Entity Neutering

Joseph Engelberg
UC San Diego

Asaf Manela
Wash. U.

William Mullins
UC San Diego

Luka Vulicevic
*UC San Diego**

March 17, 2025

Abstract

Cutting-edge LLMs are trained on recent data, creating a concern about look-ahead bias. We propose a simple solution called *entity neutering*: using the LLM to find and remove all identifying information from text. In a sample of one million financial news articles, we verify that, after neutering, ChatGPT and other LLMs cannot recognize the firm or the time period for about 90% of the articles. Among these articles, the sentiment extracted from the raw text and the neutered text agree 90% of the time and have similar return predictability, with the difference providing an upper bound on look-ahead bias. The evidence here suggests that LLMs are able to effectively neuter text while maintaining semantic content. For look-ahead bias, LLMs can be both the problem and the solution.

Keywords: Large Language Models, Look-ahead Bias, Textual Analysis

*Contact: jengelberg@ucsd.edu, amanela@wustl.edu, wmullins@ucsd.edu, lvulicevic@ucsd.edu. We thank David Hirshleifer for valuable comments, the Rady Research Fund, and Heng-chia Liu for excellent research assistance.

1. Introduction

Much of asset pricing is concerned with how information is impounded into prices, and a substantial amount of data in finance is text. For this reason, a literature has emerged dedicated to properly extracting content from text.[1]

Large Language Models (LLMs) have the potential to transform the way text is processed in financial settings, and their use by practitioners and academics has increased exponentially since OpenAI's release of GPT-3 in 2020. The academic papers that use LLMs to extract content and make predictions often include a caveat about look-ahead bias (e.g., Bybee, ????, Jha et al., 2024). For example, if we ask GPT-3.5 to read Apple's 2011 10-k and use that information to predict Apple's future stock return, some of its ability to do so may come from the fact that GPT-3.5 is trained on data through September 2021. If GPT-3.5 has been trained on articles about Apple's success in 2012 or the rapid growth of the smartphone industry in that year, it could be using this information to make its prediction. Newer LLMs - such as Claude (Anthropic) and Gemini (Google) - are frequently updated or can use web searches to answer questions, subjecting the extraction of any data from historical text to look-ahead bias.

This paper proposes a methodology, called *entity neutering*, to address look-ahead bias in LLM predictions. Recognizing the identity of the subject of a text (i.e., the "entity") is a necessary condition for look-ahead bias to operate; therefore, if the LLM cannot recognize the entity then it cannot use future information about it. However, entity neutering is more complicated than simply removing a firm's name from the text. As an example, consider the following Dow Jones Newswire story about General Mills (GIS) on June 22, 2006:

[1]See for example, Tetlock, Saar-Tsechansky, and Macskassy (2008), Loughran and McDonald (2011), Gentzkow, Kelly, and Taddy (2019), Kelly, Manela, and Moreira (2021), Garcia, Hu, and Rohrer (2023), Hoberg and Manela (2025).

Original Text	**Firm Masked**	**Entity Neutered**
General Mills Inc. (GIS) was upgraded to buy from neutral at **Merrill Lynch** on valuation, with the broker saying there's upside to its **2007** earnings estimate of **$3.16** a share. "The integration of **Pillsbury**, which has been such a detractor from the consistency that **General Mills** had been known for, is essentially over. We think now is the time that **Mills** will regain credibility with the Street and be rewarded with a higher multiple," the brokerage said.	*Company_1* was upgraded to buy from neutral at **Merrill Lynch** on valuation, with the broker saying there's upside to its **2007** earnings estimate of **$3.16** a share. "The integration of **Pillsbury**, which has been such a detractor from the consistency that *Company_1* had been known for, is essentially over. We think now is the time that *Company_1* will regain credibility with the Street and be rewarded with a higher multiple," the brokerage said.	*Company_1* was upgraded to buy from neutral at *Brokerage_1* on valuation, with the broker saying there's upside to its *time_1* earnings estimate of *number_a* a share. "The integration of *product_x*, which has been such a detractor from the consistency that *Company_1* had been known for, is essentially over. We think now is the time that *Company_1* will regain credibility with the Street and be rewarded with a higher multiple," the brokerage said.

