner" (i.e. the uncertainty interval of the performance difference is strictly positive at any point of the learning curve) in three of the 17 datasets. The opposite – PLR-E being the statistical winner over GPT-4 – can be observed in seven datasets by the end of the learning curve.

4 Analysis of embedding space

In this section, we aim to better understand the embedding vectors and their usefulness for prediction. The high performance of the regression, albeit penalised, on a 4096-dimensional space with tens of training points, is surprising.

Consider a 2-class dataset with n observations per class. With non-duplicated values on each dimension, there are $(2n)!$ possible permutations of values, of which $2(n!)^2$ separate

[4] Note that at small sample sizes the uncertainty interval around the PLR-E performance estimate is generally larger than that of GPT-4 due to the additional uncertainty created by the model estimation.

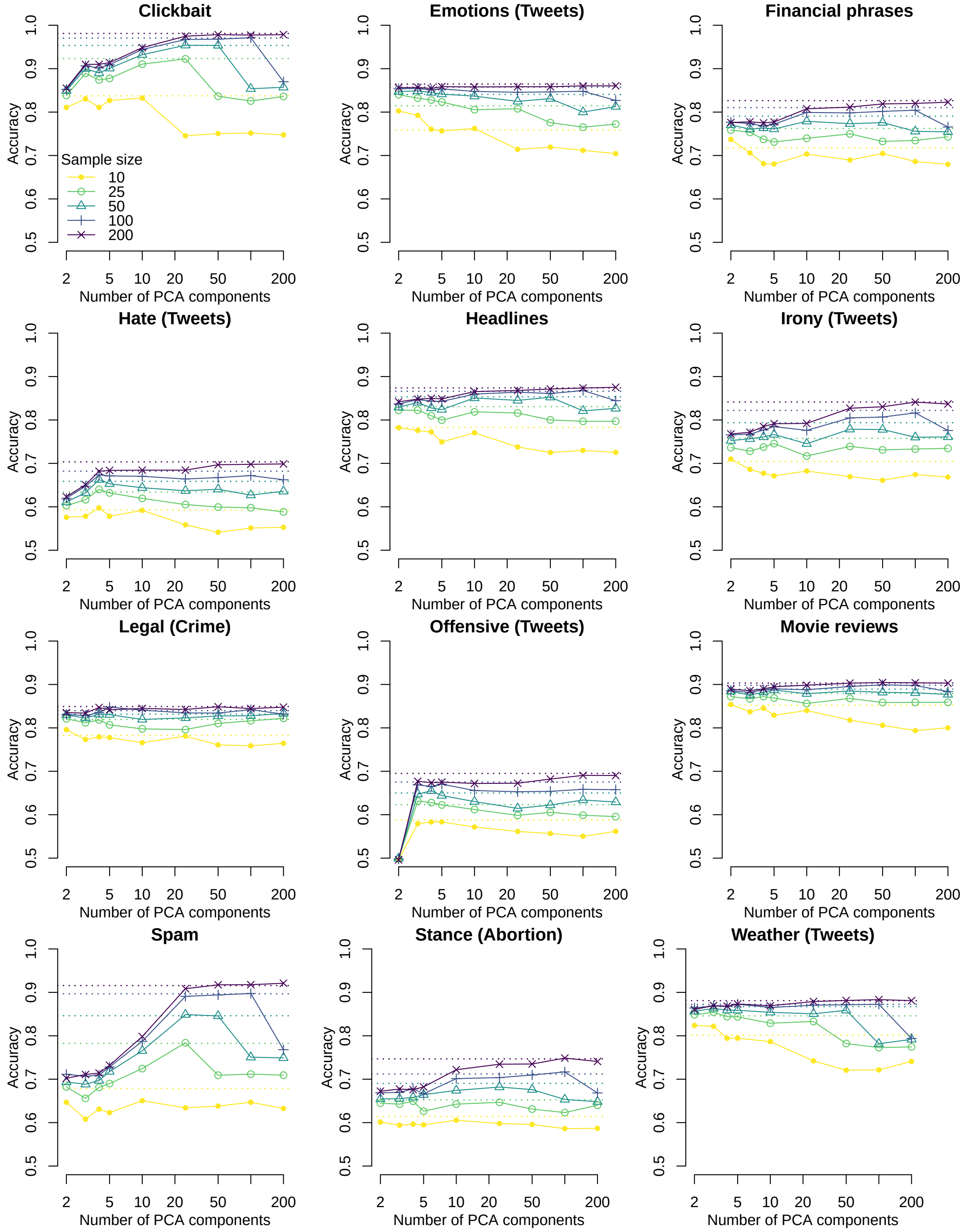

Figure 7: Accuracy as a function of the number of (unnormalised) principal components for a given sample size (colour and symbol).

the classes linearly. Thus, the expected number of embedding dimensions in which the classes separate linearly by chance is $4096 \times 2(n!)^2/(2n)!$ which is above 1 for $n \leq 7$. Therefore, for small training samples, the chance of the model focussing on arbitrary, non-predictive dimensions is high. The main reason for the strong performance of PLR-E on small sample sizes is the high correlation between the embedding vectors, which implies that the model does not need to find a sparse solution – a difficult endeavour in the high-dimensional space.

To simplify the analysis in this section, we bypass multinomial models by converting all multi-class prediction problems into a binary prediction problem by training PLR-E to differentiate the majority class from all other classes.

We show the high degree of collinearity of the embedding space using a principal component analysis (PCA). Figure C.7 in the appendix depicts the cumulative explained variance when using between 1 and 100 components. With the contextualising prefix and suffix (left panel), the first component explains 18–56% (median = 23%) of the variance and the first ten components explain 58–79% (median 63%). Without prefix and suffix, the explained variance decreases substantially to 8–29% (median = 10%) and 24–50% (median = 39%), when using the first and the first 10 components, respectively.

Figure 7 shows the predictive accuracy as a function of the number of principal components for a given sample size. The PCA is fitted to the entire dataset, rather than being restricted to the smaller labelled training subset. Furthermore, we do not normalise the components before we train the ridge regression (PLR-E). In this way, the regularisation implicitly penalises those components more that explain less variance, given these tend to have larger coefficients. For small training samples, low-dimensional PCA can equal (or even slightly exceed) the performance of the baseline approach using all dimensions (shown as dotted horizontal lines). When increasing the training sample, we often observe that the maximum performance is reached when using 30–50 components.

In some datasets we find that the performance deteriorates with a large number of components. This decrease is much more pronounced, with an earlier onset, when we normalise the PCA components to have unit variance (see Figure C.8 in the appendix). This is expected since the linear model will give weight to spurious correlations in the training data by the permutation-based argument at the beginning of this section. In contrast to the collinear embedding matrix, most of the orthogonal PCA components explain little variance in the embeddings and have a small correlation with the class labels. Therefore, 'accidentally' giving them weight in the regression model can have a strong negative impact on performance.

We also test whether we can use sparse models directly on the embedding space. Specifically, we train a Lasso regression to select at most n features with non-zero coefficients and then train an unregularised regression model on that subset of features to undo the shrinkage of the parameters due to the L_1 penalty (see Hastie et al., 2017).

Figure 9 shows the accuracy of the Lasso regression as a function of n for different sample

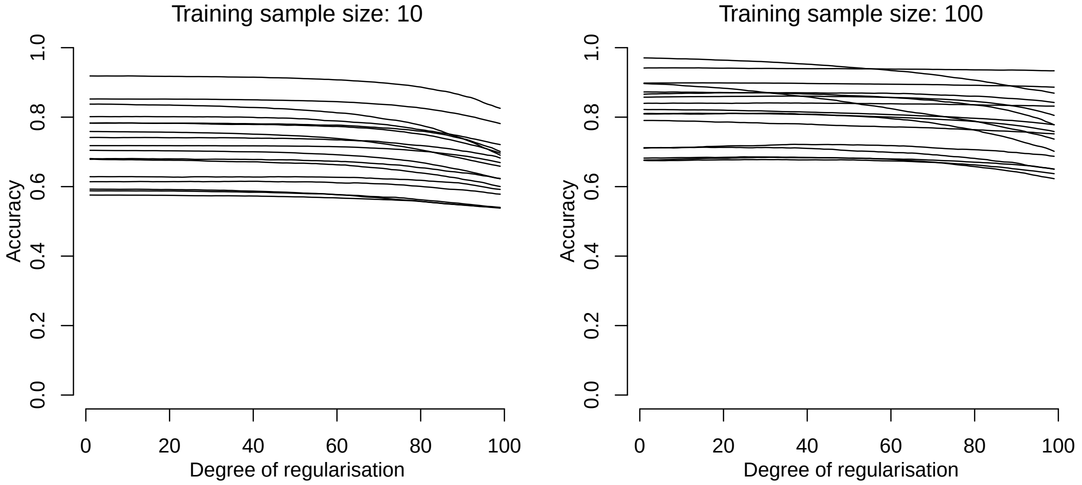

Figure 8: The dependence of the accuracy of PLR-E, trained on 10 instances (left panel) and 100 instances (right panel), on the regularisation parameter.

sizes. The performance of the baseline Ridge model is shown as dotted horizontal lines. When the training sample is small, Lasso generally falls behind the ridge baseline model. But on larger sample sizes, the Lasso regression's performance often approaches or equals that of the ridge regression Surprisingly, this is often achieved with very sparse models that use less than 5 of the 4096 embedding dimensions for prediction. In line with our other results, sparse models are less accurate when we remove the context from the prompt (see Figure C.9 in the appendix).

Finally, we test the regularisation paths which show how sensitive the performance of our PLR-E model is to the regularisation parameter λ (Equation 1). In Figure 8, each line corresponds to a different dataset and shows how the accuracy of PLR-E, trained on 10 instances (left panel) and 100 instances (right panel), changes with the regularisation parameter.[5] At both sample sizes, the performance is relatively robust to the choice of λ. The highest accuracy is consistently achieved with the lowest level of regularisation, which serves as our baseline.

5 Explainability

Explainability is an important aspect of LLM applications. In a commercial context, it is required for robust decision making, and for fulfilling legal and ethical responsibilities. Explainability also helps to pinpoint errors or unintended behaviours. Given that our model works by classifying based on training data, we need to check that the model is not attaching importance to features unrelated to the task, for example grammatical or formatting differences between different classes (Du et al., 2023). Querying a language

[5]From the 100 models on the regularisation path, we omit the model with the highest degree of regularisation as it shrinks all coefficients to 0.

