

Lookahead Bias in Pretrained Language Models

Suproteem K. Sarkar and Keyon Vafa

June 28, 2024

Comments welcome.

Empirical analysis that uses outputs from pretrained language models can be subject to a form of temporal *lookahead bias*. This bias arises when a language model's pre-training data contains information about the future, which then leaks into analysis that should only use information from the past. In this paper we develop direct tests for lookahead bias, based on the assumption that some events are unpredictable given a prespecified information set. Using these tests, we find evidence of lookahead bias in two applications of language models to social science: Predicting risk factors from corporate earnings calls and predicting election winners from candidate biographies. We additionally discuss the limitations of prompting-based approaches to counteract this bias. The issues we raise can be addressed by using models whose pretraining data is free of survivorship bias and contains only language produced prior to the analysis period of interest.

First version: March 11, 2024. Sarkar: Harvard University (suproteemsarkar@g.harvard.edu); Vafa: Harvard University (kvafa@g.harvard.edu). We thank John Y. Campbell, Jiafeng Chen, Max Miller, Sendhil Mullainathan, Ashesh Rambachan, and Andrei Shleifer for helpful comments. Sarkar gratefully acknowledges support from a National Science Foundation Graduate Research Fellowship under grant number DGE-2140743 and a Two Sigma PhD Fellowship. Vafa gratefully acknowledges support from the Harvard Data Science Initiative.

Pretrained language models are trained on large sets of historical language data (Radford et al., 2018; Devlin et al., 2019), which include newspapers, Wikipedia articles, and snapshots of language across the entire internet (Liu et al., 2019). These data contain information about *statistical properties* of the language—this information allows pretrained models to perform several linguistic tasks (Brown et al., 2020). Language is a key input into many kinds of analysis in social science (Gentzkow et al., 2019), and pretrained language models that encode statistical information about the structure of language can help researchers perform these analyses more effectively.

In addition to containing information about statistical properties of language, the pretraining data of these models also contains information about the *events* encoded in the language. This information may lead analysis that uses pretrained models to exhibit a new form of temporal *lookahead bias* (Lopez-Lira and Tang, 2023; Glasserman and Lin, 2024; Halawi et al., 2024). A researcher, for example, may be interested in analyzing a firm’s future risks given the language of one of its earnings calls. If information about the firm’s future outcomes is in the language of a model’s pretraining corpus, the researcher’s analysis may mislabel the language model’s information about these future outcomes as a genuine example of the model’s predictive ability.

This form of lookahead bias from pretraining can affect several applications of language models in social science. Many questions in social science involve a form of prediction (e.g. Kleinberg et al., 2015): A researcher may be interested in a firm’s risk factors, a politician’s electoral prospects, or a country’s growth opportunities. The additional information in a language model’s pretraining corpus may help a researcher analyze these questions by allowing her to perform new transformations on language data. However, this additional information in the pretraining corpus may also leak the content of historical events, which could lead a researcher to mismeasure the predictive performance of her analysis.

In this paper we develop direct tests for lookahead bias in pretrained language models. Tests for lookahead bias are crucial for assessing whether forecasts made using these language models are valid. Our tests rely on the assumption that some information is unpredictable given an information set of interest. We apply these tests in two settings: Predicting risk factors from corporate earnings calls from before the COVID-19 pandemic and predicting the outcomes of close U.S. House elections from candidate biographies. We find strong evidence of lookahead bias. We additionally show that prompting-based approaches do not eliminate the potential for lookahead bias. Finally, we discuss analysis procedures that would address this bias.

Figure 1: Language model outputs can leak information about future events.

Query	Output
<p>The following is a section of a corporate earnings call for Zoom Video Communications Inc:</p> <p>[earning call speech section]</p> <p>The call took place on 2019-09-05. Consider only information up to and including the earnings call. Predict the potential risks for this company in 2020</p>	<ol style="list-style-type: none">1. Competition: Zoom faces intense competition in the video conferencing market from established players such as Cisco Webex, Microsoft Teams, and Google Meet. As more companies shift to remote work, the competition is likely to intensify, leading to potential market share loss and pricing pressure for Zoom.2. Security and Privacy Concerns: With the increasing use of video conferencing, security and privacy concerns have become more prominent. Zoom faced several security issues in 2019, including a vulnerability that allowed hackers to access cameras and microphones of Mac users. The company must continue to address these concerns and ensure the security and privacy of its users' data to maintain trust and avoid reputational damage.3. Global Economic Uncertainty: The COVID-19 pandemic has created economic uncertainty, which could impact Zoom's growth in 2020. A recession or economic downturn could lead to reduced spending on technology, potentially affecting Zoom's revenue and profitability.4. Dependence on a Few Large Customers: [...]

Notes – This figure reports the output from a pretrained language model (Llama 2-70B) that is queried with an earnings call for Zoom Video Communications, Inc from 2019 and instructed to predict the firm’s risk factors for 2020. The output contains a clear reference to a language sequence that did not exist during the analysis period: “COVID-19 pandemic.” In addition, the output mentions “remote work,” which, while not inherently unpredictable, was a risk factor that became more prominent after the beginning of the COVID-19 pandemic. We provide additional details on the generation procedure in Appendix A.1.

Figure 1 provides an example of how information about future events can leak into outputs from a pretrained language model. We query a language model with an earnings call for Zoom Video Communications, Inc from September 5, 2019, and instructions to generate potential risk factors for the firm over the following year. Some of these risks relate to information from before the earnings call, including a Zoom webcam vulnerability discovered in the summer of 2019. However, the language model output also contains a clear example of information that occurred after the earnings call: The output includes “COVID-19 pandemic”—a language sequence that did not exist at the time of the call. In addition, the output discusses competition from other videoconference software as “more companies shift to remote work.” While many companies shifted to remote work in 2019, a larger shift occurred in 2020 after the start of the COVID-19 pandemic. The “COVID-19 pandemic” output reveals that language models can produce direct mentions to concepts that did not exist during the desired analysis window. The “remote work” output suggests that additional information about

historical events—even if not explicitly mentioned in a language model output—may influence a language model’s generation process.

In Section 1 we define information leakage and lookahead bias in the context of using pretrained language models to make predictions. Information leakage is a bias that occurs when an algorithm generates language using information that is not contained in the specified information set. We assume, for example, that an analysis procedure that systematically generates the language sequence “*COVID-19*” displays information leakage with respect to an information set that ends in November, 2019. Lookahead bias is an important case of information leakage. It occurs when an analysis procedure makes predictions that cannot be made with respect to the input language and the specified information set. We assume, for example, that an analysis procedure that uses language models to systematically and successfully predict the winner of close elections displays lookahead bias with respect to information sets that end before these elections. These biases can arise when the pretraining corpus of a language model contains language that occurs after the analysis period. In addition, these biases can arise when the pretraining data is *selected* based on information from after the analysis period—this selection bias can occur even if the language of the pretraining data lies fully before the analysis period. Both mechanisms for bias are present even if the text a researcher uses to query the model does not leak information about the future.

In Section 2, we develop tests for information leakage and lookahead bias, and find evidence of these biases. We first show that the outputs of language models systematically leak information about future events. We prompt a language model with the text of corporate earnings calls from September–November 2019 and instructions to predict the risks for the firm in the call in 2020. We find that the language sequence “*COVID-19*”—which did not exist at the time of each earnings call—appears in 6.8% of generations. We also find more subtle evidence of information leakage about future events. References to “*pandemic*” or “*disease outbreak*” are 3.6 times more common in outputs that predict 2020 risks than they are in outputs that predict 2019 risks. In addition, references to “*pandemic*,” “*disease outbreak*,” or “*supply chain*” are 35% more common in outputs that predict 2020 risks than they are in outputs that predict 2019 risks.