The first column is the original article, the second column masks only the firm name and ticker, and the final column is the result of entity neutering. Simply replacing the firm name and ticker with a stand-in like "*Company 1*" is not enough to assuage concerns about look-ahead bias because the LLM can identify General Mills from the other information (its broker, Pillsbury, the year and the earnings per share estimate). The third column masks this additional information, but retains the content of the article, which emphasizes the analyst's upgrade and belief that General Mills will regain credibility with Wall Street and

receive a higher multiple.

We perform this entity neutering on 911,283 articles from the Dow Jones Archive from 2000 through 2009. What distinguishes entity neutering from prior work (Glasserman and Lin, 2023, Sarkar and Vafa, 2024) is that we ask the LLM, GPT-4o mini, to anonymize the article so that a sophisticated reader could not guess the subject firm of each article.[2] Armed with this prompt, GPT effectively masks not only the subject firm but also additional details which could be used to infer the identity of the firm. In this way, we are using the LLM against itself, i.e., we are asking it to seek out and mask information that it or another LLM would naturally use to identify the entity.

We confirm that entity neutering is effective by resubmitting the neutered article to a new instance of GPT and two other leading LLMs, Llama 3 and DeepSeek R1. All LLMs struggle to correctly guess the subject firm in neutered articles, with GPT guessing correctly 9% of the time and Llama and DeepSeek guessing correctly around 7% of the time. The LLM's ability to guess the year of publication is even lower. For example, GPT is able to guess the year correctly only 2.3% of the time.

To recap, among the 911,283 articles, after neutering, GPT is only able to identify the firm or the year in 99,782 articles. We drop these "identifiable" articles from our sample and are left with 811,501 "unidentifiable" articles. One potential concern with this body of neutered texts is that it may have "thrown out the baby with the bathwater." By removing far more than just the firm name, we may have achieved our goal of preventing the LLM from recognizing the subject firm, but at the cost of losing the semantic content of the article.

To investigate this, we ask GPT to identify whether each article constitutes a positive or negative signal for the future returns of the subject firm, and additionally ask for the associated probability, as a measure of sentiment strength. We do this for both the raw and neutered articles. The sentiment of the neutered text matches the sentiment of the raw text

[2]We use GPT4o-mini via the API, which does not have access to the internet.

90.2% of the time.[3] The strong correlation between the raw and neutered texts suggests little information was lost in the neutering process, at least for the purpose of extracting textual sentiment.

Our main analysis compares return predictability using the LLM-extracted sentiment from the *neutered* text versus the *raw* text. If the return predictability coming from the raw text is powered by look-ahead bias, then we would expect to find a sharp drop off in predictability when switching to the neutered text. After all, if the LLM is looking ahead — by knowing how the subject firm will perform in the future — it can no longer use this advantage when the subject firm is unknown.

This is not what we find: the return predictability of the neutered text is similar to that of the raw text. Positive sentiment extracted from the neutered text predicts next-day abnormal returns of 9.1 bps, while from the raw text this number is 9.9 bps, with the difference indistinguishable from zero. Looking at days $t+2$ through $t+5$, we continue to find no statistical difference. We find a similar pattern for negative sentiment articles.

Only when we condition on strong sentiment do we see differences between the predicted returns from the raw and neutered texts. For example, positive articles with above-median sentiment strength (*Strong sentiment*) predict next day abnormal returns of 13.2 bps for the raw text and 10.6 bps for the neutered text. The corresponding figures among strongly negative articles are -19.7 and -14.1. Both are statistically different from each other.

We find similar results in a regression setting with controls analogous to those in Tetlock et al. (2008). For example, when predicting next day abnormal return, the coefficient on *Strong negative* sentiment is -20.5 bps when using the raw text, but -14.1 bps when using the neutered text. This wedge holds even on event days after $t+1$. For example, when predicting returns on days $t+2$ through $t+5$, the raw versus neutered coefficients on *Strong negative* sentiment are -24.5 bps and -18.8 bps; for *Strong positive* sentiment they are 11.2 bps and 7.0 bps.

[3] Among those that match, we rank the sentiment strength for both the raw and neutered articles. The rank correlation among positive and among negative sentiment articles is 72% in both cases.