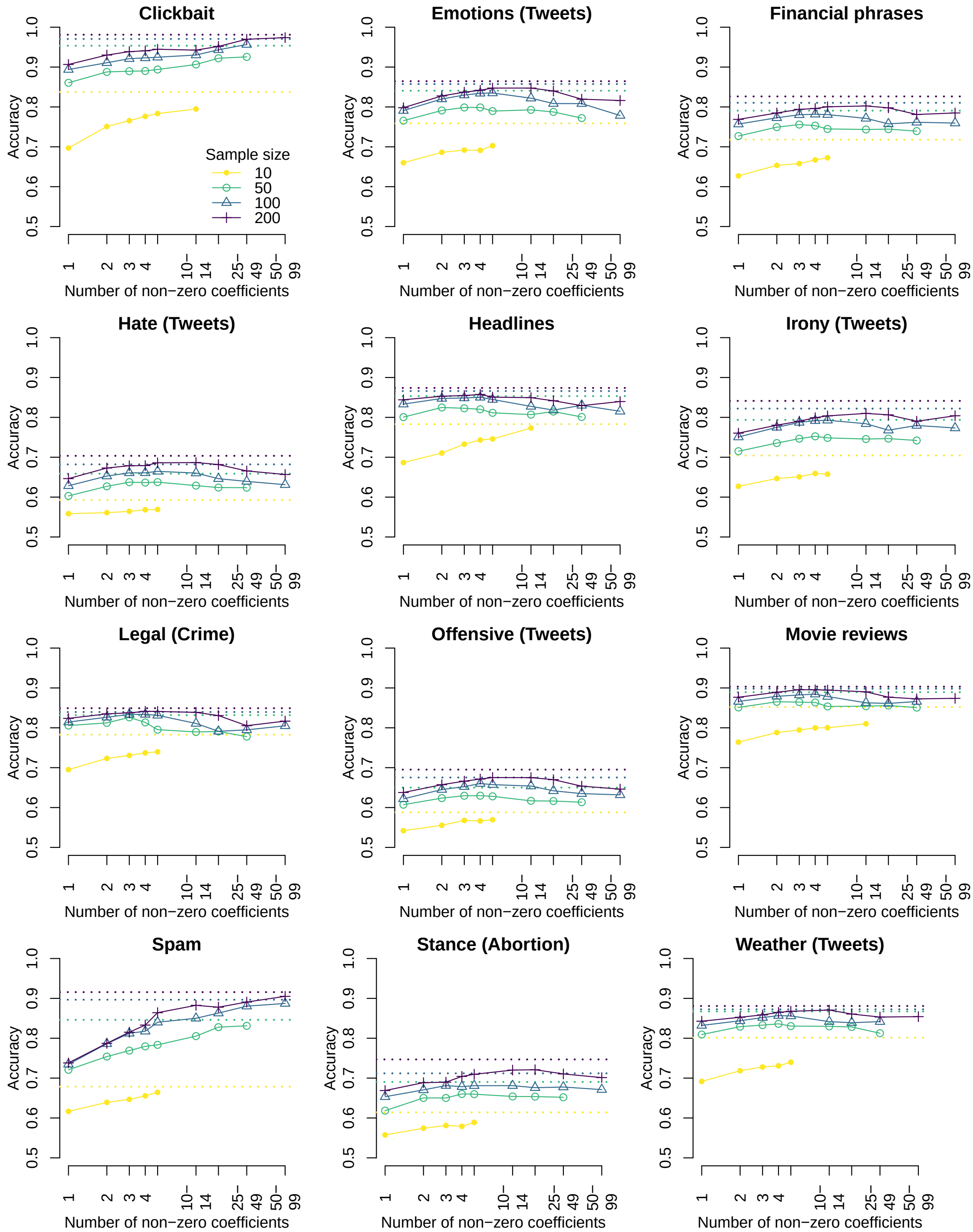

Figure 9: The accuracy of the Lasso regression as a function of the number of non-zero coefficients for different sample sizes (colour and symbol).

model to explain its decision process has been proven unreliable (Sarkar, 2024; Turpin et al., 2023; Agarwal et al., 2024). For non-proprietary models that allow access to their hidden states, probing techniques akin to our PLR-E approach are used in the literature to explain predictions (Luo and Specia, 2024; Zhao et al., 2024).

In this section, we illustrate how PLR-E for text classification can have both high performance and produce explanations which are (a) stable between models created using training sets of the same size, and (b) converge quickly to the explanations given in the large training set limit. We focus on the Financial Phrases dataset and manually annotate a set of 30 examples (15 positive and 15 negative, listed in Appendix B) with our perception of the positivity or negativity at the word level. We quantify the agreement between our annotations and the feature importances produced by the model to show that the model is attaching importance to the 'correct' parts of the phrases.

We train the PLR-E using a 10-dimensional unnormalised PCA. In the large sample limit, the model then underperforms relative to the baseline model (without PCA-based dimensionality reduction) by just 1%.

We take an example phrase, s, and tokenise it into words and symbols (i.e. not the tokenisation used in an LLM, but in the more conventional meaning). We define feature importance of the k^{th} token by deleting it and finding the decision function $d_m(k, s, c)$ for a given class, c and model $m(t)$, a model with training data size t. Then, with $d_m(, s, c)$ being the decision function for the complete phrase, the feature importance of token k is $f_m(k, s, c) = d_m(, s, c) - d_m(k, s, c)$.

The values reported in the figures are found by first constructing all the feature importances for a given example phrase and class $F_m(s, c) = \{f_m(k, s, c) | k \in 1, ..., |s|\}$ and then normalising the standard deviation of the members of $F_m(s, c)$ to 1, obtaining $\tilde{F}_m(s, c)$.

5.1 Stability and convergence

We take the average of feature importances over 20 models trained on different training sets of 200 instances as our baseline, $\tilde{F}^{\infty}(s, c)$. The error for a single model with a smaller training sample size t can be decomposed as $\tilde{F}_{m(t)}(s, c) = [\tilde{F}_{m(t)}(s, c) - \tilde{F}^t(s, c)] + [\tilde{F}^t(s, c) - \tilde{F}^{\infty}(s, c)]$, where addition and subtraction are element-wise in k. The first term is the shift in the feature importances for a particular model $m(t)$ trained on a training set of size t versus the average feature importances for all models with a training set of size t, and the second term is the shift in that average relative to the average feature importance in the large training set limit.

We report the mean absolute values of both the first and second term in Figure 10. After averaging, the first term describes the spread amongst models for a given training set size and the second term refers to their shift relative to the average over models trained on many training instances ($\tilde{F}^{\infty}(s, c)$). Even at the smallest training sample size the deviations are acceptable for important features which empirically have sizes in

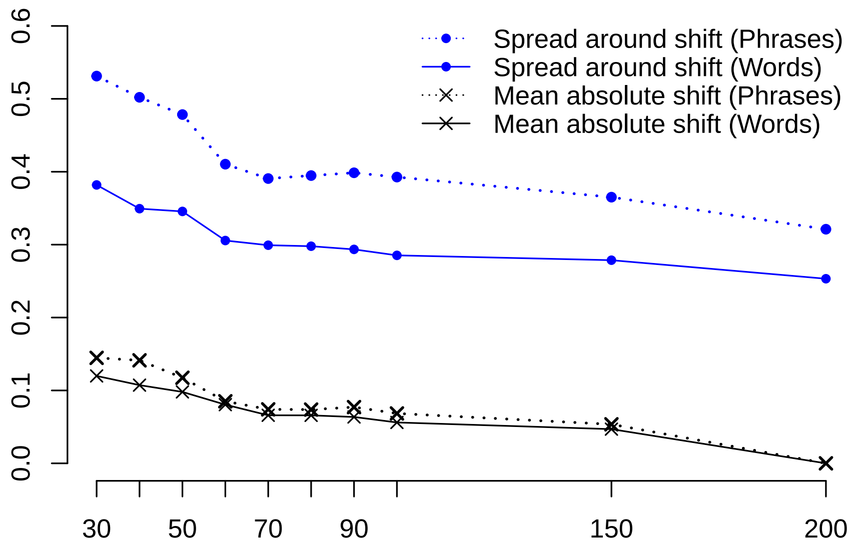

Figure 10: The decomposition of the average absolute deviation from a ground truth explanation of a model trained on different sized training samples.

$\tilde{F}$ greater than 1. This can be seen in four random examples in Figure 11 (the results from 16 other examples are shown in Figures C.10–C.12 in the appendix). The spreads in the feature importances from models trained on different random 30-instance training sets are shown by the black bars and are small in fractional terms, particularly for the important features. The shifts relative to the large training set limit, which are not represented in the figures, are around three times smaller than the spreads.

5.2 Correspondence with human annotations

The two authors independently labelled the positive and negative sentiments of individual words and phrases in the set of 30 examples.

Figure 11 shows the word-level results in four random examples (the results from 16 other examples are shown in Figures C.10–C.12 in the appendix). Visually, it is clear that the model's importance scores are sensible and often align with the annotators' judgement. Even when they diverge, the scores are usually reasonable—for instance, assigning a negative score to the word *compared* in the bottom-right panel of Figure 11. There is a scatter in the points around 0 even for words to which the annotators attached no importance; sometimes there is a 'leakage' effect where words within the same phrase as an important word have additional importance; and finally, there are some anomalies around the end of the example. All are likely due to the effect of removing words agrammatically from the phrase.

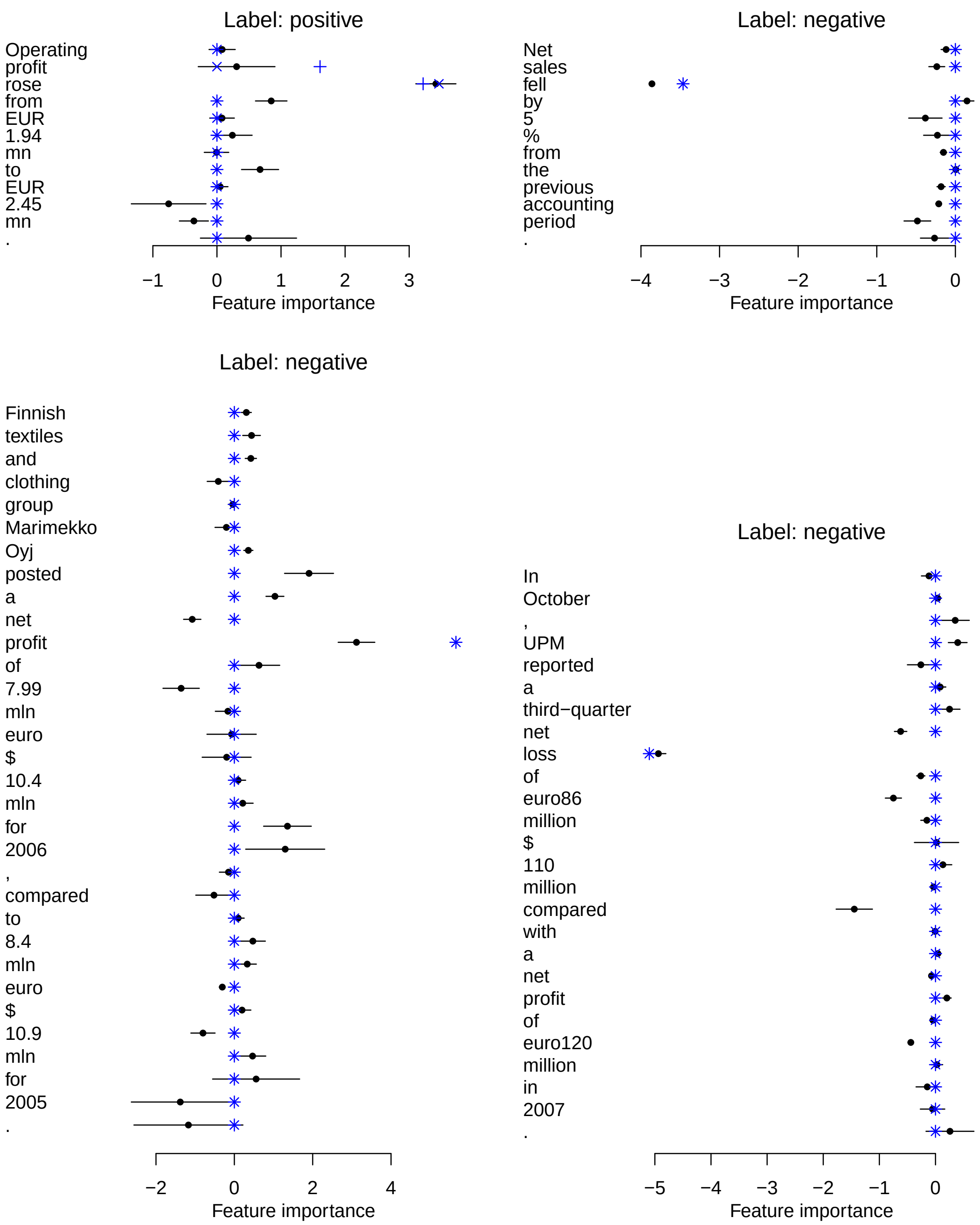

Figure 11: The word-level feature importances according to the PLR-E model (black) are compared with annotated importances by the authors (blue, + and × represent the two annotators).