We next show that predictions based on language model outputs can be affected by lookahead bias. We argue that a strong test for whether an analysis that uses a language model exhibits lookahead bias is whether it can predict an unpredictable event. We evaluate whether a language model can predict an outcome of a close election—a

“natural experiment” that is commonly assumed to be unpredictable prior to the date of the election (Eggers et al., 2015). We show that even for very close elections, with victory margins that range from 0.5% to 2%, a language model can predict the election winner with 70–80% accuracy. These results provide evidence of lookahead bias in analysis that uses pretrained language models.

Our tests are one-sided. They may not detect information leakage in all circumstances. But when our tests detect leakage, leakage is guaranteed with respect to the assumptions about the information set. In this sense, the tests are conservative: They are designed to find the most extreme symptoms of information leakage. If our tests lead a language model to generate unpredictable language sequences, the model has demonstrated the potential for lookahead bias. Many domains in economics—including asset pricing and macroeconomic forecasting—make predictions in environments with low inherent predictability. Any kind of lookahead bias could lead analyses in these domains to mismeasure predictive performance.

In Section 3, we discuss prompting-based strategies to address these biases that are based on changing the text used to prompt a language model. We identify limitations of these approaches. We show that including an instruction to not use future information in a language model prompt does not eliminate information leakage. We additionally show that censoring identifying information from a prompt—like the date or company of an earnings call—does not guarantee this information cannot be inferred from the censored prompt. We find that language models can predict 70% of firm names from earnings call segments in which the firm name has been censored. We also find that the year a language model infers from the text of an earnings call segment with censored dates has a correlation of 0.79 with the actual year of the call. These results are consistent with the findings in other domains that machine learning models can infer identifying information from data, even if humans cannot infer it.¹ Our results demonstrate that prompting-based strategies do not eliminate the potential for lookahead bias.

In Section 4, we discuss analysis procedures that are not subject to information leakage from pretraining. If a language model’s pretraining corpus contains language from before the analysis window—and the corpus’ language is not selected based on information from after the analysis window—an analysis procedure that uses this language model does not have the potential for the kind of lookahead bias we discuss.

¹For example, Gichoya et al. (2022) find that machine learning models can infer race from medical images – even from images that the authors have “degraded” and from which humans cannot infer race.

At minimum, we argue that researchers should separately report the performance of analysis that occurs within a language model’s pretraining window and after a language model’s pretraining window. We also identify a class of language models—language models with time subscripts (for example, the StoriesLM family, Sarkar, 2024)—that can be used to conduct analysis without the potential for lookahead bias. While these kinds of models are not yet trained at the scale of the largest currently-available pretrained models, they do allow researchers to perform transformations on language data without the potential for lookahead bias from pretraining.

Related literature Our paper contributes to the literature on lookahead bias in social science. Elton et al. (1996) show how mutual fund survivorship bias—which arises when data availability for past periods is conditional on future performance—can bias measures of fund performance. Kapoor and Narayanan (2022) identify cases of “temporal leakage” in social science. In contrast to these papers, which discuss and identify examples of information leakage in the data used by a researcher, our paper develops tests for and finds evidence of a form of lookahead bias that arises from the pretraining procedure of a language model that a researcher applies to data. Leakage is a well-studied problem that can take many forms—in this paper, we study temporal leakage from pretraining, which can arise when language models are used to make predictions about future events.

Our paper also contributes to the literature on leakage in language models. One part of this literature studies a form of leakage known as *data contamination*, which occurs when a model is (pre)trained on data that is also in the test set used to evaluate the model (Dodge et al., 2021; Golchin and Surdeanu, 2023; Sainz et al., 2023). For example, Oren et al. (2023) present a procedure for identifying whether a language model memorizes language from the test set. Another strand of this literature studies a form of leakage that occurs when language models generate sensitive information from the training corpus (Inan et al., 2021; Carlini et al., 2019, 2022). In contrast to these papers, which study memorization of language sequences in the pretraining corpus, our paper studies information leakage from the pretraining corpus. This form of information leakage can occur even when a model does not leak exact language sequences from the corpus.

One recently proposed strategy to assess lookahead bias is to separately report analysis results from an “out-of-sample” portion of data that lies after a language model’s pretraining cutoff (Jha et al., 2023; Bybee, 2023; Glasserman and Lin, 2024). All

predictions on the “out-of-sample” portion will be free from lookahead bias. However, if models perform similarly across samples, it does not mean there is no lookahead bias on the original sample; the inherent predictability may vary across samples (see Section 4 for more discussion). Similarly, if models perform differently across samples, it does not guarantee lookahead bias. In contrast, our tests assess whether a language model generates unpredictable information. If our tests detect lookahead bias, the bias is guaranteed.

In recent interesting work, Glasserman and Lin (2024) study the effects of lookahead bias when using language models to predict stock returns from news headlines. They propose to mitigate lookahead bias by removing company names from prompts, with the goal of anonymizing the identity of a company and preventing a language model from using future information about the company. The censorship strategy is valid as long as a language model cannot infer information related to the company (e.g. company name or industry) after identifying information is removed from a prompt. We find that identifier censorship is not always a valid strategy. For example, we show that after removing a firm’s name from an earnings call, a language model can still predict which firm the call refers to with 70% accuracy. This evidence is consistent with evidence in other domains that shows that machine learning models can infer information from “degraded” data, even if humans cannot (e.g. Gichoya et al., 2022).

Finally, our paper contributes to the broader literature on pretrained language models in economics and finance (e.g. Chen and Sarkar, 2020; Jha et al., 2020; Wu et al., 2023; Rajan et al., 2023; Hansen et al., 2023). We find direct evidence of a new form of lookahead bias that may affect analysis that uses these pretrained language models. We identify analysis procedures that are not subject to the bias that can be used to advance research in this area.

1. Organizing Framework

We define information leakage and lookahead bias in the context of pretrained language models. Both these biases are defined with respect to an information set of interest. Information leakage occurs when an algorithm produces language while using information outside of the information set. Lookahead bias—an important example of information leakage—occurs when an analysis procedure makes predictions that cannot be made using the information set.

1.1. Research Design

A researcher aims to predict an outcome Y_{t+1} from language data X_t . She requires that her prediction only uses information available in a prespecified information set \mathcal{I} . The researcher's object of interest is the following conditional expectation:

$$\mu(X_t; \mathcal{I}) = \mathbb{E}[Y_{t+1} | X_t; \mathcal{I}]. \quad (1)$$

The researcher builds a model $\widehat{\mu}(X_t; \mathcal{I})$ to approximate $\mu(X_t; \mathcal{I})$. Throughout this paper, we assume that the researcher's information set \mathcal{I} contains no information made available after time t . We thus refer to $\widehat{\mu}(X_t; \mathcal{I})$ as a *predictive* model; it uses the information available at time t to make a prediction about an outcome at time $t + 1$.

Many economic analyses involve building predictive models from language data. We highlight examples from three domains below.

Finance. A researcher aims to predict a firm's stock return Y_{t+1} over year $t + 1$ given the language of its earnings call X_t held in year t . As this exercise is predictive, the researcher requires that the analysis only considers information \mathcal{I} about the economic environment made available in year t . The risk factors that influence this stock return—which could be generated using language model outputs—should only correspond to factors that are predictable from this past information.