The return predictability of the raw text can be larger than that of the neutered for two reasons: (i) look-ahead bias from the LLM or (ii) better information content from the raw text. In Section 2, we propose an econometric framework for thinking about the look-ahead bias problem which builds on Ludwig, Mullainathan, and Rambachan (2024). We show that, under reasonable assumptions, the *difference* in predictability between the raw and neutered text constitutes an *upper bound* on the contribution of look-ahead bias to return predictability. Using these bounds and our specifications with strong sentiment, we find that the improvement in return predictability — as measured by a reduction in mean squared error (MSE) — coming from look-ahead bias is *at most* 38% when predicting next-day returns and 32% when predicting returns on days $t+2$ through $t+5$. The upper bounds are 17% and 14% when using the specifications where we group sentiment into the coarser positive and negative categories.

Of course, as LLMs become more powerful, these conclusions could change. Future LLMs might be more effective at looking ahead when extracting measures of sentiment or they might be able to extract more information from raw (vs. neutered) text. However, because our approach uses the LLM itself to solve the problem of look-ahead bias, neutering will also become more effective over time. Thus, the analysis here provides a recipe for dealing with concerns about LLMs' look-ahead bias in the future. First, we recommend asking the LLM to neuter the text so that the subject entity, as well as tangential but identity-related information, is unidentifiable. Next, we recommend feeding the neutered text back to the LLM to see if it can identify the entity from the neutered text. If entity neutering is successful, the identification rate for the LLM should be small. Finally, we recommend dropping the observations in which the LLM could identify the entity in question and performing the analysis using both the neutered and the raw text among this subset. If the researcher's conclusions are generally unchanged using the neutered versus raw text, then this is evidence that their results are not driven by look-ahead bias. If they are different, we recommend calculating the bounds suggested here in order to assess at most how much of

the contribution to predictability may be coming from look-ahead bias.

Our paper relates to two strands of the finance literature. The first focuses on measuring content from text for prediction purposes. Beginning with Tetlock (2007) and Tetlock et al. (2008), researchers have become interested in extracting the information content from financial text for the purposes of predicting important variables such as stock returns or firm earnings. While this literature began with a simple bag-of-words approach (Loughran and McDonald, 2011), it has evolved as text-based algorithms and computational techniques have improved (for surveys see Section 3 of Kearney and Liu (2014) and Hoberg and Manela (2025); for more recent approaches see Ke, Kelly, and Xiu (2019), Lopez-Lira (2023) and Bybee, Kelly, and Su (2023)). Because LLMs represent the cutting edge of textual sentiment extraction for prediction purposes (Chen, Kelly, and Xiu, 2022), our contribution is to propose a method which allows researchers to use cutting-edge LLM models while addressing the look-ahead bias problem.

We also contribute to the burgeoning LLM literature in finance and accounting. While some authors use LLMs in settings where look-ahead bias is not an issue (Eisfeldt, Schubert, and Zhang, 2023), many uses involve settings in which the LLM is asked to predict a variable during a time period it may have been trained on. Existing work takes two approaches to the look-ahead bias problem. The first restricts the use of the LLM so that its predictions are always out of sample, ensuring *chronological consistency* (Sarkar and Vafa, 2024, Ludwig, Mullainathan, and Rambachan, 2024, He, Lv, Manela, and Wu, 2025).[4] While some papers have implemented this approach by applying LLMs exclusively to time periods after their training cutoff dates (e.g., Lopez-Lira and Tang, 2023), a more common strategy is to analyze periods after the cutoff date as a robustness test (e.g., Bybee, ????, Jha, Qian, Weber, and Yang, 2024).

[4]Sarkar (2024) provides the Stories LM models, with training cutoff dates from 1901 through 1963, while He, Lv, Manela, and Wu (2025) provides the ChronoBERT and ChronoGPT series of models, with cutoff dates from 1999 through 2024. These transformer models are not instruction fine-tuned and so require an additional training or regression step before they can be used to, for example, extract sentiment or predict returns.

The second approach to the look-ahead bias problem is to mask the name of the relevant entity before using the text for prediction (Glasserman and Lin, 2023, Sarkar and Vafa, 2024, Jha et al., 2024). Entity neutering belongs to this line of work. Our innovation is to ask the LLM itself to use its best judgment to neuter the text of any identifying information. To the best of our knowledge, entity neutering achieves recognition rates for firms far below what has been found in the literature. For example, after using our approach, an LLM can recognize the subject firm less than 10% of the time, while in Sarkar and Vafa (2024) this number is 70% (and 65% in our sample using the same approach). We also contribute by introducing a methodology by which researchers can estimate bounds on the share of their predictability that is coming from look-ahead bias.