To investigate the effect of agrammatical word removal, we also implemented a simple algorithm based on a constituency parse proposed by Stanza (Qi et al., 2020). This approach produces grammatical (though not necessarily sensible) phrases by removing entire grammatical units rather than individual words. The mean absolute spreads are about 30% larger and the shifts 10% larger, but the scatter in the points decreases: quantitatively, the Spearman rank correlation of the first annotator's feature importances with those from the model is 0.31 for the word-level explanations and 0.66 for phrase-level explanations. We do not discuss these results further here due to the difficulty in labelling and interpreting these phrase-level explanations, that is, determining a ground-truth as to which points should be non-zero. A number of the perturbations in each case are not sensible, or are sensible and have a sentiment but no longer relate to the original topic. This makes labelling very challenging and uncertain, but this is a broader problem with explainability in language tasks not specific to our work.

We have shown that the PLR-E method can provide explanations which are stable with respect to model and training set size. In this case, at the cost of 10 labelled training instances per class, the model achieves the accuracy of a flagship model, but also benefits from stable and sensible explanations.

6 Conclusion

A penalised regression on embeddings allows local, generative language models to achieve comparable performance to the flagship GPT-4 model in text classification problems. In fact, no more labelled instances are required than are needed for statistically validating the performance of GPT-4. A large number of experiments demonstrated the robustness of our results. An analysis of the embedding space reveals that a handful of the 4096 embedding vectors often suffice to train an accurate linear model on the embeddings. In addition to general advantages of locally hosted models such as privacy, availability, and low cost, we show that our approach enables stable and sensible explanations.

References

Abbas, M., Y. Zhou, P. Ram, N. Baracaldo, H. Samulowitz, T. Salonidis, and T. Chen (2024). Enhancing in-context learning via linear probe calibration. In *International Conference on Artificial Intelligence and Statistics*, pp. 307–315. PMLR.

Achiam, J., S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agarwal, C., S. H. Tanneru, and H. Lakkaraju (2024). Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.

Alain, G. and Y. Bengio (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Anthropic, A. (2024). The Claude 3 model family: Opus, Sonnet, Haiku. *Claude-3 Model Card*.

Ardia, D., K. Bluteau, and K. Boudt (2019). Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values. *International Journal of Forecasting 35*(4), 1370–1386.

Barbieri, F., J. Camacho-Collados, L. Neves, and L. Espinosa-Anke (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Basile, V., C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics 48*(1), 207–219.

Bertsch, C., I. Hull, R. L. Lumsdaine, and X. Zhang (2025). Central bank mandates and monetary policy stances: Through the lens of federal reserve speeches. *Journal of Econometrics*, 105948.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems 33*, 1877–1901.

Campbell, J., R. Ren, and P. Guo (2023). Localizing lying in Llama: Understanding instructed dishonesty on true-false questions through prompting, probing, and patching. *arXiv preprint arXiv:2311.15131*.

Chakraborty, A., B. Paranjape, S. Kakarla, and N. Ganguly (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 9–16. IEEE.

Chen, L., M. Zaharia, and J. Zou (2023). Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Chen, N., N. Wu, S. Liang, M. Gong, L. Shou, D. Zhang, and J. Li (2023). Beyond surface: Probing llama across scales and layers. *arXiv preprint arXiv:2312.04333*.

Chen, Z., S. Gössi, W. Kim, B. Bermeitinger, and S. Handschuh (2023). Finbert-fomc: Fine-tuned finbert model with sentiment focus method for enhancing sentiment analysis of fomc minutes.

Cho, H., H. J. Kim, J. Kim, S.-W. Lee, S.-g. Lee, K. M. Yoo, and T. Kim (2023). Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 12709–12718.

Cook, T. R., S. Kazinnik, A. L. Hansen, and P. McAdam (2023). Evaluating local language models: An application to financial earnings calls. *Available at SSRN 4627143*.

Dettmers, T., R. Svirschevski, V. Egiazarian, D. Kuznedelev, E. Frantar, S. Ashkboos, A. Borzunov, T. Hoefler, and D. Alistarh (2023). SpQR: A sparse-quantized representation for near-lossless LLM weight compression. *arXiv preprint arXiv:2306.03078*.

Du, M., F. He, N. Zou, D. Tao, and X. Hu (2023). Shortcut learning of large language models in natural language understanding. *Communications of the ACM 67*(1), 110–120.

Ellingsen, J., V. H. Larsen, and L. A. Thorsrud (2022). News media versus FRED-MD for macroeconomic forecasting. *Journal of Applied Econometrics 37*(1), 63–81.

European Union (2024). EU AI Act. `https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.html`.

Frank, A. (2010). UCI machine learning repository. *http://archive. ics. uci. edu/ml*.

Gemini Team, Google (2023). Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gössi, S., Z. Chen, W. Kim, B. Bermeitinger, and S. Handschuh (2023). Finbert-fomc: Fine-tuned finbert model with sentiment focus method for enhancing sentiment analysis of fomc minutes. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 357–364.

Gu, A. and T. Dao (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gurnee, W. and M. Tegmark (2023). Language models represent space and time. *arXiv preprint arXiv:2310.02207*.

Hansen, S. and M. McMahon (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. *Journal of International Economics 99*, S114–S133.

Hastie, T., R. Tibshirani, and R. J. Tibshirani (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hubert, P. and L. Fabien (2017). Central bank sentiment and policy expectations.

Jawahar, G., B. Sagot, and D. Seddah (2019). What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Jiang, T., S. Huang, Z. Luan, D. Wang, and F. Zhuang (2023). Scaling sentence embeddings with large language models. *arXiv preprint arXiv:2307.16645*.

Kim, W., J. Spörer, C. L. Lee, and S. Handschuh (2024). Is small really beautiful for central bank communication? evaluating language models for finance: Llama-3-70b, gpt-4, finbert-fomc, finbert, and vader. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 626–633.

Konstantinidis, T., G. Iacovides, M. Xu, T. G. Constantinides, and D. Mandic (2024). Finllama: Financial sentiment classification for algorithmic trading applications. *arXiv preprint arXiv:2403.12285*.

Kostina, A., M. D. Dikaiakos, D. Stefanidis, and G. Pallis (2025). Large language models for text classification: Case study and comprehensive review. *arXiv preprint arXiv:2501.08457*.

Li, X., X. Zhu, Z. Ma, X. Liu, and S. Shah (2023). Are Chatgpt and GPT-4 general-purpose solvers for financial text analytics? an examination on several typical tasks. *arXiv preprint arXiv:2305.05862*.

Lin, J., J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han (2024). Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems 6*, 87–100.

Liu, S.-Y., C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen (2024). Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Liyanage, C., R. Gokani, and V. Mago (2023). GPT-4 as a twitter data annotator: Unraveling its performance on a stance classification task. *Authorea Preprints*.

Loughran, T. and B. McDonald (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance 66*(1), 35–65.

Luo, H. and L. Specia (2024). From understanding to utilization: A survey on explainability for large language models. *arXiv preprint arXiv:2401.12874*.

Ma, S., H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei (2024). The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*.

Malo, P., A. Sinha, P. Korhonen, J. Wallenius, and P. Takala (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology 65*.

Mohammad, S., F. Bravo-Marquez, M. Salameh, and S. Kiritchenko (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international Workshop on Semantic Evaluation*, pp. 1–17.

Mohammad, S., S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry (2016). SemEval-2016 task 6: Detecting stance in tweets. In S. Bethard, M. Carpuat, D. Cer, D. Jurgens, P. Nakov, and T. Zesch (Eds.), *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 31–41.

Muennighoff, N., N. Tazi, L. Magne, and N. Reimers (2022). Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine 17*(8), 857–872.

Pang, B. and L. Lee (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Peng, B., E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV, et al. (2023). Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Pfeifer, M. and V. P. Marohl (2023). Centralbankroberta: A fine-tuned large language model for central bank communications. *The Journal of Finance and Data Science 9*, 100114.

Qi, P., Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Rathje, S., D.-M. Mirea, I. Sucholutsky, R. Marjieh, C. E. Robertson, and J. J. Van Bavel (2024). Gpt is an effective tool for multilingual psychological text analysis. *Proceedings of the National Academy of Sciences 121*(34), e2308950121.

Rui Meng, Ye Liu, S. R. J. C. X. Y. Z. S. Y. (2024). SFR-Embedding-2: Advanced text embedding with multi-stage training.

Šakota, M., M. Peyrard, and R. West (2023). Fly-swat or cannon? cost-effective language model choice via meta-modeling. *arXiv preprint arXiv:2308.06077*.

Sarkar, A. (2024). Large language models cannot explain themselves. *arXiv preprint arXiv:2405.04382*.

Savelka, J., K. D. Ashley, M. A. Gray, H. Westermann, and H. Xu (2023). Can GPT-4 support analysis of textual data in tasks requiring highly specialized domain expertise? *arXiv preprint arXiv:2306.13906*.

Shah, A., S. Paturi, and S. Chava (2023). Trillion dollar words: A new financial dataset, task & market analysis. *arXiv preprint arXiv:2305.07972*.

Sinha, A. and T. Khandait (2021). Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pp. 589–601. Springer.

Stability AI (2024). StableLM Zephyr 3B. `https://huggingface.co/stabilityai/stablelm-zephyr-3b`. Fine-tuned version of StableLM 3B with alignment tuning for chat and instruction tasks.

Sun, M., Z. Liu, A. Bair, and J. Z. Kolter (2023). A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Thorsrud, L. A. (2020). Words are the new numbers: A newsy coincident index of the business cycle. *Journal of Business & Economic Statistics 38*(2), 393–409.