Political economy. A researcher aims to predict whether a politician will win an election Y_{t+1} given the language of the politician's biography X_t . As this exercise is predictive, the researcher only wants the analysis to consider information \mathcal{I} about the political environment made available prior to the election. Information about how candidate characteristics may influence electability should be from prior to the election at $t + 1$.

Macroeconomics. A researcher aims to predict a country's employment level Y_{t+1} in year $t + 1$ given the language of its monetary policymakers X_t in year t . As this exercise is predictive, she only wants the analysis to consider information \mathcal{I} about the economy made available in year t . Information about how policies may influence employment should be from prior to the realized employment period $t + 1$.

1.2. Leakage and Lookahead Bias

The researcher’s goal is to build a predictive model from language data. We focus on a setting in which the researcher uses a pretrained language model as part of this procedure.

A language model is a function $f(X; \mathcal{M})$ that takes as input text X to generate new text. Language models are pretrained on historical language data that corresponds to an information set \mathcal{M} . For example, the pretraining data can include Wikipedia articles and snapshots of language across the internet (Liu et al., 2019). We include the information set \mathcal{M} in our notation to make a language model’s dependence on this information set explicit.

A researcher can use many strategies to incorporate a language model into her predictive model. One option is to use an analysis procedure $\widehat{\mu}(X_t; \mathcal{I}) = f(X_t; \mathcal{M})$ that directly uses a language model to forecast Y_{t+1} from X_t . For example, a researcher could query a language model to predict whether a candidate will win an election given the candidate’s biography. The researcher could also indirectly incorporate language model outputs using a transformation, $\widehat{\mu}(X_t; \mathcal{I}) = g(f(X_t; \mathcal{M}); \theta)$, parameterized by θ . For example, the researcher could use a language model to generate risk factors from a company’s earnings call, and then apply a risk pricing model to these generated risk factors.

A researcher who uses a language model to make predictions faces a potential problem: The language model may use information that is not contained in the desired information set \mathcal{I} . We refer to this problem as **leakage**. Specifically, leakage occurs when $\mathcal{M} \not\subseteq \mathcal{I}$. Since we consider settings in which the researcher’s information set \mathcal{I} contains information up to time t , leakage occurs when the language model is pretrained using information from after time t .

To demonstrate how leakage can influence empirical analysis, note that the researcher’s data follows the structural relationship,

$$\begin{aligned} Y_{t+1} &= \mu(X_t; \mathcal{I}) + \varepsilon_{t+1} \\ \varepsilon_{t+1} &\perp\!\!\!\perp X_t, \mathcal{I} \end{aligned}$$

since $\mu(X_t; \mathcal{I})$ is a conditional expectation. In this sense ε_{t+1} is the **irreducible error** in this prediction task, corresponding to the component of Y_{t+1} that is unpredictable from X_t and \mathcal{I} . No function of a text sequence X_t and the information set \mathcal{I} can be correlated with ε_{t+1} . If a model’s predictions are correlated with the irreducible error,

the model is using information not in the information set. We refer to this as **lookahead bias**. Specifically, we define lookahead bias as

$$\text{Cov}(\widehat{\mu}(X_t; \mathcal{I}), \varepsilon_{t+1}) \neq 0 \quad (2)$$

Lookahead bias is a kind of information leakage that leads a language model-based prediction procedure to use information not in the prespecified information set \mathcal{I} .

1.3. Opportunities for Information Leakage

When does an analysis procedure that uses a language model have the potential for information leakage—including lookahead bias? Consider an analysis procedure that uses a language model whose pretraining data $D_{i \rightarrow j}$ contains language from time i to time j . Consider an information set \mathcal{I}_t of all information up to time t . Information leakage can arise if the language of the pretraining data includes information not in \mathcal{I}_t , or if the selection procedure of the pretraining data includes information not in \mathcal{I}_t .

Information leakage from the language of the pretraining corpus. Information leakage can occur when the pretraining data includes language from after the analysis window, so that $j > t$. For example, a pretraining corpus may contain information about the COVID-19 pandemic from 2020, while a researcher’s desired information set \mathcal{I}_t is information about economic events up to Fall 2019. Outputs from this language model may use information about the pandemic that is not present in the researcher’s specified information set.

Information leakage from the selection of the pretraining corpus. Information leakage can also occur when the pretraining corpus is selected based on information from after the analysis period. For example, consider a language model whose pretraining corpus consists of financial news articles about firms that have survived up to time $k > t$. Even if the training cutoff date falls before the analysis period ($j < t$), the selection of pretraining language based on firm survivorship may lead to lookahead bias in analysis that uses the language model.

For example, suppose that small firms and large firms are equally likely to have poor profits in a given quarter. However, when a small firm has poor profits it is more likely to shut down than a large firm that has poor profits. This would lead the pretraining corpus, which selects firms that have survived up to time $k > t$, to oversample higher-

performing small firms. A language model pretrained on this corpus may thus generate high estimates of the performance of small firms. This would lead to a bias in the predicted performance for these firms—this bias would not occur if the pretraining corpus was selected based on the researcher’s desired information set \mathcal{I}_t .

2. Evidence of Information Leakage and Lookahead Bias

We develop tests to identify information leakage and lookahead bias from in pretrained language models. Our tests are based on the assumption that, given an information set of interest, some language sequences cannot be generated and some events cannot be predicted. These tests are one-sided. While they may not identify all instances of leakage, they will only detect leakage when it is present. We design our tests to be conservative in this way so we can assess the potential for even the most extreme symptoms of lookahead bias. We present results that find evidence of leakage and lookahead bias in two settings: Predicting risk factors from corporate earnings calls and predicting election winners from candidate biographies.

Section 2.1 shows that the content of language model outputs leaks information about the future. We find that language sequences that did not exist at the time of the analysis period appear in a language model’s generations. In addition, the frequencies of generated sequences relate to information we argue was not the information set at the time of the analysis period. Section 2.2 shows that inference based on language model generations can be subject to lookahead bias. We find that a standard class of “natural experiments” that are assumed to be unpredictable based on past information can be predicted using a language model’s outputs.

2.1. Language model outputs can leak information about the future

How can we be certain that a language model’s generations correspond to information leakage, and not just good predictive ability? A language model prompted to generate risk factors for a firm may successfully generate the firm’s future outcomes either because information about the firm’s future leaks into the generation, or because the model’s pretraining procedure improves its ability to forecast risks. To make progress on this question, we assume that some kinds of language cannot be generated under a given information set. We say that a language model exhibits information leakage if it systematically generates language sequences that did not exist during the analysis period.

Testing for information leakage. To test for information leakage, we consider the problem of predicting a firm’s risk factors given the language of its corporate earnings calls. We assume that given the language of an earnings call from November 2019 and information up to November 2019, a language model displays information leakage if it systematically generates language outputs related to the COVID-19 pandemic.

How can information about the pandemic appear in language model outputs? One mechanism is through *direct leakage*: This occurs when the phrase “COVID-19”, which was not in the information set in November 2019, systematically appears in the generated risk factors. Another, more subtle mechanism is through *indirect leakage*: This occurs when the output contains risk factors, like supply chain shortages, that were made more likely by the pandemic.