The remainder of the paper proceeds as follows. Section 2 provides an econometric framework, Section 3 describes the data, and Section 4 discusses the empirical analysis. Section 5 concludes.

2. Econometric framework

We build on the econometric framework developed by Ludwig et al. (2024). They define training leakage as the phenomenon in which texts in the researcher's sample are oversampled in the LLM's training data, so that the oversampled texts exhibit unusual predictive ability. We extend their analysis by incorporating a *neutering function* designed to mitigate training leakage when using large language models (LLMs) for prediction. Our goal is to apply a transformation that neuters the features responsible for this leakage while preserving as much of the predictive content of the texts as possible.

We begin by defining training leakage and showing how it relates to look-ahead bias. We then define a neutering function and describe its desired properties. Finally, we derive an easy-to-estimate upper bound on training leakage.

2.1 A panel data adaptation of Ludwig et al. (2024)

We begin by adapting the Ludwig et al. (2024) notation to a panel setting so that we can be better relate what they call *training leakage* to what Sarkar and Vafa (2024) call *look-ahead bias* in the LLM setting. Index time by $t = 1 \dots T$ and observations belonging to the same unit within a period (e.g., firm) by $i = 1 \dots I$ as common in panel settings. The indicators for training set inclusion are summarized by the matrix τ whose elements τ_{it} are sampling indicators for whether a particular string is collected in the training dataset. The relevant text pieces are $\mathcal{R} \subseteq \Sigma^*$. The researcher's dataset is a matrix d whose elements d_{it} indicate if the researcher collected a particular string, assumed to be all relevant texts.

Each text piece x_{it} is linked to observable economic variables (Y_{it+1}, W_{it}), so that altogether the researcher observes $(x_{it}, Y_{it+1}, W_{it})$ for each text piece with $d_{it} = 1$.

In a prediction problem, the researcher prompts an LLM trained on τ to predict the variable based on text pieces $\hat{Y}_{it+1} = m(x_{it}; \tau)$. For some non-negative loss function ℓ, the researcher calculates the sample average loss

$$\frac{1}{N} \sum_{i,t} D_{it} \ell \left(Y_{it+1}, m \left(x_{it}; \tau \right) \right)$$

where $N = \sum_{i,t} D_{it}$ is the number of text pieces collected.

Importantly, from the perspective of the researcher, the LLM is treated as a black box whose training set τ is a random variable. Define a context $Q(\cdot) \in \mathcal{Q}$ that summarizes the sampling distribution over economically relevant texts and beliefs about the LLM's sampling distribution over training datasets. $Q(\cdot)$ is a joint distribution over the sampling indicators $(D, \mathcal{T})$. Denote by $q_{it}^{\mathcal{T}|D}(\tau_{it}) = Q(\mathcal{T}_{it} = \tau_{it} | D_{it} = 1)$ the conditional probability that the string x_{it} is sampled by the LLM's training dataset given that it is sampled by the researcher. Denote the marginal probabilities by $q_{it}^{\mathcal{T}}(\tau_{it}) = Q(\mathcal{T}_{it} = \tau_{it})$ and $q_{it}^{D}(d_{it}) = Q(\mathcal{D}_{it} = d_{it})$.

Ludwig et al. (2024, Definition 1) defines an LLM with guarantee M as a general-purpose technology for prediction if, for every research context $Q(\cdot)$ and every loss function ℓ, the

following out-of-sample inferential property holds:

$$E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)\right] = E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)|\mathcal{T}=\tau\right], \tag{1}$$

In short, this means that that the average loss computed on the research sample is a valid proxy for the true (population) loss.