Toney-Wails, A., C. Schoeberl, and J. Dunham (2024). Ai on ai: Exploring the utility of gpt as an expert annotator of ai publications. *arXiv preprint arXiv:2403.09097*.

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Turpin, M., J. Michael, E. Perez, and S. Bowman (2023). Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems 36*, 74952–74965.

Vajjala, S. and S. Shimangaud (2025). Text classification in the llm era–where do we stand? *arXiv preprint arXiv:2502.11830*.

Van Hee, C., E. Lefever, and V. Hoste (2018). Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 39–50.

Wang, L., N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei (2023). Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Xiao, S., Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie (2024). C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649.

Zampieri, M., S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Zhang, B., K. Chang, and C. Li (2024). Simple techniques for enhancing sentence embeddings in generative language models. *arXiv preprint arXiv:2404.03921*.

Zhang, W., Y. Deng, B. Liu, S. J. Pan, and L. Bing (2023). Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

Zhao, H., H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du (2023). Explainability for large language models: A survey.

Zhao, H., H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du (2024). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology 15*(2), 1–38.

Zhu, F., D. Dai, and Z. Sui (2024). Language models understand numbers, at least partially. *arXiv preprint arXiv:2401.03735*.

Appendix

A Datasets

Financial Phrases. The dataset contains 4,843 English financial news sentences categorised into positive, negative, and neutral sentiment by 16 annotators. The dataset was assembled by Malo et al. (2014) and can be downloaded from Huggingface. We sample of random subset of 4,838 of the sentences of this dataset.

Central Banking. This dataset contains 2,480 statements of the Federal Open Market Committee based on speeches, meeting minutes and press conferences. The statements have been labelled as having a hawkish, dovish, or a neutral monetary policy stance. The annotators were given a detailed guide that defined hawkishness and dovishness for eight different categories (economic status, dollar value change, energy/house prices, foreign nations. Fed expectations/actions/assets, money supply, keywords/phrases, labour), which illustrates the complexity of the classification problem. This dataset has been assembled by Shah et al. (2023) and is made available on GitHub GitHub.

Clickbait. This dataset contains headlines of 32,000 articles. The prediction task is to predict whether a headline is *clickbait* or not, where clickbait is defined as a catchy headline that lures readers to click on a link. This dataset has been assembled by Chakraborty et al. (2016) and is made available from the one of the authors' GitHub page. We sample of random subset of 800 of the sentences of this dataset, balancing the two classes.

Weather (Tweets). The dataset contains 1000 tweets that were assigned to the labels *positive sentiment, negative sentiment, neutral sentiment, I can't tell, tweet not related to weather* by 20 human annotators. The data can be downloaded from data.world. We removed the 238 tweets in the categories *I can't tell, tweet not related to weather* from the dataset.

Headlines. The dataset contains 10,570 news headlines on gold in its role as a commodity. Human annotators have annotated the price sentiment (*positive, negative, neutral, none*) and six boolean variables reflecting the content of the tweet: *price direction up, price direction constant, price direction down, asset comparison, past information, future information.* The dataset has been assembled by Sinha and Khandait (2021) and can be downloaded from Kaggle. We use the mutually exclusive variables *past information* and *future information* as our binary classes. We randomly sample a subset of 639 headlines, balancing the two classes.

Tweet Eval: Emotions, Irony, Hate, Offensive, and Stance on feminism, atheism and abortion. Tweet Eval (Barbieri et al., 2020) is the repository of a benchmark study and contains seven heterogeneous tweet classification tasks from which we used five in our study, all of which can be downloaded from GitHub.

Emotions. The dataset contains 5,052 tweets, each expressing one of four emotions: *anger, joy, sadness, optimism.* The data was assembled by Mohammad et al. (2018). We randomly sample 4,653 tweets from the dataset.

Irony. The dataset contains 4,601 tweets which are either ironic or not. The dataset was assembled by Van Hee et al. (2018). We randomly sample 2,526 tweets from the data, balancing the two classes.

Stance (Abortion, Atheism, Feminism). The dataset contains 4,870 annotated tweets that express a stance (favour, against, neutral) towards six targets in the United States: atheism, feminism, abortion, climate change, Hillary Clinton. We use the first three in our analysis, respectively subsampling 867, 681, and 881 tweets. The dataset was assembled by Mohammad et al. (2016)

Offensive. The dataset contains 14,100 tweets which human annotators have labelled as offensive or not. The dataset was assembled by Zampieri et al. (2019). We randomly sampled 2,014 sentences, balancing the classes.

Hate. The dataset contains 13,000 tweets which human annotators have labelled as hate speech or not. The dataset was assembled by Basile et al. (2019). We randomly sampled 2,007 sentences, balancing the classes.

Spam. This dataset contains 1,956 comments under YouTube videos. The task is to identify whether these are spam or not. The dataset was assembled by T.C. Alberto and J. V. Lochter and is available from the UCI machine learning repository (Frank, 2010) under the name *YouTube Spam Collection*. We randomly sampled 743 sentences, balancing the classes.

Movie reviews. This dataset contains 10,662 sentences from Rotten Tomatoes movie reviews with either positive or negative sentiment. The dataset was assembled by Pang and Lee (2005) and can be downloaded from Huggingface. We randomly sampled 2,000 sentences, balancing the classes.

Legal. The dataset contains 2,811 legal questions by laypeople that have been assigned to non-mutually-exclusive labels (drawn from the Legal Issues Taxonomy) using crowdsourcing. We create three binary classification tasks based on the labels: payment and debt (Money), employment and job (Work) and criminal issues (Crime). We respectively sub-sample 728, 724, and 648 legal questions for the three tasks, balancing the classes. The legal questions are often long, which is why we only use the first few sentences until 100 tokens are reached. The dataset is named *Learned Hands Data* and has been assembled by the Legal Innovation & Technology Lab' of Suffolk Law School and Stanford's Legal Design Lab, with the former institute providing a download link.

B Sentences classified for explainability

These are the sentences which were classified for the explainability section. The first 15 have positive sentiment, and the rest have negative sentiment. Please contact the authors to discuss the segmentation and labelling, and the data produced from that exercise.

1. The company 's scheduled traffic , measured in revenue passenger kilometres RPK , grew by just over 2 % and nearly 3 % more passengers were carried on scheduled flights than in February 2009 .
2. Finnish pharmaceuticals company Orion reports profit before taxes of EUR 70.0 mn in the third quarter of 2010 , up from EUR 54.9 mn in the corresponding period in 2009 .
3. O'Leary 's Material Handling Services , located in Perth , is the leading company in Western Australia that supplies , installs and provides service for tail lifts .
4. Net cash flow from operations is expected to remain positive .
5. In the reporting period , the company 's operating profit grew by 43.2 % to EUR 6 million .
6. Finnish Metso Paper has been awarded a contract for the rebuild of Sabah Forest Industries ' (SFI) pulp mill in Sabah , Malaysia .
7. The agreement strengthens our long-term partnership with Nokia Siemens Networks .

8. The diluted loss per share narrowed to EUR 0.27 from EUR 0.86 .
9. Operating profit rose from EUR 1.94 mn to EUR 2.45 mn .
10. Nokia bought Chicago-based Navteq in 2008 , acquiring a maps database to compete with Google s maps as well as with navigation device companies such as TomTom NV and Garmin Ltd. .
11. Sales for the Department Store Division increased by 15 % and sales for the clothing store subsidiary Seppala increased by 8 % Meanwhile sales for Hobby Hall decreased by 12 % .
12. However , the total orders received will still be above last year s levels .
13. We are very proud to be able to use this kind of innovative mobile service for voting in elections .
14. Finnish electronics manufacturer PKC Group Oyj (OMX Helsinki : PKC1V) said on Wednesday (31 December) that it has completed the acquisition of MAN Nutzfahrzeuge AG 's cable harness business from MAN Star Trucks & Buses Spolka zoo in Poland .
15. ‘ For Nordea , moving into the new headquarters signifies the beginning of a new era .
16. However , the suspect stole his burgundy Nissan Altima .
17. Finnish Bank of +àland reports its operating profit fell to EUR 4.9 mn in the third quarter of 2007 from EUR 5.6 mn in the third quarter of 2006 .
18. stores 16 March 2010 - Finnish stationery and gift retailer Tiimari HEL : TII1V said yesterday that it will cut a total of 28 jobs in its units Tiimari Retail Ltd and Gallerix Finland Ltd as a result of the closure of shops .
19. In October , UPM reported a third-quarter net loss of euro86 million $ 110 million compared with a net profit of euro120 million in 2007 .
20. Employing 112 in Finland and 280 abroad , the unit recorded first-quarter 2007 sales of 8.6 mln eur , with an operating loss of 1.6 mln eur .
21. Net sales fell by 5 % from the previous accounting period .
22. The company plans to close two of the three lines at the plant , where some 450 jobs are under threat .
23. Operating profit , excluding non-recurring items , totaled EUR 0.2 mn , down from EUR 0.8 mn in the corresponding period in 2006 .
24. The Elcoteq group recently announced that the last three months of the previous year brought to it a major loss of more than half a billion kroons (EUR 32 mln) for the fifth quarter running .
25. More than 14,000 customers were left powerless .
26. Operating profits in the half were 0.8 m , down from 0.9 m as Glisten invested in the brand and the management team .
27. Cash flow after investments amounted to EUR45m , down from EUR46m .
28. The majority of the company 's personnel in Finland is temporarily laid off from one to six weeks in the period from February to June 2009 period .

29. Finnish textiles and clothing group Marimekko Oyj posted a net profit of 7.99 mln euro $ 10.4 mln for 2006 , compared to 8.4 mln euro $ 10.9 mln for 2005 .

30. The operating loss amounted to EUR 0.8 mn , compared to a profit of EUR 3.9 mn a year earlier .

C Additional results

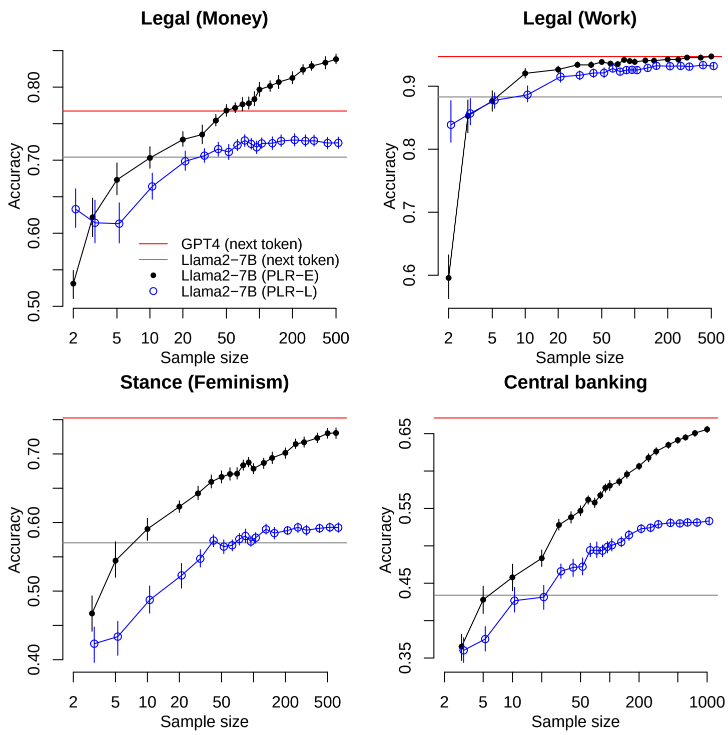

Figure C.1: Continues Figure 1. The accuracies of the zero-shot next token text predictions from GPT-4 and Llama2-7B, along with with the learning curves for the PLR-L and PLR-E methods applied to our baseline model (Llama2-7B q4.0).