To test for information leakage, we query a language model with corporate earnings calls from September–November 2019 and instruct the model to generate each firm’s potential risk factors. We obtain earnings call data from the StreetEvents database. We filter to earnings calls that occurred between September 1, 2019 and November, 30 2019. We isolate the initial speech section, which does not include analyst Q&A, from each earnings call. We consider the first 2,000 characters of each earnings call speech. To reduce computation costs, we randomly sample 1,000 earnings call speeches from this period. For each call, we query the model with the following prompt, substituting the bracketed terms with specific information for each earnings call:

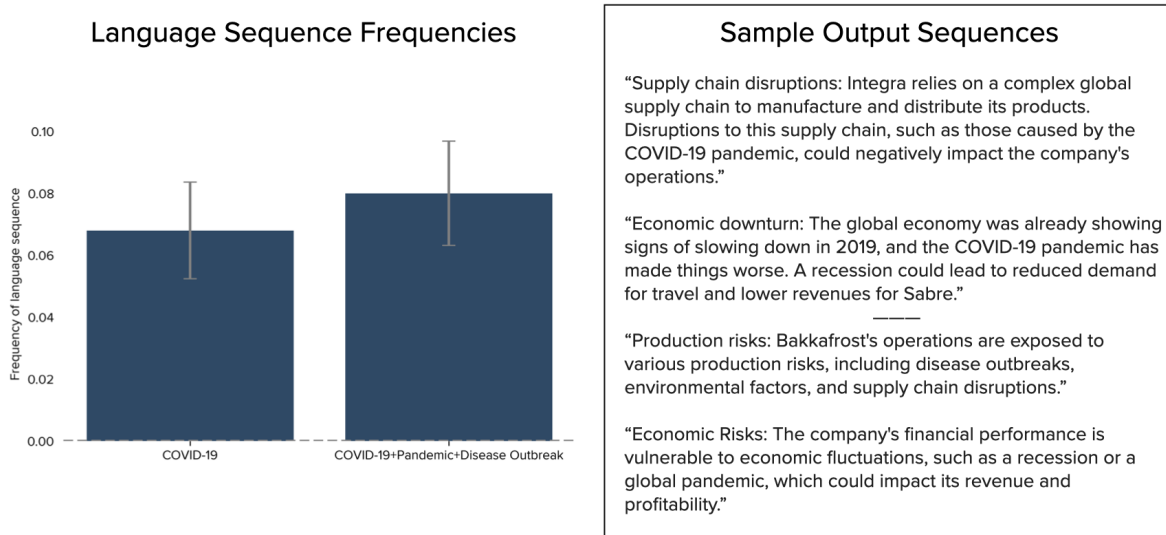
The following is a section of a corporate earnings call for [firm]:
[earnings call section]
The call took place on [date].
Consider only information up to and including the earnings call.
Predict the potential risks for this company in 2020

We generate outputs using the Llama-2 70B language model introduced by Meta in Touvron et al. (2023). We use this model because it is publicly available and the cutoff date for its training corpus (July 2023) is made public.² In addition, the model’s weights are frozen at a point in time, which allows for experiment reproducibility—something we would not have if we used API-based models that update frequently. To allow for natural language question answering, we use the version of the model that has undergone instruction tuning and reinforcement learning from human feedback.³ We assess the degree of direct and indirect leakage in these language model output.

²The base model’s pretraining data cutoff is September 2022, and the tuning data cutoff is July 2023.

³We include additional information on the generation procedures for our results in Appendix A.2.

Figure 2: Language model outputs directly leak information about future events.



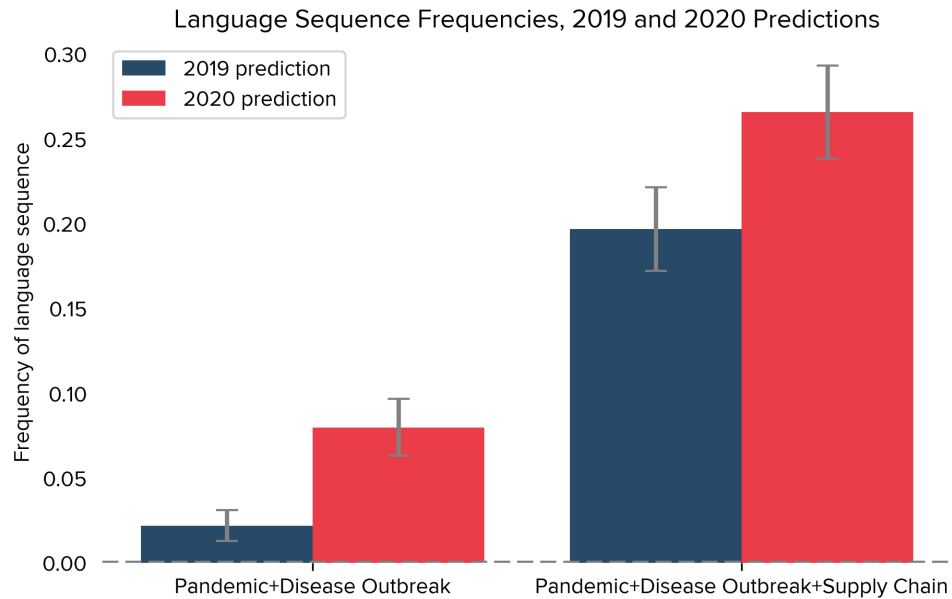
Notes – This figure summarizes outputs from language models prompted with corporate earnings calls from 2019 and instructed to predict risks for the firm in each call. The left panel reports that 6.8% of generations include the language sequence “COVID-19” and 8.0% of generations include the language sequences “COVID-19,” “Pandemic,” or “Disease Outbreak.” The right panel includes four excerpts selected from these language model outputs. Error bars report 95% confidence intervals..

Language model outputs directly leak information about future events We perform a case-insensitive string match to count how often language model outputs mention a given set of search sequences. We find that 6.8% of the outputs include the language sequence “COVID-19.” If we broaden the language filter to include “pandemic” and “disease outbreak,” we find that 8.0% of outputs include these language sequences.

Figure 2 shows the frequencies of the language sequences {“COVID-19”} and {“COVID-19”, “pandemic”, “disease outbreak”} in the language model outputs. The first finding—that 6.8% of generations include the language sequence “COVID-19”—is clear evidence of direct information leakage. The sequence “COVID-19” is a sequence we assume was not in the information set during the analysis period, but systematically appears in the language model output.

Language model outputs indirectly leak information about future events. A language model can leak information about the pandemic indirectly even if it does not directly mention “COVID-19”. For example, if topics associated with COVID-19 are

Figure 3: Language model outputs indirectly leak information about future events.



Notes – This figure reports the frequency of language sequences in outputs from language models prompted with corporate earnings calls and instructed to predict future risks. We color in blue the results that use outputs from models prompted to predict 2019 risks using 2018 earnings calls. We color in red results that use outputs from models prompted to predict 2020 risks using 2019 earnings calls. Predicted 2020 risks are 3.6 times more likely to mention “pandemic” or “disease outbreak,” and 35% more likely to mention “pandemic,” “disease outbreak,” or “supply chain.” Error bars report 95% confidence intervals.

mentioned more frequently in risks a language model generates for 2020 than they are in risks a model generates for 2019, the model may have indirectly leaked information about the pandemic. To test this mechanism, we re-run the analysis described above using earnings calls from 2018. We use a random sample of 1,000 earnings calls that took place between September 1, 2018 and November 30, 2018, and query the model to predict risks for each firm in 2019. The modified prompt is:

The following is a section of a corporate earnings call for [firm]:
 [earnings call section from 2018]
 The call took place on [date].
 Consider only information up to and including the earnings call.
 Predict the potential risks for this company in 2019

Figure 3 reports the frequencies of pandemic-related phrases across language model outputs that predict risks in 2019 and risks in 2020. We first consider the set of phrases

{“*pandemic*”, “*disease outbreak*”}. We find this set in 2.2% of generated risks for 2019 and 8.0% of generated risks for 2020. We next consider the same set of phrases augmented with “*supply chain*.” The reasoning behind this search criteria is to measure the potential for an even broader kind of indirect leakage: While “*pandemic*” and “*disease outbreak*” are directly semantically related to the pandemic, “*supply chain*” risks also became much more prevalent in 2020 after the start of COVID-19. We find language from this set in 19.7% of risks for 2019 and 26.6% of risks for 2020. While this evidence is not as direct a test for information leakage as the presence of the new language sequence “*COVID-19*,” it does suggest that information leakage can apply to the overall distribution of language model outputs. A language model need not output the sequence “*COVID-19*” for it to be leaking information—leakage may be more subtle, increasing the frequencies of phrases like “*supply chain*” that are associated with the pandemic.