Ludwig et al. (2024, Proposition 1) shows that this property holds if and only if there is no training leakage; that is, if

$$\text{Training leakage} = -E_Q\left[\sum_{i,t} D_{it}\Big(\frac{q_{it}^{\mathcal{T}|D}}{q_{it}^{\mathcal{T}}} - 1\Big)\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)\right] = 0. \tag{2}$$

Intuitively, Equation (2) captures training leakage: if the probability that a text is in the training set is the same in the research sample as in the overall population (i.e., $q_{it}^{\mathcal{T}|D} = q_{it}^{\mathcal{T}}$ for all i, t with $D_{it} = 1$), then the ratio $q_{it}^{\mathcal{T}|D}/q_{it}^{\mathcal{T}}$ equals one and the leakage term is zero. If not, the loss computed on the research sample is biased. This condition is central to ensuring that the LLM generalizes out-of-sample.

2.2 Look-ahead bias

Look-ahead bias is a special case of training leakage in a time series or panel setting. Consider the case where the researcher believes that the LLM was trained on some text that is also included in the researcher's sample. Thus, for some time periods t and units i, the probability of text piece x_{it} appearing in the training sample, conditional on it being in the researcher sample ($D_{it} = 1$), is greater than the marginal probability: $q_{it}^{\mathcal{T}|D}\left(\tau_{it}\right) > q_{it}^{\mathcal{T}}\left(\tau_{it}\right)$.

For example, in the case where we know the LLM was trained on x_{it}, the conditional probability is 1 while the marginal probability is between 0 and 1. If the researcher also believes that the predictive loss is smaller for those texts which the LLM was trained on - perhaps because the texts it has seen were indicative of Y_{it+1} - then the training leakage

term (2) is positive, and the estimated sample loss underestimates the population loss:

$$E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)\right] = E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)|\mathcal{T}=\tau\right] + \text{Training leakage.} \tag{3}$$

The exact reason why x_{it} would be indicative of Y_{it+1} depends on the context. For a stark example, suppose x_{it} includes the following text:

> ABC Corp. (ABC:NYSE) launched a new XYZ phone at the CES conference on January 7, 2025 in Las Vegas.

The following day, another headline appears which describes the launch as a flop and a negative reaction by the stock market:

> Stock market news — January 8, 2025. ABC Corp. (ABC:NYSE) shares drop 30 percent after XYZ phone disappoints.

Suppose a researcher prompts an LLM to predict the next day stock market reaction based only on the new product announcement. If the LLM was trained on both pieces of text and was intelligent enough to connect them, it would be able to predict the negative market reaction $Y_{it+1} = -30\%$ better than an LLM that was trained on neither piece of text. In this context, the expected predictive loss would be abnormally low for an observation believed to be oversampled in the LLM's training set.

As Equation (3) shows, the negative covariation between the oversampling term $\left(\frac{q_{it}^{\mathcal{T}|D}}{q_{it}^{\mathcal{T}}} - 1\right)$ and the predictive loss term $\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)$ would result in an underestimation of loss in the general population, or equivalently, overestimation of out-of-sample predictive performance.

One solution to avoiding look-ahead bias is to make sure there is no overlap between the

training and research samples.[5] Developing timestamped models that allow researchers to guarantee there is no overlap between their training and research samples is an active topic of research. The Sarkar (2024, StoriesLM) and He et al. (2025, ChronoBERT and ChronoGPT) series of open-weight language models are early steps in this direction. But their modest size may provide insufficient language processing compared with state-of-the-art gated LLMs. In such cases, applying a transformation to the text before feeding it to the model may be preferred. We next formalize such a transformation and its desired properties.

2.3 Neutering Functions

To mitigate training leakage (or look-ahead bias), we introduce a *neutering function*, based on training data τ,

$$n(\cdot;\tau) : R \to R,$$

which transforms an original text x_{it} into a *neutered* version $n(x_{it};\tau)$.

2.3.1 Generalization Neutrality

We say that the compound model $m(n(\cdot;\tau);\tau)$ *generalizes* on the neutered texts if the neutered loss exhibits no training leakage. That is, the neutering model $n(\cdot;\tau)$ is *generalization neutral* for the predictive model $m(\cdot;\tau)$ if

$$E_Q\left[\sum_{i,t} D_{it}\Big(\frac{q_{it}^{\mathcal{T}|D}}{q_{it}^{\mathcal{T}}} - 1\Big)\ell\left(Y_{it+1}, m\left(n\left(x_{it};\tau\right);\tau\right)\right)\right] = 0. \tag{4}$$

Intuitively, this property means that the probability of a text being in the training set is either independent of its inclusion in the research sample, or that this oversampling is