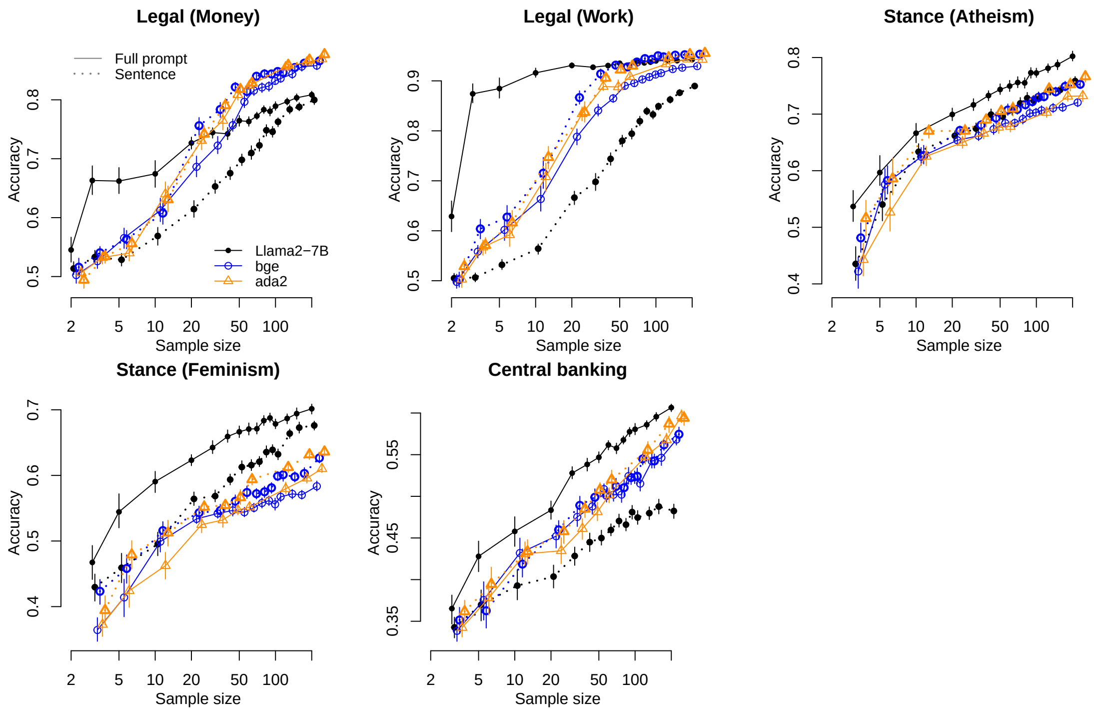

Figure C.2: Continues Figure 3. The accuracy of PLR-E when trained on embeddings from different models and prompts.

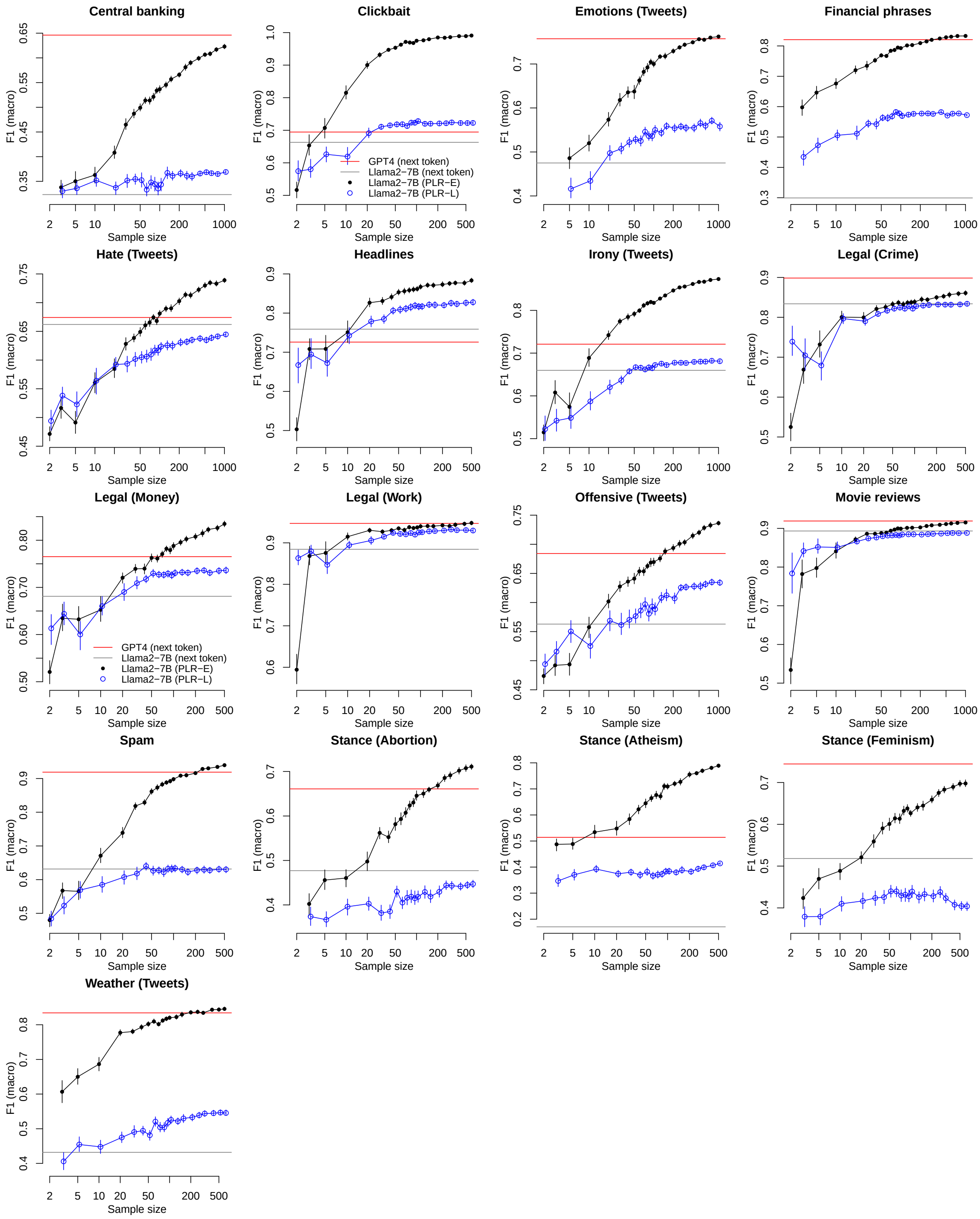

Figure C.3: The F1 macro score of the zero-shot next token text predictions from GPT-4 and Llama2-7B, along with with the learning curves for the PLR-L and PLR-E methods applied to our baseline model (Llama2-7B q4.0). This is the analogue, using F1 instead of accuracy, of Figures 1 and C.1.

Type	Prefix	Suffix
Baseline	The following sentence contains financial news	Does the sentence have (a) positive, (b) negative, (c) neutral sentiment?
No a,b,c,d	The following sentence contains financial news	Does the sentence have positive, negative, neutral sentiment? Answer with a single word.
No prefix	–	Does the sentence have positive, negative, neutral sentiment?
No choices	–	What is the sentiment of the sentence?
Minimal instructions	Sentiment of	–
Distortions	The following sentence contains financial news	Does the sentence have (a) positive, (b) negative, (c) neutral sentiment? X9asd7bV
No instructions	–	–
No instructions + distortion	–	X9asd7bV

Table 3: Different types of prompts are shown in a case where the possible answers are positive, negative and neutral. 'X9asd7bV' is an example random alphanumeric string of that form which is inserted into the prompt in that position.

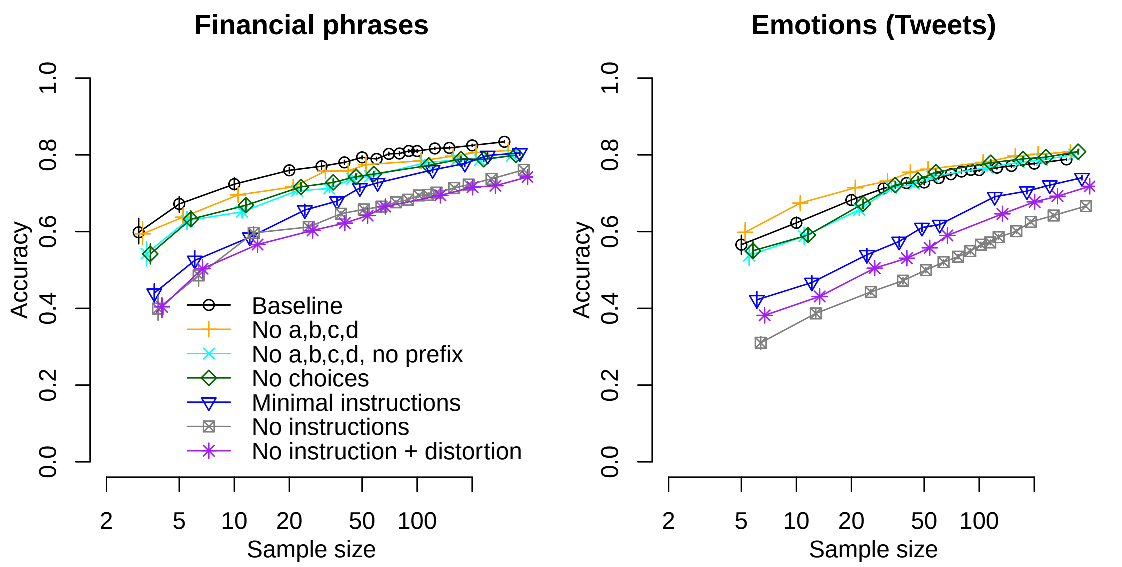

Figure C.4: Learning curves for surrounding prompts with different prefix and suffix. See Table 3 for the specifications of the prompts in the legend.