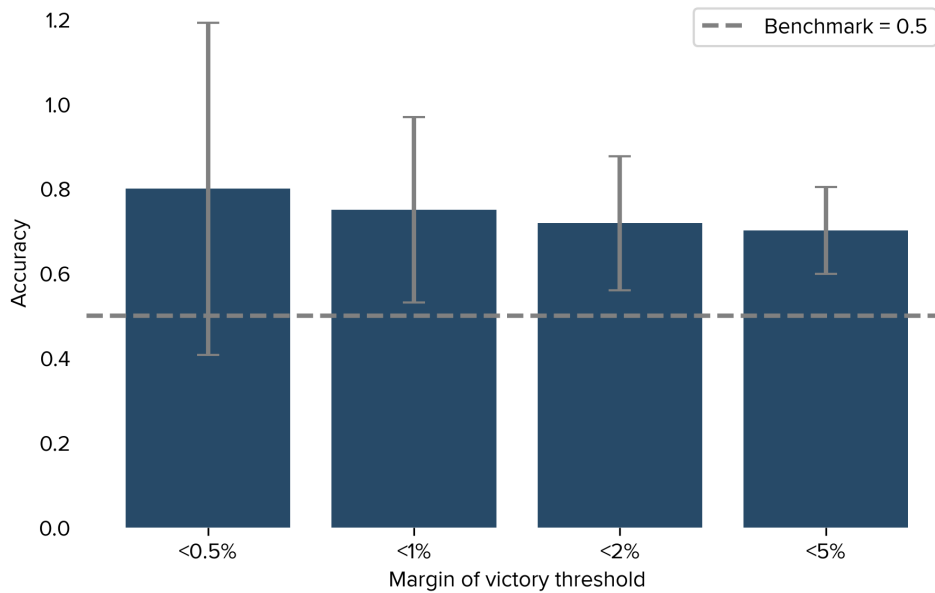
2.2. Predictions from language model outputs can exhibit lookahead bias

How can we find evidence of lookahead bias in predictions that use language model outputs? Recall from Equation (2) that lookahead bias occurs when a prediction from a pretrained language model-based analysis procedure correlates with the irreducible error with respect to the analyst’s specified information set. One way to test for lookahead bias is to identify a domain in which all the variation in the outcome is from the irreducible error: A classic example of such a domain is a *natural experiment*. We show that language model outputs can be used to predict the outcomes of close elections, which we assume are examples of natural experiments (Eggers et al., 2015).

Testing for lookahead bias. We download the results of the past five U.S. House elections—from 2014-2022—from the MIT election lab. For each contested election, we identify the top two candidates by vote share. For each of these candidates, we download their biography from Ballotpedia. For 732 contested races, we find matching biographies on Ballotpedia for both candidates in the race. For each race, we use the following prompt:

Use information only from before election day [year]
The two candidates in the [year] U.S. House election are [candidate 1] and [candidate 2]
The bio for [candidate 1] is [candidate 1 bio]
The bio for [candidate 2] is [candidate 2 bio]

Figure 4: Predictions that use language model outputs can determine the results of “natural experiments.”



Notes – This figure reports the accuracy of an analysis procedure that uses language model outputs to predict U.S. House election winners. Each bar reflects the accuracy of the procedures for races within the margin of victory reported on the horizontal axis. Predictive accuracy ranges from 70%–80%, even for very close elections. Error bars report 95% confidence intervals.

Out of [candidate 1] and [candidate 2], the candidate more likely to win the [year] election is

We generate outputs using the Llama-2 70B language model (Touvron et al., 2023). We use the base model—not the instruction-tuned version—so that our generated output corresponds only to the name of a candidate.

For each race, we compute the margin of victory with respect to the two-candidate vote share of the winning candidate. We define the two-candidate vote share as the total votes for the winning candidate divided by the total votes for both the winning and second-place candidate. We define the two-candidate margin of victory as the winner’s two-candidate vote share minus the second place finisher’s two-candidate vote share.

Language model outputs can determine the results of “natural experiments.” Figure 4 evaluates the accuracy of the language-model based prediction for elections within varying two-candidate margins of victory. Even for very close elections, the model’s accuracy ranges from 70% to 80%. This result demonstrates that the results of very

close elections, which are typically assumed to be unpredictable, can be predicted using language model outputs.

3. Limitations of Prompting-Based Approaches

Our previous results found evidence of information leakage in language model outputs and lookahead bias in predictions that use language model outputs. How might a researcher address these biases? We first discuss the limitations of two prompting-based approaches that do not directly relate to the pretraining procedure of the language model used for analysis.

One prompting-based approach is to prompt a language model to not use information from beyond the analysis period. We note that our analyses in Section 2 includes these kinds of prompts, and still find evidence of lookahead bias. We additionally show that prompting does not even remove the most obvious symptom of direct information leakage. As of now, there is no statistical framework that suggests that prompt design will eliminate lookahead bias.

Another approach is to censor, or mask, information about dates or language source identities in the text used to query a model. The motivation behind this approach is to remove identifiers from a language sequence, as these identifiers could lead a language model to produce output that correlates with future information about those identifiers. We show that such a strategy does not guarantee the censorship of identifying information in our setting—language models can infer identifiers from language sequences even if these identifiers are removed. We find that stronger forms of identifier censorship, which have the side effect of removing even more information about the text input that may be useful for prediction, still do not guarantee that a language model from cannot infer the identifying information.

3.1. Prompt Design

Recent work has argued that prompt design may lead language models to “unlearn” information (Pawelczyk et al., 2023) from training data. As information about time appears to be encoded in the weights of language models (Nylund et al., 2023), one might believe that prompting could help to remove temporal information from language model outputs. However, the kinds of information leakage that we discuss in this paper might not be addressed by simple interpolations in parameter space. It is also not clear how these interpolations could be conducted using only natural-language prompts.

For example, our main results in Section 2.1 and Section 2.2 include prompts with instructions to not use information from after the analysis period. For both of these sets of results, prompting alone is not enough to solve the lookahead bias problem. To assess the effects of prompting in this research design, we replicate the results from Figure 2 using a new prompt

The following is a section of a corporate earnings call for [firm]:
[earnings call section]
The call took place on [date].
Predict the potential risks for this company in 2020

The prompt in this new analysis removes the line “Consider only information up to and including the earnings call” that we included in our main analysis.