[5]Suppose that the LLM $m(\cdot;\tau)$ is trained on data up to some *knowledge cutoff* period K while the research sample is entirely composed of observations after K. This means that $\tau_{it} = 0$ for all $t > K$, and in this research context, $d_{it} = 0$ for all $t < K$. It follows that for observations in the research sample, when $D_{it} = 1$, we know the conditional probability $q_{it}^{\mathcal{T}|D}(\tau_{it})$ that text piece x_{it} appears in the training sample is zero, as is the marginal probability $q_{it}^{\mathcal{T}}(\tau_{it})$ that text piece x_{it} is in the training sample. Thus, when there is no overlap between training and research samples, the training leakage term (2) vanishes.

orthogonal to the predictive loss based on these text pieces. In other words, $n(\cdot;\tau)$ removes the training leakage present in the original texts.

Testing for generalization neutrality

The goal of our neutering function is to reduce the covariance between the terms in Equation (4) to zero. A natural approach to testing for generalization neutrality is to ask the LLM whether it can identify the subject entity after de-identification. As Sarkar and Vafa (2024) notes:

> *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.*

In Section 4.2, we perform these tests by asking ChatGPT (and other LLMs) if it can recognize the subject firm or industry after the text has been transformed by our neutering function.

2.3.2 Loss Fidelity

Generalization neutrality alone is not desirable. Deleting all text pieces, for example, removes training leakage but destroys all information content. Thus, preserving the information content of text is also a desirable property which we formalize below.

We say that the neutering function $n(\cdot;\tau)$ exhibits *loss fidelity* for model $m(\cdot;\tau)$ if, out-of-sample, the expected loss based on the raw text is similar to the loss based on the neutered text:

$$E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(x_{it};\tau\right)\right)\right] = E_Q\left[\sum_{i,t} D_{it}\ell\left(Y_{it+1}, m\left(n\left(x_{it};\tau\right);\tau\right)\right)\right]. \tag{5}$$

Testing for Loss Fidelity

In Section 4, we test for loss fidelity by comparing the predictability of the original text with that of the neutered text. To the extent that they are different, we can estimate an

upper bound on training leakage based on in-sample comparisons between neutered versus raw text performance, as we show next.

2.4 A bound on training leakage

Suppose the researcher estimates average loss based on the neutered text and finds that it is higher than based on the raw text. What can the researcher infer about training leakage in this context?

Denote the increase in sample average loss due to neutering as

$$\Delta_Q\left(m,n,\ell;\tau\right)=E_Q\left[\sum_{i,t}D_{it}\left\{\ell\left(Y_{it+1},m\left(n\left(x_{it};\tau\right);\tau\right)\right)-\ell\left(Y_{it+1},m\left(x_{it};\tau\right)\right)\right\}|\mathcal{T}=\tau\right].$$

Further denote training leakage in the raw text by

$$leakage_{raw}=E_Q\left[\sum_{i,t}D_{it}\ell\left(Y_{it+1},m\left(x_{it};\tau\right)\right)\right]-E_Q\left[\sum_{i,t}D_{it}\ell\left(Y_{it+1},m\left(x_{it};\tau\right)\right)|\mathcal{T}=\tau\right],$$

and in the neutered text by

$$\begin{aligned}leakage_{neutered}=&E_Q\left[\sum_{i,t}D_{it}\ell\left(Y_{it+1},m\left(n\left(x_{it};\tau\right);\tau\right)\right)\right]\\&-E_Q\left[\sum_{i,t}D_{it}\ell\left(Y_{it+1},m\left(n\left(x_{it};\tau\right);\tau\right)\right)|\mathcal{T}=\tau\right].\end{aligned}$$

Manipulation based on the above definitions provides the following decomposition:

$$\underbrace{\Delta_Q\left(m,n,\ell;\tau\right)}_{\substack{\text{Sample loss increase}\\\text{due to neutering}}}=\underbrace{E_Q\left[\sum_{i,t}D_{it}\Big(\ell\left(Y_{it+1},m\left(n\left(x_{it};\tau\right);\tau\right)\right)-\ell\left(Y_{it+1},m\left(x_{it};\tau\right)\right)\right]}_{\text{Loss infidelity}}\tag{6}$$

$$+\underbrace{\left(leakage_{raw}-leakage_{neutered}\right)}_{\text{Reduction in leakage due to neutering}}.$$

As decomposition (6) reveals, an increase in sample loss, or equivalently a decrease in predictive performance in sample due to neutering, can result from training leakage in the raw text. But it can also arise because of loss infidelity or because of lingering training leakage post-neutering. We can make progress if we assume that neutering does not improve prediction, which we find plausible, and that the neutering model attains generalization neutrality.