Datasets	GPT-4	Llama2 7B		Llama2 13B		Zephyr 3B		ada2	bge
	Token	PLR-E	Token	PLR-E	Token	PLR-E	Token	PLR-E	PLR-E
Central banking	**0.65**	0.54	0.32	0.56	0.42	0.56	0.24	0.50	0.45
Clickbait	0.69	**<u>0.97</u>**	0.67	**<u>0.97</u>**	<u>0.75</u>	<u>0.96</u>	0.40	<u>0.96</u>	<u>0.94</u>
Headlines	0.73	**<u>0.87</u>**	<u>0.76</u>	<u>0.86</u>	0.62	<u>0.86</u>	0.45	<u>0.84</u>	<u>0.79</u>
Spam	0.92	0.90	0.64	0.89	0.64	<u>0.92</u>	0.55	**<u>0.93</u>**	0.91
Financial phrases	**0.82**	0.79	0.30	0.80	0.60	0.81	0.68	0.54	0.52
Weather (Tweets)	**0.83**	0.82	0.43	0.81	0.67	**<u>0.83</u>**	0.77	0.78	0.75
Irony (Tweets)	0.72	**<u>0.82</u>**	0.66	<u>0.81</u>	0.50	0.66	0.52	0.59	0.59
Emotions (Tweets)	**0.76**	0.70	0.48	0.68	0.61	0.74	0.65	0.53	0.52
Offensive (Tweets)	0.68	0.67	0.56	0.66	0.62	**<u>0.70</u>**	**<u>0.70</u>**	0.65	0.66
Hate (Tweets)	0.67	<u>0.68</u>	0.66	0.60	0.64	**<u>0.71</u>**	<u>0.67</u>	<u>0.69</u>	<u>0.67</u>
Stance (Feminism)	**0.74**	0.63	0.52	0.61	0.45	0.58	0.36	0.44	0.43
Stance (Abortion)	0.66	0.65	0.48	**<u>0.70</u>**	0.51	0.55	0.37	0.52	0.51
Stance (Atheism)	0.51	<u>0.71</u>	0.17	**<u>0.77</u>**	0.28	<u>0.70</u>	0.48	<u>0.52</u>	<u>0.54</u>
Movie reviews	**0.92**	0.90	0.89	0.89	0.87	0.90	0.86	0.82	0.75
Legal (Money)	0.77	<u>0.80</u>	0.69	<u>0.77</u>	0.73	<u>0.78</u>	0.68	**<u>0.85</u>**	<u>0.83</u>
Legal (Work)	0.95	0.94	0.88	<u>0.96</u>	**<u>0.97</u>**	0.94	0.89	0.93	0.91
Legal (Crime)	**0.90**	0.83	0.83	0.86	0.86	0.86	0.77	0.84	0.81
Mean	0.76	**<u>0.78</u>**	0.59	**<u>0.78</u>**	0.63	<u>0.77</u>	0.59	0.70	0.68
Median	0.74	**<u>0.80</u>**	0.64	**<u>0.80</u>**	0.62	<u>0.78</u>	0.65	0.69	0.67

Table 4: Comparison of the F1 macro score of different models. PLR-E methods are trained on 100 samples. This is the analogue, using F1 instead of accuracy, of Table 2.

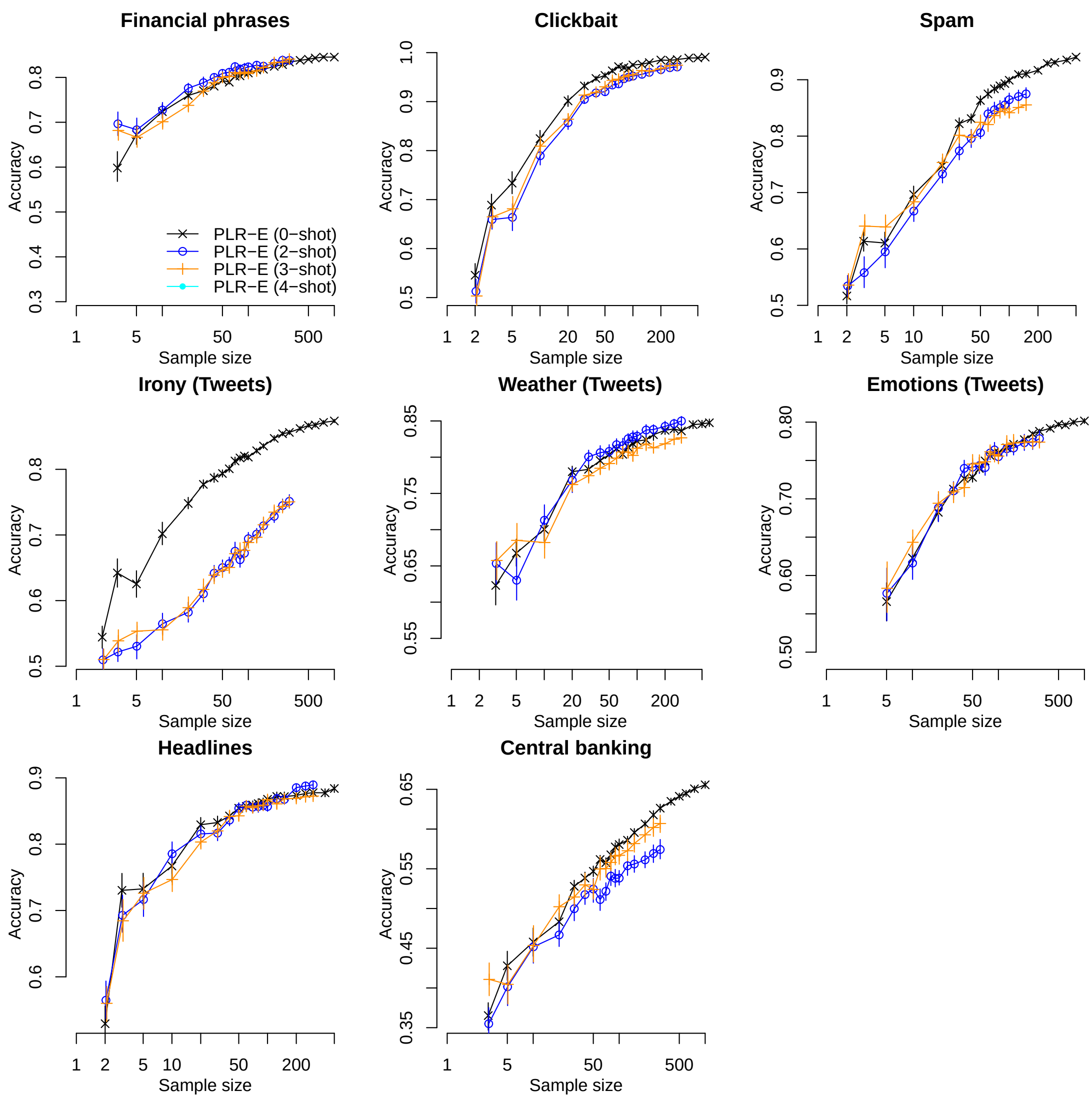

Figure C.5: Using PLR-E on the embeddings from zero- and few-shot prompting of our baseline model.

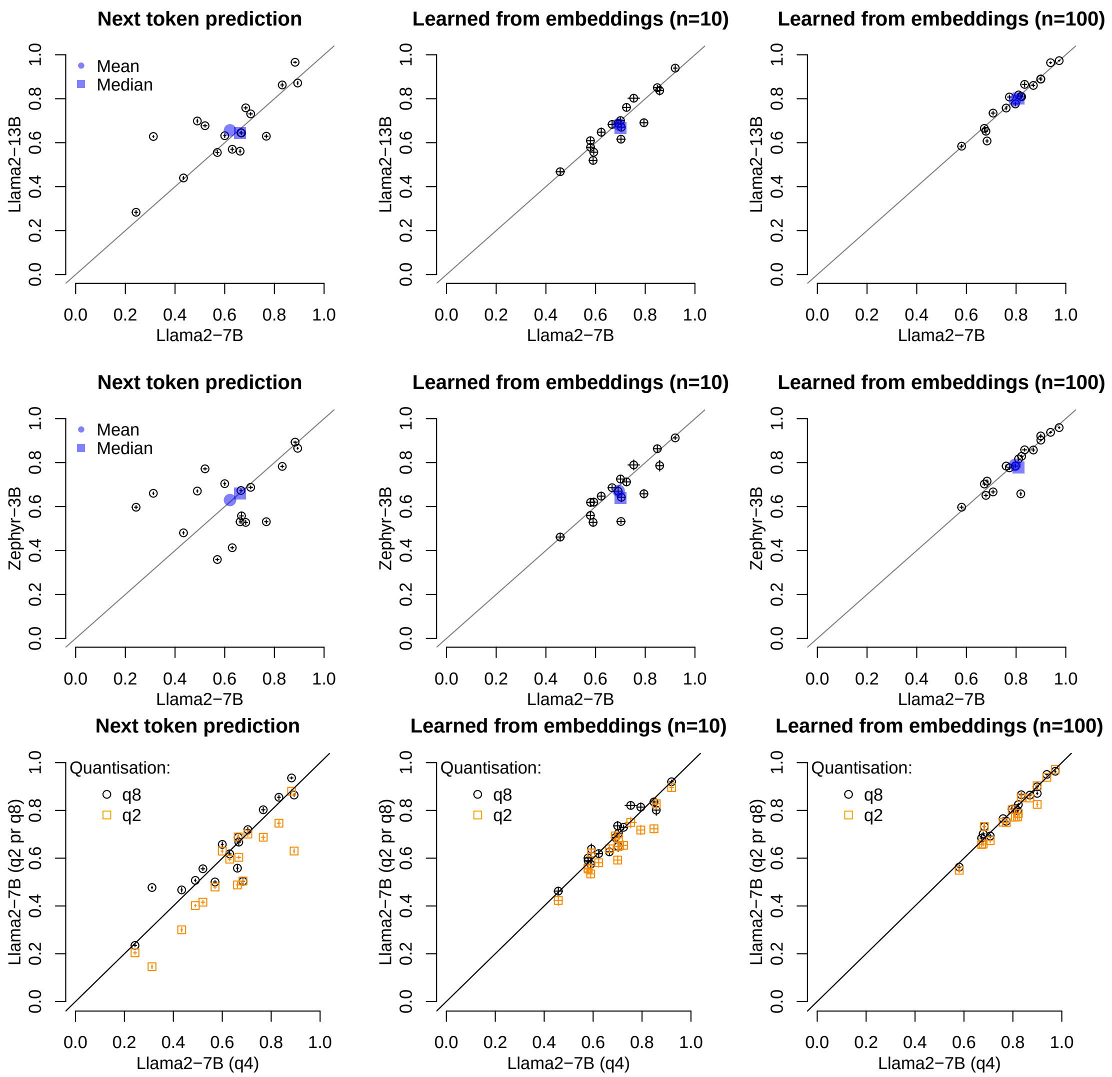

Figure C.6: The accuracy of PLR-E at different sample sizes when using the embeddings from the baseline model is compared against PLR-E using embeddings from other generative models (top two panels) or different quantisations of the baseline model (bottom panel).