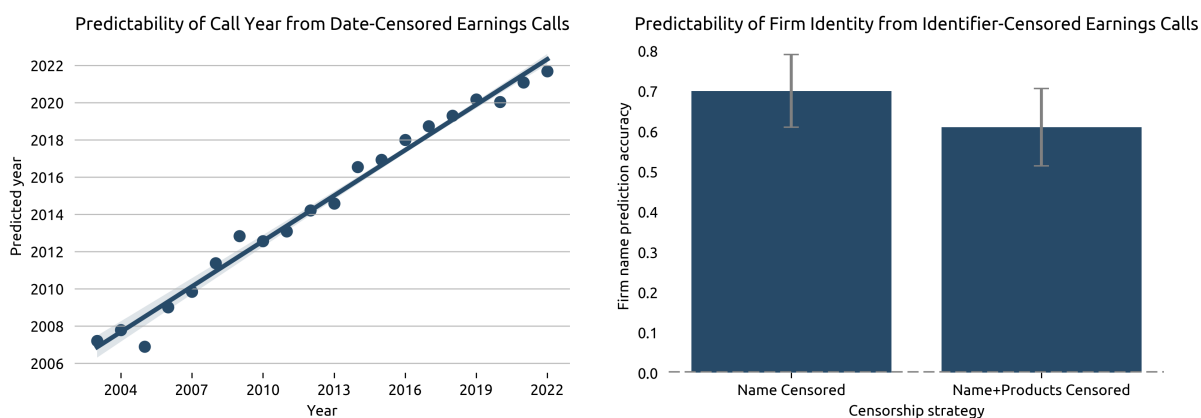
Using this new prompt, which does not include the no-lookahead instruction, we find that “COVID-19” is mentioned in 12.2% of generations. In our main results, where we include an instruction to consider only information available during the analysis period, we find that “COVID-19” is mentioned in 6.8% of generations. While the instruction to not look ahead in our main results decreases the frequency of direct leakage in our outputs, it does not eliminate even this most obvious symptom of information leakage. We also show in Section 2.1 that indirect leakage—differences in the frequencies of generated topics—is affected by lookahead bias even with a prompt to not use future information. As of now, there is no statistical framework that suggests prompting will eliminate lookahead bias in analysis on outputs from pretrained language models.

3.2. Information Masking

Another approach to addressing information leakage is to mask identifying information from the text used to query the language model (Glasserman and Lin, 2024). The reasoning behind this approach is that if a prompt does not contain identifying information, the model’s generation will not correlate with future events related to the identifier. This approach is effective as long as removing identifiers from text sequences makes information related to the identifier unpredictable.

Does censoring identifying information guarantee a model is unable to infer the identifier? We find this is not always the case. We perform two exercises that remove identifying information from prompts and find a language model can still infer this information. First, we randomly sample 50 corporate earnings calls from each year across the 20-year period 2003–2022. We censor all years and all month names in

Figure 5: Masking identifiers from language model prompts does not guarantee the prompts are de-identified.



Notes – This figure reports results from using language models to infer identifying information from earnings calls segments that censor identifying information. The left panel is a binned scatter plot of a language model’s prediction of the year a call took place versus the actual year of the call. The right panel plots the accuracy of a language model’s prediction of the firm’s identity in an earnings call across two strategies that remove identifying information from the call.

each of these 1,000 earnings calls. We then predict, using the GPT-4 API, the year that corresponds to each censored earnings call. The first panel of Figure 5 presents a binned scatter plot of the predicted year versus the true year and finds a strong positive relationship—the correlation between predicted year and true year is 0.79. Second, we randomly sample 100 corporate earnings calls from the September–November 2019 dataset used in Section 2.1. We censor references to the firm’s name, and in another test additionally censor references to the firm’s products. We then predict, using the GPT-4 API, the name of the firm that corresponds to each censored earnings call. The second panel of Figure 5 shows the accuracy of this prediction—the firm name can be reconstructed with 70% accuracy from a call segment with the name censored, and with 61% accuracy from a call segment with the name and products censored.⁴ We additionally show that de-identification becomes less effective as the amount of information used to query a language model increases—Figure A1 shows that identification accuracy increases as the number of characters from each call used to query the language model increases.

These results demonstrate that masking does not guarantee the information in text used to prompt a language model is de-identified. In addition, removing such

⁴We include additional information about the masking and generation procedures in Appendix A.2.

information can also remove context that would be important for a prediction task. “Slow and steady” in the earnings call of a manufacturing firm might forecast something different from “slow and steady” in the earnings call of a technology firm. Masking does not guarantee de-identification of information from language, and could remove important context from a prediction task.

4. Addressing the Bias

How can a researcher address the lookahead bias problem from pretrained language models? We argue the solution lies in using models whose pretraining data is (1) free of survivorship bias and (2) contains only language produced prior to the analysis period of interest. Section 1 shows that lookahead bias can arise when an analysis procedure uses information that is not available in the desired analysis period. If the pretraining data of a language model satisfies these two properties, an analysis procedure that uses its parameters will not be affected by information leakage from after the analysis period.

This kind of empirical strategy is already being used in some work that applies language models to social science. For example, Li et al. (2023), Lopez-Lira and Tang (2023), and Halawi et al. (2024) restrict their analysis periods to after the pretraining cutoffs of the language models they use. However, while the pretraining cutoffs for many commonly-used pretrained language models fall in the past 1–10 years, researchers may be interested analyzing prediction problems over a longer time period. How can research that analyzes this historical data proceed?

An initial test: Reporting results from the out-of-sample analysis window. At minimum, we argue that if researchers use this historical data, they should separately report results from analysis that is *in-sample*—before the language model’s pretraining cutoff, and analysis that is *out-of-sample*—after the language model’s pretraining cutoff. Jha et al. (2023) and Bybee (2023), for example, evaluate the stability of their estimates out of sample. If the conditional distribution of the outcome given the text input does not vary across these analysis windows, a difference between the out-of-sample and in-sample results may suggest lookahead bias. For example, Glasserman and Lin (2024) show that a language model-based trading strategy no longer performs at traditional levels of statistical significance when evaluated out-of-sample versus in-sample.

While the split-sample strategy may find predictability differences that are consistent

with lookahead bias, it is an indirect test. Differences in predictive ability between sample splits may indicate differences in inherent predictability rather than lookahead bias. For example, if a model is worse at predicting stock returns in an out-of-sample split, it may be the case that the out-of-sample split corresponds to a period in which it was inherently more difficult to predict returns, or a short period with few samples. Conversely, a split-sample strategy may report similar predictive ability across sample splits—even if the analysis procedure has lookahead bias—if predictability is higher in the out-of-sample split.

This kind of analysis also assumes that a researcher’s analysis window includes information from after the language model’s pretraining cutoff date. In many cases, a researcher may be interested in analyzing a historical period that completely lies before a given language model’s pretraining cutoff date. The solution to this problem does not lie in moving around the analysis window, but instead in moving around the language model’s pretraining cutoff.

Language models with historical pretraining cutoffs already exist. Schweter et al. (2022) pretrain a language model on a multilingual corpus that ends in 1914. Manjavacas and Fonteyn (2021) pretrain a language model on an English-language corpus from 1450-1950. van Strien (2023) pretrains a semantic similarity model on an English-language corpus from 1920–1989. All of these models are available on the Hugging Face Hub. A potential challenge of using a language model with a pretraining cutoff that lies many years before the analysis period is that the distribution of language in the pretraining data may be less representative of the language in a researcher’s analysis period. Removing information that leaks the content of future events may also remove information that reveals relevant statistical properties of language. How can a researcher reduce the impact of this second effect?

A more systematic approach: Language models with time subscripts. Our proposed solution is for researchers to use language models whose pretraining cutoff dates lie before—but only shortly before—the analysis period. In essence, researchers can select from a family of language models with time subscripts. This model-selection procedure allows researchers to conduct analysis without lookahead bias from pretraining. In addition, it allows for the the pretraining corpus to potentially be more representative of the language in the researcher’s analysis period.