Proposition 1 (Bound on training leakage)**.** *Suppose that neutering model $n(\cdot, \tau)$ is generalization neutral for the predictive model $m(\cdot; \tau)$ and that neutering does not improve prediction (loss infidelity is nonnegative). Then training leakage (or look-ahead bias) in the raw text is bounded from above by the increase in sample loss due to neutering. Formally,*

$$leakage_{raw} \leq \Delta_Q(m, n, \ell; \tau).$$

This result shows that under reasonable assumptions about the neutering model, a simple in-sample estimate of the reduction in performance provides an upper bound on training leakage in the raw data.

In Section 4 we provide estimates of this upper bound in our empirical setting.

3. Data

To illustrate entity neutering in practice, we begin with a text-based dataset which has been shown to have return predictability in practice: the Dow Jones archive. The archive contains all stories from the Dow Jones Newswire and the Wall Street Journal, as well as firm-specific press releases which come across the wire. Beginning with Tetlock et al. (2008) which shows a dictionary-based approach to sentiment extraction can predict next-day returns, many papers have found that the textual sentiment extracted from the Dow Jones archive has return predictability (e.g., Ke et al., 2019, He et al., 2025).

Our data from the Dow Jones archive are from 2000 to 2009, and we filter the articles

following the procedures in Ke et al. (2019). The number of articles remaining after each stage of article filtering is in appendix Table A1.

Daily return data are from CRSP and additional firm variables are from Compustat. SIC code groupings are taken from Kenneth French's website for the 49-industry portfolios.

4. Empirical analysis

Figure 1 provides a roadmap for our analyses. We first pass all texts through an LLM with a neutering prompt. The key elements of the prompt include the instruction to remove natural identifiers (e.g., names, dates, products, industries) but additionally instructs the LLM to use its best judgment in removing contextual clues to the entity's identity.

To validate the neutering procedure, we then feed the neutered text to a new instance of the original LLM and ask it to identify the firm, its ticker, its industry, and the year of the article. We refer to this as the *neutering test* in Figure 1. We also feed the neutered text to other LLMs and ask them to identify the subject firm.

After removing from the sample all articles in which the LLM can identify either the firm or the year, we then extract the sentiment from the raw and neutered text and evaluate how correlated the sentiment scores are - the *sentiment test*.

The final step uses the extracted sentiment to predict returns using the raw text and the neutered text. We compare the predictability for both — the *predictability test* — and where different, we estimate bounds on look-ahead bias.

4.1 Entity neutering

We perform our entity neutering process on our sample of Dow Jones articles by asking GPT-4o-mini (GPT) to remove any entity-identifying information from the articles. Specifically, for each article, we tell the LLM:

> *Your role is to ANONYMIZE all text that is provided by the user. After you have anonymized a text, NOBODY, not even an expert financial analyst should*

be able to read the text and know the identity of the company nor the industry the company operates in.

We give specific examples of what might be entity-identifying:

> *This includes the names of individuals, locations, industries, sectors, product names and types and generic product lines, services, times, years, dates and all numbers and percentages in the text including units. These should be replaced with name_x, location_x, industry_x, sector_x, product_x, product_type_x, product_line_x, service_x, time_x, year_x, date_x and number a, b, c, respectively.*

We also instruct the LLM to use its best judgment to mask entity-identifying information:

> *You should also ANONYMIZE any other information which one could use to identify the company or make an educated guess at its identity.*

The full prompt is in Appendix A.1.

4.2 A Neutering test: Can the LLM identify the entity from the neutered text?

To provide evidence for generalization neutrality (as defined in Section 2.3.1) we ask can an LLM identify the subject firm or year from the neutered text? To answer that question, we feed each neutered article back to the LLM and ask it. Specifically, we give it each neutered article, along with the following prompt:

> *You will receive a body of text which has been anonymized. You are omniscient. Use all your knowledge and the context to identify which company the text is about, as well as the year it was written. Make your best guess based on information and context if you are unsure.*

The full prompt is in Appendix A.2.