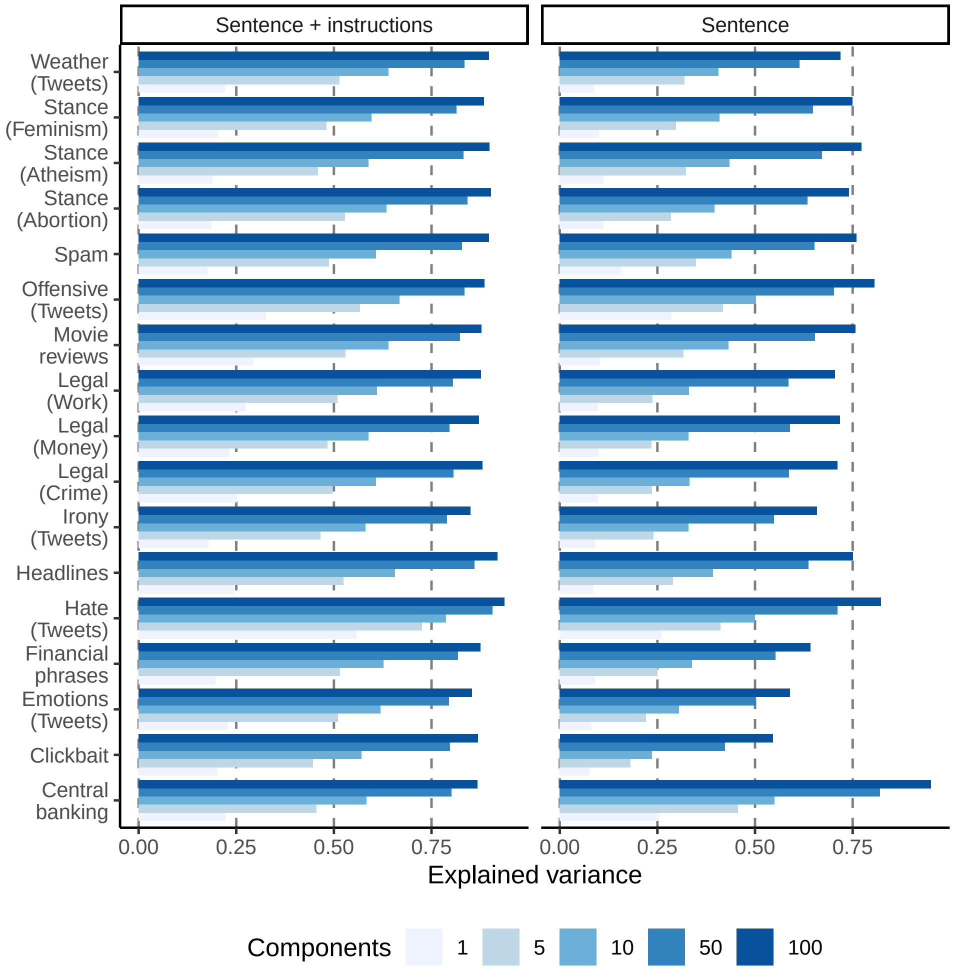

Figure C.7: The cumulative explained variance of PCAs on the embeddings produced by our baseline model with (left panel), and without (right panel), a surrounding prompt.

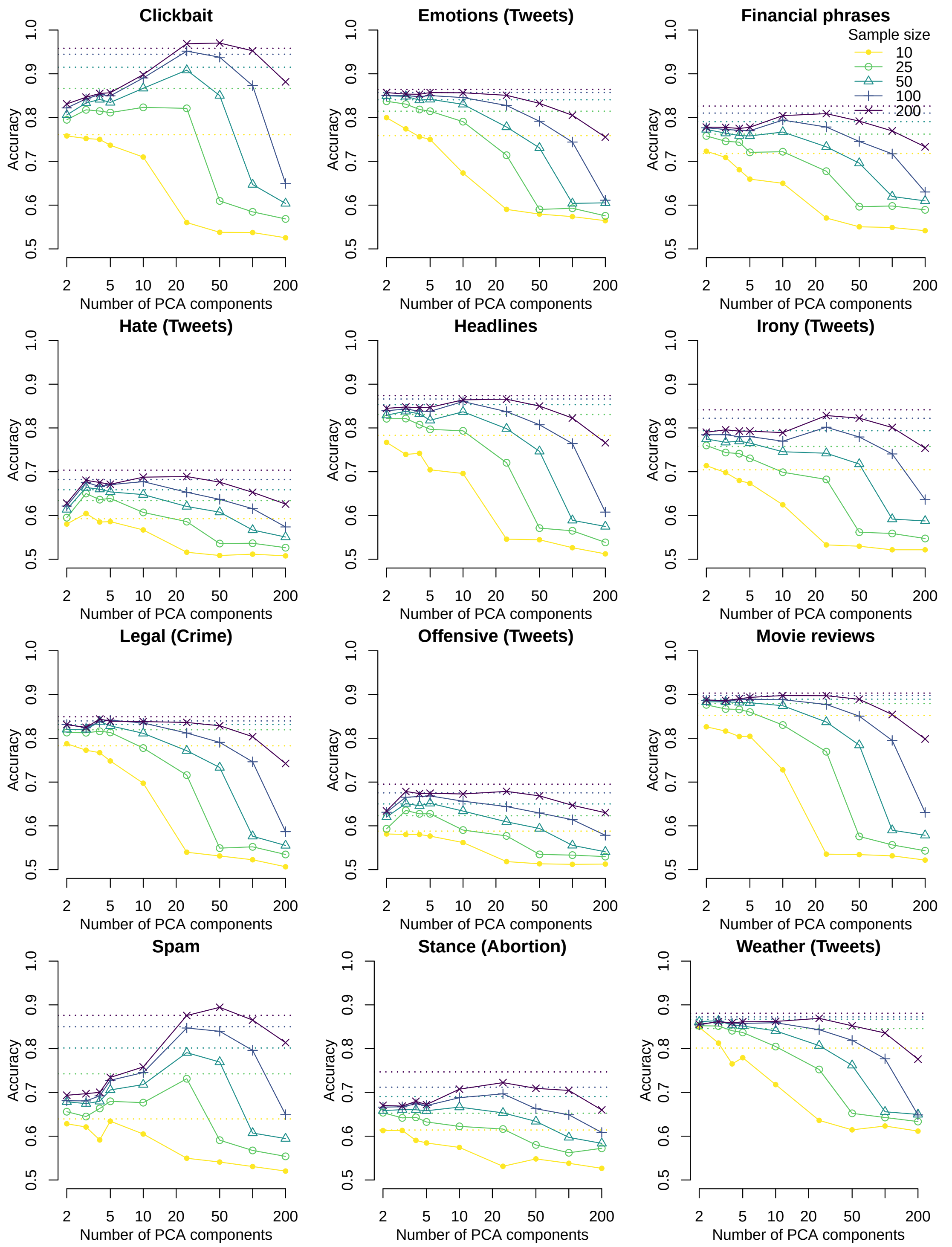

Figure C.8: Accuracy as a function of the number of (normalised) principal components for a given sample size (colour and symbol), c.f. Figure 7.

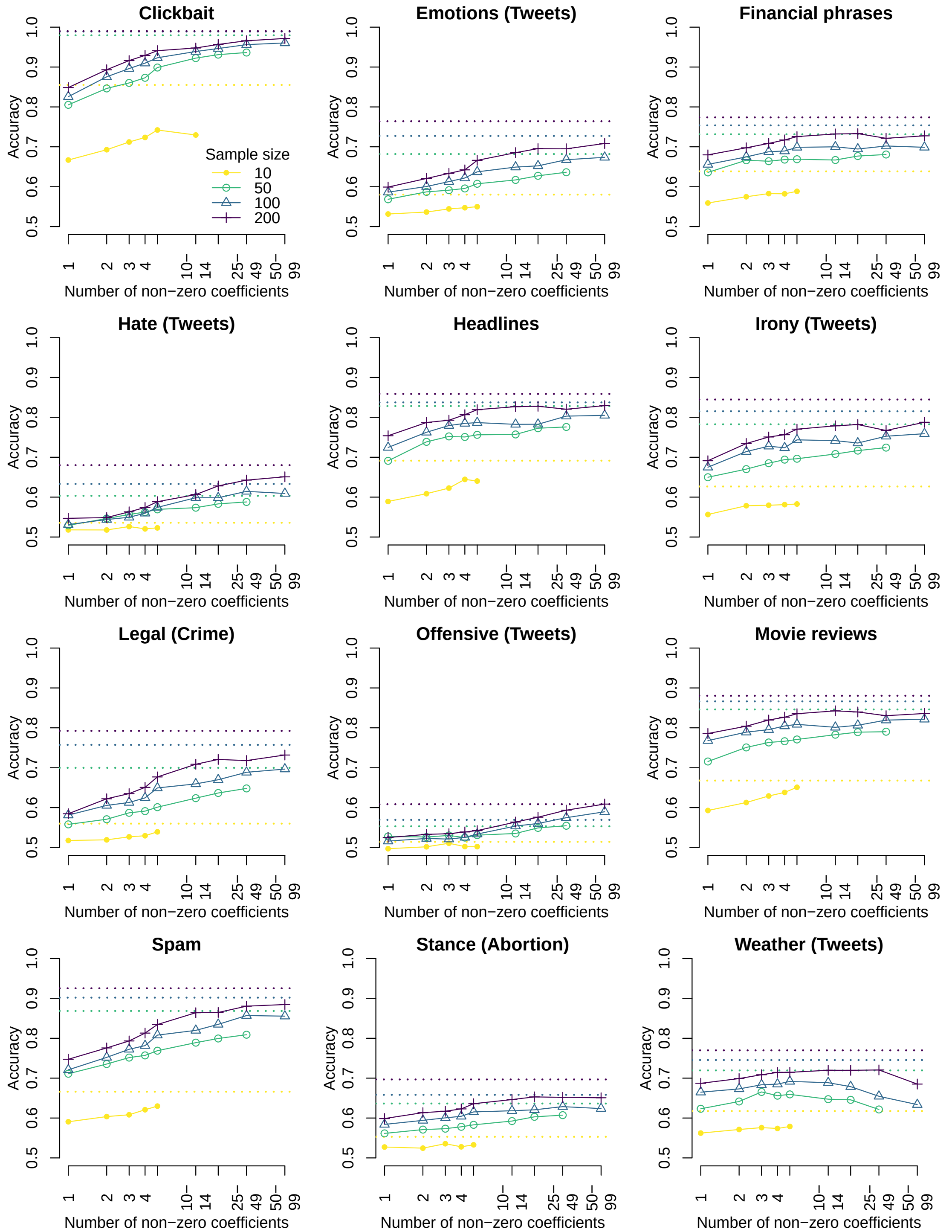

Figure C.9: The accuracy of the Lasso regression as a function of the number of non-zero coefficients for different sample sizes (colour and symbol). The embeddings are produced without the surrounding prompt, c.f. Figure 9.