While it may be computationally expensive to train these models, some such models already exist. StoriesLM (Sarkar, 2024) is a family of transformer models that sequen-

tially expands the pretraining window. The model family is pretrained on news articles from the American Stories dataset (Dell et al., 2024) over the first half of the 20th century, and is available on the Hugging Face Hub. Each model in the family is trained on an additional year of pretraining data. Researchers can download these pretrained models and apply them to their analyses.

These models are only a start: There are several opportunities to research and develop new classes of language models with time subscripts. New models may use larger architectures, include additional historical data, or conduct richer sets of pretraining procedures—including those that involve language generation. New research could explore the properties of these model families—for example, by evaluating how changes in the temporal distribution of pretraining data affects language model performance, or how rolling forward the pretraining window affects language model representations.

5. Discussion

We discuss a form of lookahead bias that results from the pretraining procedure of language models. We develop direct tests to identify this bias, and find that it can affect analysis across domains in economics. We identify limitations of prompting-based approaches to counteract this bias.

The issues we raise are addressable. We identify one analysis procedure that is not subject to lookahead bias from pretraining: Selecting from a family of language models with time subscripts. Model families and analysis procedures that avoid these issues are publicly available, and there are clear next steps to research their statistical properties and improve their performance.

References

- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei**, “Language Models are Few-Shot Learners,” July 2020. arXiv:2005.14165 [cs]. 1
- Bybee, Leland**, “Surveying Generative AI’s Economic Expectations,” *SSRN Electronic Journal*, 2023. 5, 19
- Carlini, Nicholas, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song**, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” in “28th USENIX security symposium (USENIX security 19)” 2019, pp. 267–284. 5
- , **Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang**, “Quantifying memorization across neural language models,” *arXiv preprint arXiv:2202.07646*, 2022. 5
- Chen, Jiafeng and Suproteem Sarkar**, “A Semantic Approach to Financial Fundamentals,” in “Proceedings of the Second Workshop on Financial Technology and Natural Language Processing” - Kyoto, Japan May 2020, pp. 22–26. 6
- Dell, Melissa, Jacob Carlson, Tom Bryan, Emily Silcock, Abhishek Arora, Zejiang Shen, Luca D’Amico-Wong, Quan Le, Pablo Querubin, and Leander Heldring**, “American stories: A large-scale structured text dataset of historical us newspapers,” *Advances in Neural Information Processing Systems*, 2024, 36. 21
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova**, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 2019. arXiv:1810.04805 [cs]. 1
- Dodge, Jesse, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner**, “Documenting large web-text corpora: A case study on the colossal clean crawled corpus,” *arXiv preprint arXiv:2104.08758*, 2021. 5

- Eggers, Andrew C., Anthony Fowler, Jens Hainmueller, Andrew B. Hall, and James M. Snyder**, “On the Validity of the Regression Discontinuity Design for Estimating Electoral Effects: New Evidence from Over 40,000 Close Races,” *American Journal of Political Science*, January 2015, 59 (1), 259–274. 4, 14
- Elton, Edwin J., Martin J. Gruber, and Christopher R. Blake**, “Survivor Bias and Mutual Fund Performance,” *Review of Financial Studies*, October 1996, 9 (4), 1097–1120. 5
- Gentzkow, Matthew, Bryan Kelly, and Matt Taddy**, “Text as Data,” *Journal of Economic Literature*, September 2019, 57 (3), 535–574. 1
- Gichoya, Judy Wawira, Imon Banerjee, Ananth Reddy Bhimireddy, John L Burns, Leo Anthony Celi, Li-Ching Chen, Ramon Correa, Natalie Dullerud, Marzyeh Ghassemi, Shih-Cheng Huang et al.**, “AI recognition of patient race in medical imaging: a modelling study,” *The Lancet Digital Health*, 2022, 4 (6), e406–e414. 4, 6
- Glasserman, Paul and Caden Lin**, “Assessing Look-Ahead Bias in Stock Return Predictions Generated by GPT Sentiment Analysis,” *The Journal of Financial Data Science*, 2024, 6 (1), 25–42. 1, 5, 6, 17, 19
- Golchin, Shahriar and Mihai Surdeanu**, “Time travel in LLMs: Tracing data contamination in large language models,” *arXiv preprint arXiv:2308.08493*, 2023. 5
- Halawi, Danny, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt**, “Approaching Human-Level Forecasting with Language Models,” February 2024. arXiv:2402.18563 [cs]. 1, 19
- Hansen, Stephen, Peter John Lambert, Nicholas Bloom, Steven Davis, Raffaella Sadun, and Bledi Taska**, “Remote Work across Jobs, Companies, and Space,” Technical Report w31007, National Bureau of Economic Research, Cambridge, MA March 2023. 6
- Inan, Huseyin A., Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim**, “Training Data Leakage Analysis in Language Models,” February 2021. arXiv:2101.05405 [cs]. 5
- Jha, Manish, Hongyi Liu, and Asaf Manela**, “Does Finance Benefit Society? A Language Embedding Approach,” *SSRN Electronic Journal*, 2020. 6

- , **Jialin Qian, Michael Weber, and Baozhong Yang**, “ChatGPT and Corporate Policies,” *SSRN Electronic Journal*, 2023. 5, 19
- Kapoor, Sayash and Arvind Narayanan**, “Leakage and the Reproducibility Crisis in ML-based Science,” July 2022. arXiv:2207.07048 [cs, stat]. 5
- Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan, and Ziad Obermeyer**, “Prediction Policy Problems,” *American Economic Review*, May 2015, 105 (5), 491–495. 1
- Li, Edward Xuejun, Zhiyuan Tu, and Dexin Zhou**, “The Promise and Peril of Generative AI: Evidence from ChatGPT as Sell-Side Analysts,” *SSRN Electronic Journal*, 2023. 19
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov**, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” July 2019. arXiv:1907.11692 [cs]. 1, 8
- Lopez-Lira, Alejandro and Yuehua Tang**, “Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models,” September 2023. arXiv:2304.07619 [cs, q-fin]. 1, 19
- Manjavacas, Enrique and Lauren Fonteyn**, “MacBERTh: Development and evaluation of a historically pre-trained language model for English (1450-1950),” in “Proceedings of the Workshop on Natural Language Processing for Digital Humanities” 2021, pp. 23–36. 20
- Nylund, Kai, Suchin Gururangan, and Noah A. Smith**, “Time is Encoded in the Weights of Finetuned Language Models,” December 2023. arXiv:2312.13401 [cs]. 16
- Oren, Yonatan, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto**, “Proving Test Set Contamination in Black Box Language Models,” November 2023. arXiv:2310.17623 [cs]. 5
- Pawelczyk, Martin, Seth Neel, and Himabindu Lakkaraju**, “In-Context Unlearning: Language Models as Few Shot Unlearners,” October 2023. arXiv:2310.07579 [cs]. 16
- Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and others**, “Improving language understanding by generative pre-training,” 2018. Publisher: OpenAI. 1

Rajan, Raghuram, Pietro Ramella, and Luigi Zingales, “What Purpose Do Corporations Purport? Evidence from Letters to Shareholders,” Technical Report w31054, National Bureau of Economic Research, Cambridge, MA March 2023. 6

Sainz, Oscar, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre, “NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark,” *arXiv preprint arXiv:2310.18018*, 2023. 5

Sarkar, Suproteem K., “StoriesLM: A Family of Language Models With Sequentially-Expanding Pretraining Windows,” March 2024. 5, 20

Schweter, Stefan, Luisa März, Katharina Schmid, and Erion Cano, “hmBERT: Historical Multilingual Language Models for Named Entity Recognition,” July 2022. *arXiv:2205.15575 [cs]*. 20

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample, “LLaMA: Open and Efficient Foundation Language Models,” February 2023. *arXiv:2302.13971 [cs]*. 11, 15, 27

van Strien, Daniel, “headline-similarity,” October 2023. 20

Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann, “BloombergGPT: A Large Language Model for Finance,” December 2023. *arXiv:2303.17564 [cs, q-fin]*. 6

A. Appendix

A.1. Zoom Example

The following speech segment from an earnings call by Zoom Video Communications, Inc held on September 5, 2019 was used to fill in the prompt for Figure 1.

