Lookahead Bias in Pretrained Language Models

Anonymous Authors[1]

Abstract

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 pretraining 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 forecasting: 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.

1. Introduction

Pretrained language models are trained on large datasets of historical language, 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.

In addition to containing information about statistical properties of language, the pretraining data of these models also contains information about the *historical events* encoded in the language. This information may lead analysis that uses pretrained models to exhibit lookahead bias (Glasserman & Lin, 2024; Lopez-Lira & Tang, 2023; Halawi et al., 2024).

For example, a researcher 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 forecasting ability.

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 how to address the bias through pretraining strategies that leverage the structure of time-indexed language data.

2. Framework

Suppose a researcher aims to forecast an outcome Y_{t+1} from language data X_t. The researcher requires that the forecast only uses information available in a prespecified information set $\mathcal{I}$. The researcher's object of interest is the conditional expectation $\mu(X_t; \mathcal{I}) \equiv \mathbb{E}[Y_{t+1} \mid X_t, \mathcal{I}]$.

A language model is a function $f(X; \mathcal{M})$ that takes as input data X and whose pretraining data and parameters induce information set $\mathcal{M}$. We include $\mathcal{M}$ in our notation to make a language model's dependence on the induced information set explicit.

A researcher who uses a language model to make forecasts faces a potential issue: The language model may use information that is not contained in the desired information set. We refer to this issue as **temporal leakage**, i.e. $\mathcal{M} \not\subseteq \mathcal{I}$.

Lookahead bias occurs when temporal leakage influences forecasts. Define $\varepsilon_{t+1} \equiv Y_{t+1} - \mu(X_t; \mathcal{I})$ to be the irreducible error in this forecasting task. ε_{t+1} corresponds to the component of Y_{t+1} that is unpredictable from X_t and $\mathcal{I}$. No function of a language sequence X_t and the information set $\mathcal{I}$ can be correlated with ε_{t+1}.

Suppose a researcher uses language model outputs to form forecasts $\widehat{\mu}(X_t; \mathcal{I}) = g(f(X_t; \mathcal{M}); \theta)$. If these forecasts are correlated with the irreducible error, the model is using information not in the information set. We refer to this issue as **lookahead bias**.

We define lookahead bias in this setting as

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

3. Evidence of Leakage and Lookahead Bias

We develop tests to identify temporal leakage and lookahead bias 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.

3.1. Evidence of Temporal Leakage

To test for temporal leakage, we consider the task 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 can occur if the language sequence *"COVID-19"*, which was introduced in February 2020, 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.

Testing for direct temporal leakage To test for 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 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.

For each call, we query the model with the following prompt, substituting the bracketed terms 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"

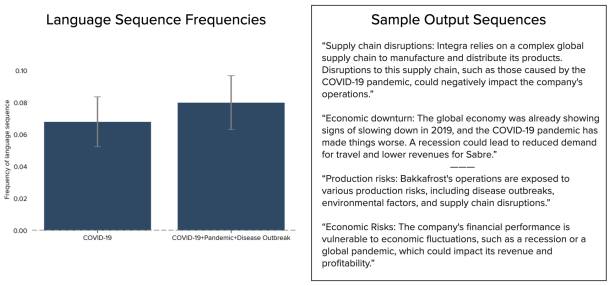

Figure 1. Language model outputs directly leak information about future events. This figure summarizes outputs from language models prompted with corporate earnings calls from 2019 and instructed to predict risks for the firm in each call. The left panel reports that 6.8% of generations include the language 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.

We generate outputs using the Llama-2 70B language model (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. 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 RLHF.

Evidence of direct leakage Figure 1 shows the frequencies of the language sequences *"COVID-19"* as well as *"COVID-19," "Pandemic,"* or *"Disease Outbreak"* in the language model outputs. All matches are case-insensitive. The first finding—that 6.8% of generations include the language sequence *"COVID-19"*—is clear evidence of direct temporal 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. The benchmark for this statistic should be 0% for a model that does not exhibit leakage with respect to November 2019.

Testing for indirect leakage 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 the pandemic are 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 sam-

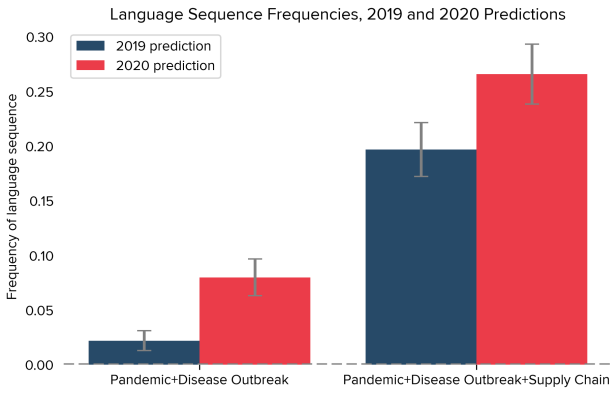

Figure 2. Language model outputs indirectly leak information about future events. This figure reports the frequency of language sequences in outputs from language models prompted with corporate earnings calls and instructed to predict future risks. We color in blue the results that use outputs from models 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.

ple of 1,000 earnings calls that took place between September 1, 2018 and November 30, 2018. We modify the prompt to predict risks in 2019.