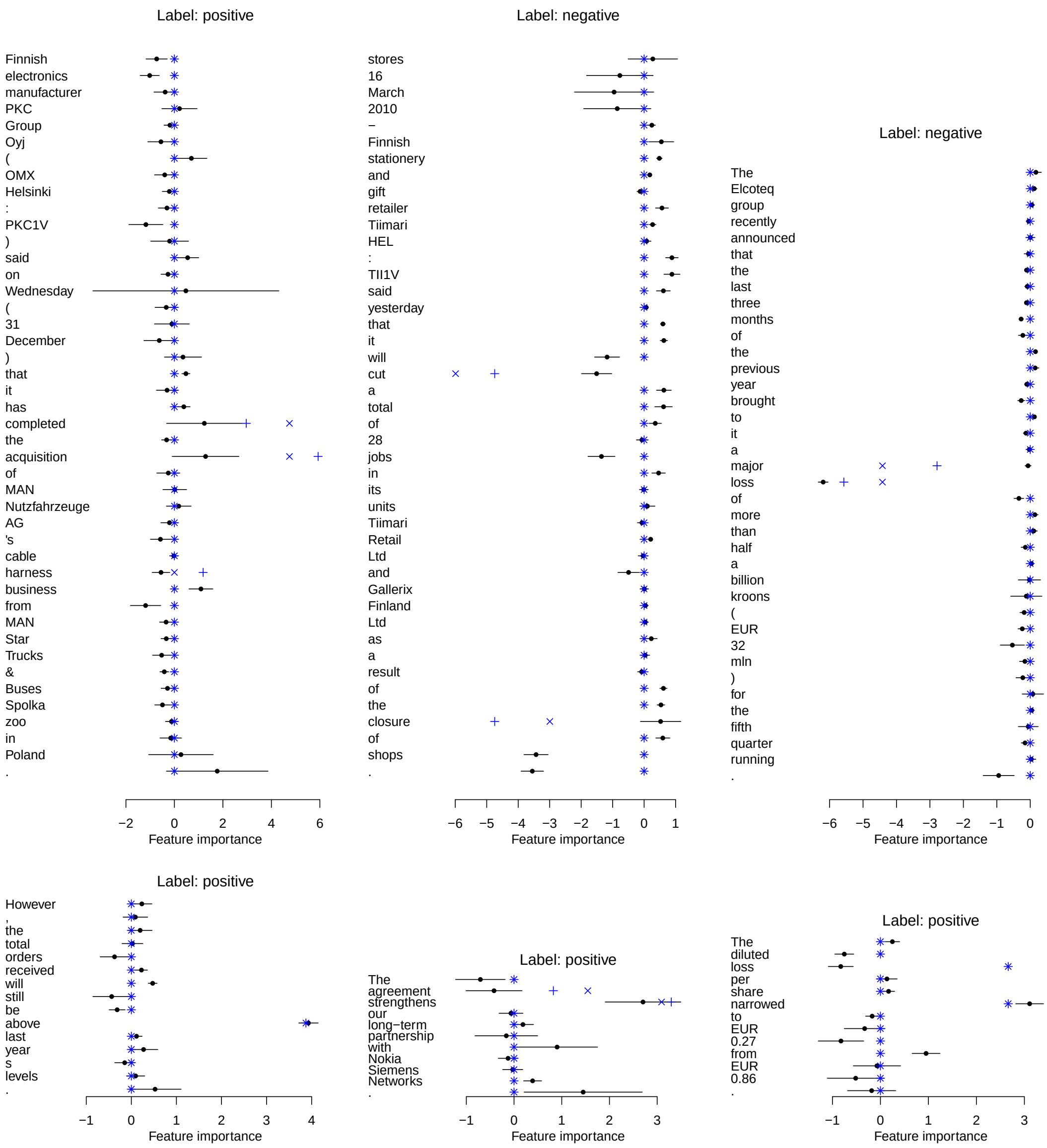

Figure C.10: The word-level feature importances according to the PLR-E model (black) are compared with annotated importances by the authors (blue, + and × represent the two annotators).

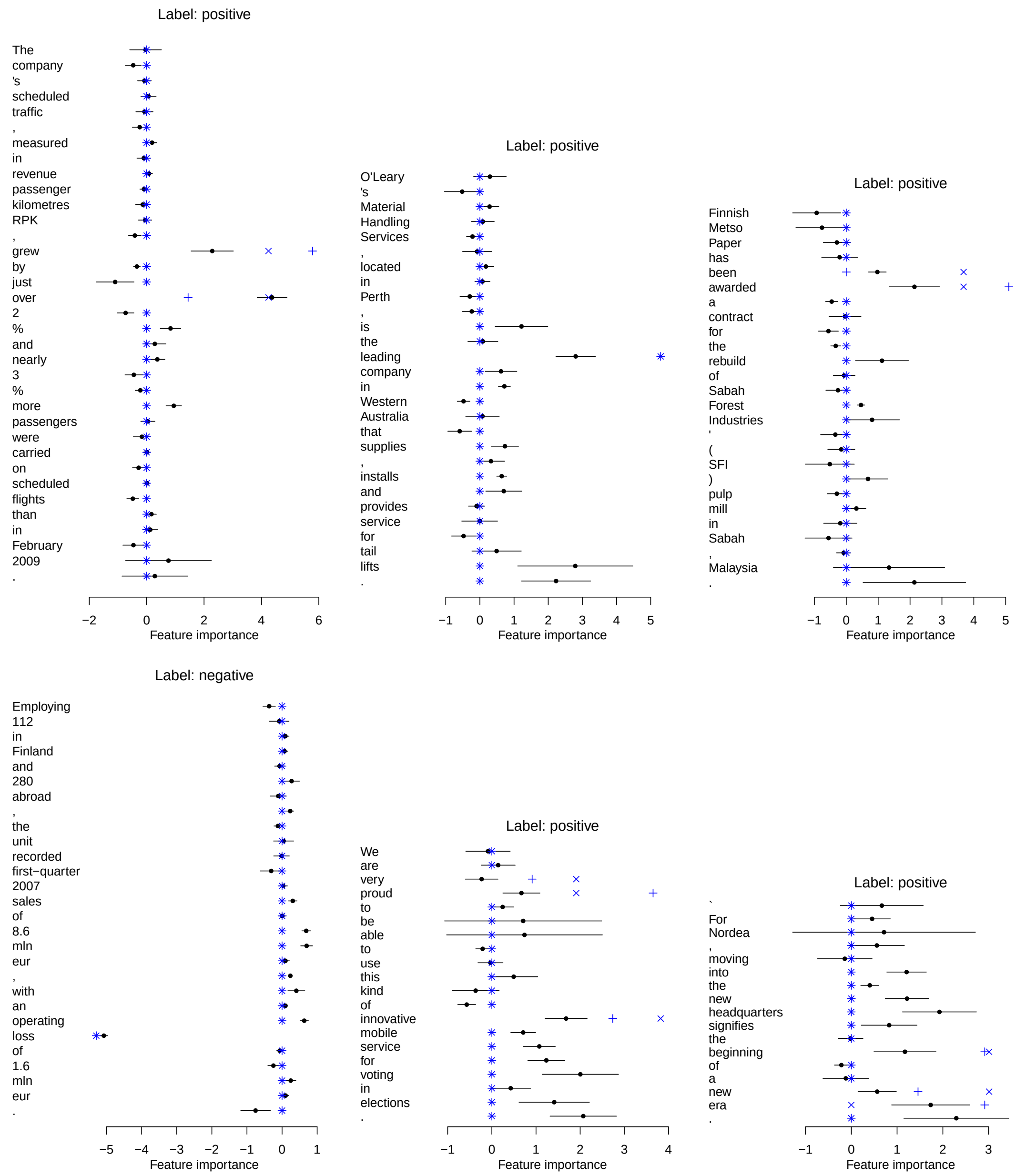

Figure C.11: The word-level feature importances according to the PLR-E model (black) are compared with annotated importances by the authors (blue, + and × represent the two annotators).

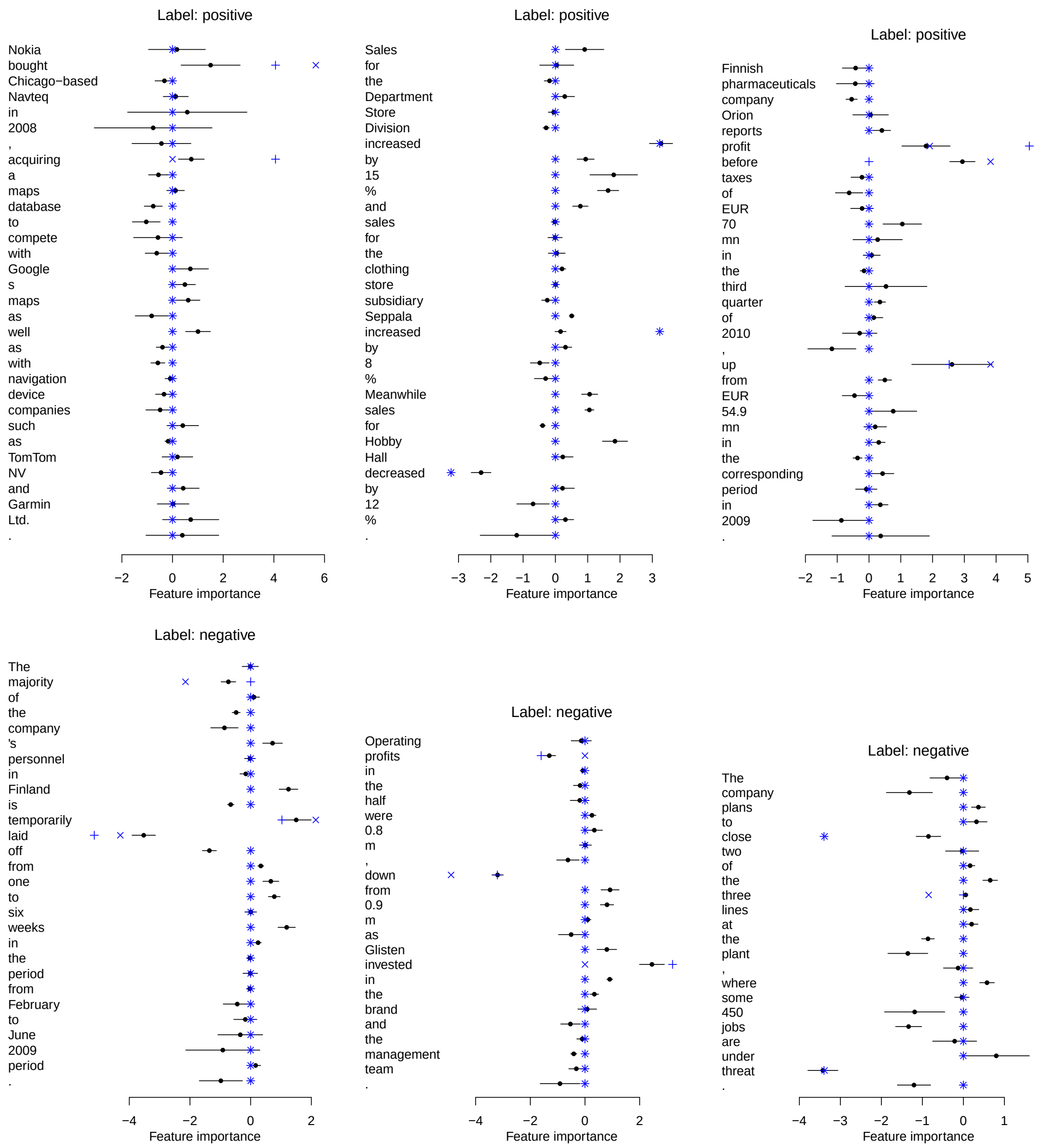

Figure C.12: The word-level feature importances according to the PLR-E model (black) are compared with annotated importances by the authors (blue, + and × represent the two annotators).