Hello, everyone, and welcome to Zoom's earnings webinar for the second quarter of fiscal 2020. Joining me today will be Zoom's Founder and CEO, Eric Yuan; and Zoom's CFO, Kelly Steckelberg.

Our earnings press release was issued today after the market close and may be downloaded from the Investor Relations page on the zoom.com website. Also, on this page, you'll be able to find a copy of today's prepared remarks and a slide deck with financial highlights that, along with our earnings press release, include a reconciliation of GAAP to non-GAAP financial results.

During this call, we will make forward-looking statements about our future financial performance and other future events or trends, including guidance. These statements are only predictions that are based on what we believe today and actual results may differ materially. These forward-looking statements are subject to the risks and other factors that could affect our performance and financial results and which we discuss in detail in our filings with the SEC, including today's earnings press release and our latest 10-Q. Zoom assumes no obligation to update any forward-looking statement that we may make on today's call. And with that, let me turn the discussion over to Eric.

Eric S. Yuan, Zoom Video Communications, Inc. - Founder, President, CEO, Chairman & Secretary Thank you, Tom. Hey, thank you all, and welcome to everyone joining us on today's Zoom webinar.

I'm very pleased to report that we had a remarkable second quarter and continue to deliver a unique combination of high growth with increased profitability and free cash flow. As Kelly will discuss in a moment, the first half momentum in our business has enabled us to meaningfully raise our revenue and the profitability outlook for the rest of the year.

Our strong second quarter results are evidence that organizations are turning to Zoom as their strategic technology partner to help them improve their communication and collaboration. While we continue to attract the customers of all sizes and across several industry segments, let's discuss one of our largest wins of the quarter.

I'm proud to welcome HSBC to the Zoom family. HSBC is one of the largest financial services organizations in the world with over 3,900 offices in 67 countries. HSBC will standardize on Zoom platform by deploying to 290,000 hosts and to 5,500 conference rooms. HSBC will consolidate onto Zoom's video-first unified communications platform for both internal and external meetings. By standardizing on Zoom, HSBC will consolidate costs and create an enhanced frictionless experience for end users. This enterprise-wide deployment represents one of the

largest customer commitments to Zoom in our history and reflects our growing momentum with global customers.

The output in the figure was generated using the Llama 2-70B Chat model (Touvron et al., 2023). The parameters used for the generation were {temperature = 0, top_k = 60, top_p = 1, repetition_penalty = 1, max_tokens = 128}.

A.2. Data Processing

Main results: Language model generations Our main results are generated from language models using the following parameters.

- Firm risk generation:
 - architecture: Llama 2-70B Chat
 - temperature: 0
 - top_p: 1
 - repetition_penalty: 1
 - max_tokens: 128
- Election winner generation:
 - architecture: Llama 2-70B
 - temperature: 0
 - top_p: 1
 - repetition_penalty: 1
 - max_tokens: 6

Section 2 includes the prompts for each of these generations.

Additional results: Effects of masking In Section 3.2, we discuss how language models can predict identifiers from language even if direct references to those identifiers are removed. All of the language model outputs in this section were generated using the OpenAI API using the “gpt-4-0125-preview” checkpoint between March 5–7, 2024 .

For year imputation, we first replace all string matches of years and month names from each earnings call segment with the string “_”. We then impute names using the following prompt

The following is a segment of the earnings call of a firm, in which all dates have been replaced with the character _

[earnings call segment]

Predict the most likely year for this earnings call. Return only a year.

For name imputation, we first remove name references using the prompt

The following is a segment of the earnings call of the firm [firm name]

Return the segment, but replace all instances of a firm's name with the character _

[earnings call segment]

We verify for each of the 100 calls that firm names have been removed. We then impute names using the prompt

The following is a segment of the earnings call of a firm whose name has been replaced with the character _

[earnings call segment]

Predict the most likely company name for this earnings call. Return only a company name.

For the second name imputation result, we remove name and product references using the prompt

The following is a segment of the earnings call of the firm [firm name]

Return the segment, but replace all instances of a firm's name and all instances of the firm's products with the character _

[earnings call segment]

We then impute firm names using the prompt

The following is a segment of the earnings call of a firm whose name and products have been replaced with the character _

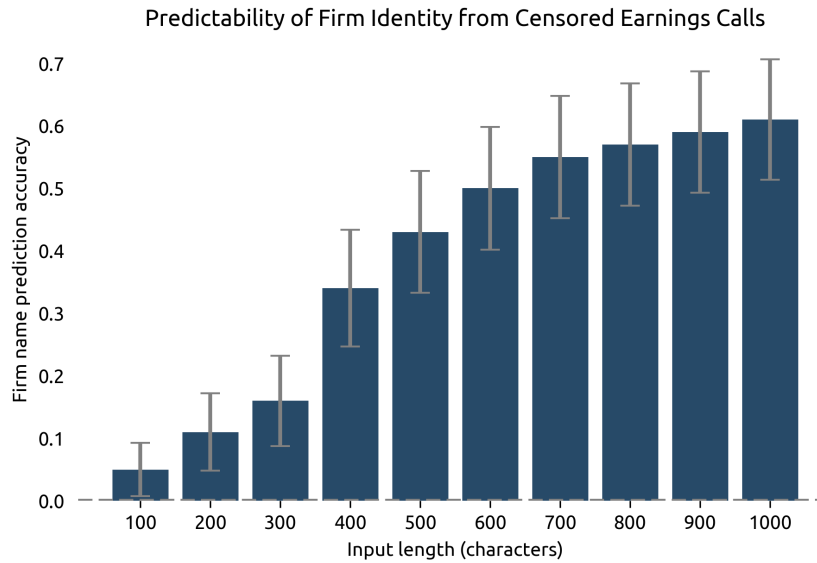
[earnings call segment]

Predict the most likely company name for this earnings call. Return only a company name.

A.3. De-Identification of Masked Prompts Across Input Lengths

We assess the ability to infer firm names from name-censored earnings calls across subsets of the calls of varying length. We use the same 100 earning calls speeches as in Section 3.2, but limit the number of characters used to query the model. We skip the

Figure A1: Identification accuracy increases as input length increases.



Notes – This figure reports results from using language models to infer firm identity from earnings calls segments that censor the firm’s name. Each bar plots the accuracy of a language model’s prediction of the firm’s identity in an earnings call with the company name removed, conditional on the number of characters from the call used to query the language model.

first 100 characters of the earnings call speech, which are typically used for greetings, and then input the next k characters from the call for $k \in \{100, 200, \dots, 1000\}$. We use the same identification procedure as in Appendix A.2. All queries were run on June 20, 2024.

A.4. Acknowledgement

Sarkar is supported by a National Science Foundation Graduate Research Fellowship and a Two Sigma PhD Fellowship. This project is supported, in part, by funding from Two Sigma Investments, LP. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of Two Sigma Investments, LP. Neither the National Science Foundation nor Two Sigma had a right to review this paper prior to its circulation.