Evidence of indirect leakage Figure 2 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"*}. As before, all matches are case-insensitive. 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 a direct test for information leakage like the presence of the new language sequence *"COVID-19,"* it suggests 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.

3.2. Evidence of Lookahead Bias

How can we find evidence of lookahead bias in predictions that use language model outputs? Recall from Section 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 obtain the results of 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}
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 so that generated output corresponds only to the name of a candidate.

Evidence of lookahead bias Figure 3 evaluates the accuracy of the language-model based prediction for elections within varying margins of victory among the top two candidates. 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.

4. Limitations of Mitigation Strategies

We discuss the limitations of prompting- and masking-based strategies to mitigate lookahead bias.

Limitations of Prompting Recent work has argued that prompt design may lead language models to unlearn information from training data (e.g. Pawelczyk et al., 2023). As information about time appears to be encoded in the weights of language models (Nylund et al., 2023), one might believe

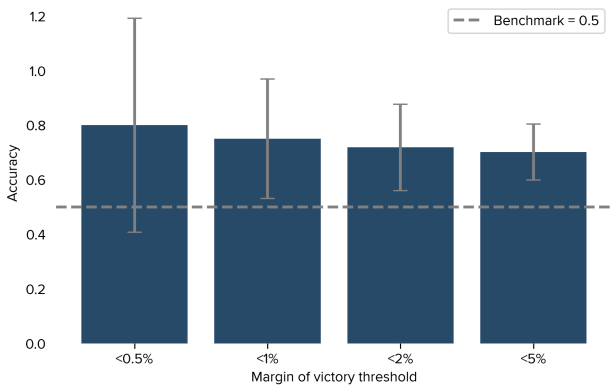

Figure 3. Predictions that use language model outputs can determine the results of "natural experiments." This figure reports the accuracy of an analysis procedure that uses language model outputs to predict U.S. House election winners. Each bar reflects the accuracy of the procedures for races within 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.

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 results in Section 3.1 and Section 3.2 include prompts with instructions to not use information from after the analysis period. For both of these results, prompting alone is not enough to address lookahead bias.

Limitations of Masking Another approach to addressing information leakage is to mask identifying information from the text used to query a language model (e.g. Glasserman & Lin, 2024). The reasoning behind this approach is that if a prompt does not contain any identifying information about the firm (for example name, industry, management), the model's generation would not correlate with future events related to the firm.

Does censoring identifying information guarantee a model is unable to infer a firm's identity? We find this is not always the case. In Appendix A.1, we show that language models can predict time periods and firm identifiers from censored data. 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 information can also remove context that would be important for forecasting.

5. Addressing the Bias Through Pretraining

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 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 sequentially 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.

6. Discussion

We discuss how pretraining can introduce temporal leakage and lookahead bias into language models. We develop direct tests to identify this bias, and find that it can affect analysis across multiple domains. 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. Pretrained model families that help to avoid these issues are publicly available, and there are clear next steps to research their statistical properties and improve their performance.

Impact Statement

Language models are increasingly deployed in high-stakes settings in which information cutoffs are key—including in financial analysis, political science, and economic forecasting. We find that language models exhibit lookahead bias—they can leak information not contained in prespecified information sets. This bias can reduce the credibility and reliability of language models applied in these high-stakes settings.

Our work identifies pretraining and analysis strategies that address the lookahead bias issue, and points people toward models to use in these domains. The strategies we discuss may improve the reliability of language models applied to these settings.

References

Dell, M., Carlson, J., Bryan, T., Silcock, E., Arora, A., Shen, Z., D'Amico-Wong, L., Le, Q., Querubin, P., and Heldring, L. American stories: A large-scale structured text dataset of historical us newspapers. *Advances in Neural Information Processing Systems*, 36, 2024.

Eggers, A. C., Fowler, A., Hainmueller, J., Hall, A. B., and Snyder, J. M. 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*, 59(1):259–274, January 2015. ISSN 0092-5853, 1540-5907.

Glasserman, P. and Lin, C. Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis. *The Journal of Financial Data Science*, 6(1):25–42, 2024.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching Human-Level Forecasting with Language Models, February 2024. arXiv:2402.18563.

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

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

Nylund, K., Gururangan, S., and Smith, N. A. Time is Encoded in the Weights of Finetuned Language Models, December 2023. arXiv:2312.13401.

Pawelczyk, M., Neel, S., and Lakkaraju, H. In-Context Unlearning: Language Models as Few Shot Unlearners, October 2023. arXiv:2310.07579.

Sarkar, S. K. Storieslm: A family of language models with sequentially-expanding pretraining windows. *SSRN*, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models, February 2023. arXiv:2302.13971.

A. Appendix

A.1. Limitations of Information Masking

Another approach to addressing information leakage is to mask identifying information from the text used to query the language model (Glasserman & 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.

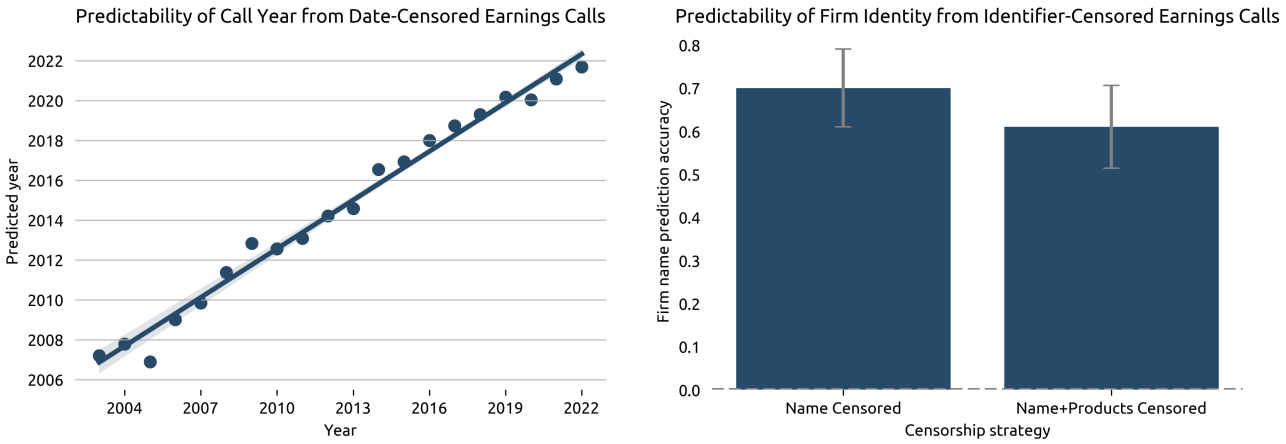

Figure 4. Masking identifiers from language model prompts does not guarantee the prompts are de-identified. This figure reports results from using language models to infer identifying information from earnings calls segments that censor identifying information. The left panel is a binned scatter plot 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.

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 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 4 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 3.1. 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 4 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. We additionally show that de-identification becomes less effective as the amount of information used to query a language model increases—Figure 5 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 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.

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

Additional results: Effects of masking In Appendix A.1, 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

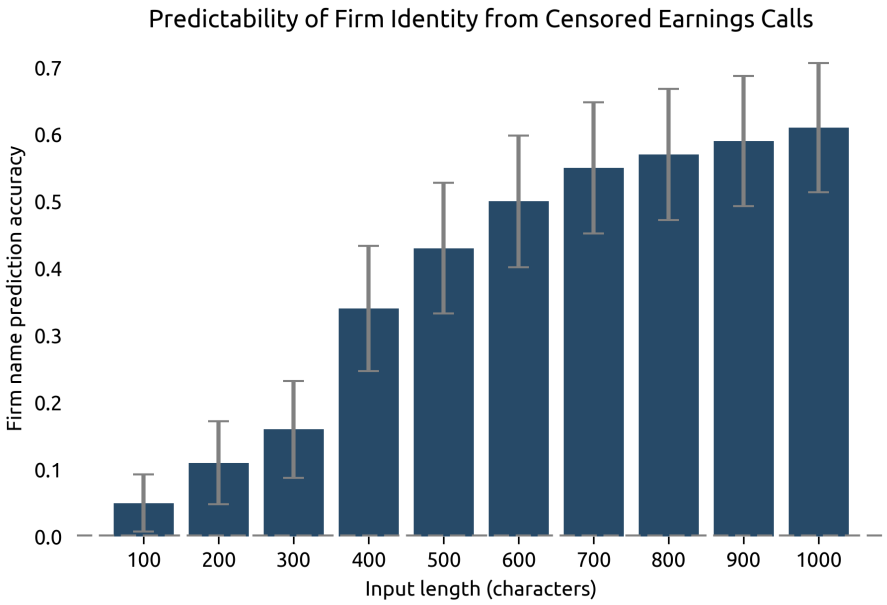

Figure 5. Identification accuracy increases as input length increases. This figure reports results from using language models to infer firm identity from earnings calls segments that censor the firm's name. Each bar plots the accuracy of a language model's prediction of the firm's identity in an earnings call with the company name removed, conditional on the number of characters from the call used to query the language model.

> 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 Appendix A.1, but limit the number of characters used to query the model. We skip the 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 $k \in \{100, 200, \ldots, 1000\}$. We use the same identification procedure as in Appendix A.2.