The results are presented in Table 1 and Figure 2. Among the 911,283 articles with sentiment, GPT is able to identify the subject firm in 82,753 of the cases, or 9.1% of the time and it is able to identify the year of the article correctly only 2.3% of the time. It is able to identify *either* the year or the name of the firm only 11.0% of the time. Given these low rates of identification, we conclude the neutering process is relatively successful.

We next explore the added value of neutering versus a simpler approach, such as masking only the firm's name, in Figure 2. When we only mask the firm's name, GPT is able to use the remaining contextual clues in the articles to identify the subject firm correctly 65.2% of the time, in line with the findings in Sarkar and Vafa (2024). In addition, it is able to correctly guess the year of the article 90.8% of the time. We then follow the de-identification approach of Glasserman and Lin (2023) using the Google Knowledge Graph (GKG), which also masks up to 20 words and products associated with the subject firm. Figure 2 shows that entity neutering strongly outperforms this approach as well. For example, GPT was able to correctly guess the subject firm of articles de-identified with GKG 47% of the time.

Figure A2 shows that the entity neutering approach is robust to substantially increasing article length — despite longer articles providing the LLM with more material from which to identify the entity. While the identification rate of either firm or article year rises above the 11% average as the number of characters in the article increases, it stabilizes and remains below 18% even as the number of characters exceeds 10,000.

4.2.1 Robustness to alternate LLMs

A natural concern is that a LLM model retains some memory of past interactions, even when they are run in different instances and on different days. This suggests that the LLM that performed the neutering should not be the one to test its validity. The final rows of Table 1 demonstrate that LLMs other than of GPT also display the same overall inability to identify the subject of a neutered article.

We feed the neutered articles to three different LLMs: Meta's Llama 3.2 class of models

with 1 and 3 billion parameters and Deepseek's R1 model with 1.5 billion parameters. Across all three LLMs we find low rates of entity identification. For example, Llama's more powerful model identifies the firm correctly 7.2% of the time. Likewise, Deepseek's R1 identifies the firm correctly 7.2% of the time. The success of our validation tests does not appear to depend on the LLM performing the testing.

4.3 Sentiment Similarity: How correlated is the raw vs. neutered sentiment?

While we have shown that neutering is an effective way of removing entity-identifying information from our news articles, there is a risk that it comes at the cost of information content, as we argue in Section 2.3.2.

In this section, we explore to what extent our entity neutering has "thrown out the baby with the bathwater." We begin with the subsample of articles for which GPT was unable to guess either the firm or year in Section 4.2. Thus, we take our sample of 811,501 original articles and 811,501 neutered articles and ask the LLM to classify the textual sentiment of each article for the purposes of return predictability. We explicitly mention future returns in the prompt because the purpose of the exercise is to consider a setting in which the LLM might engage in look-ahead bias. Specifically, we prompt GPT with:

> *You are an expert Financial Analyst. Your task is to read news articles and infer if the reported news is a good or bad signal for the future returns of the security being written about. Based on your answer, I will choose to buy or sell that security. After reading the news, provide a bull or bear signal as two numbers: the direction and the magnitude.*

The full prompt is in Appendix A.3.

To ascertain how much information content has been lost in the neutering process, we compare the textual sentiment of the original articles with that of the neutered articles. Among the 409,821 original articles with positive sentiment according to GPT, 91.4% of

these also have positive sentiment in the corresponding neutered article. Likewise, among the 322,482 original articles that GPT assigned negative sentiment, 88.8% of these are negative in the corresponding neutered article.

We also find similarity in the sentiment *magnitudes* assigned by GPT. Among the 409,821 articles that GPT assigns positive sentiment to both the original and neutered versions, the pair-wise rank correlation of their magnitude is 72.5%. Among negative sentiment articles, the rank correlation is 72.2%.

4.4 Return Predictability: How similar is raw vs. neutered return predictability?

Our final set of tests compares the return predictability of raw sentiment with the return predictability of the neutered sentiment.

We begin by presenting event studies in Figure 3. Sentiment from each article is measured on the publication date, day t, and and we plot abnormal returns using the approach in Daniel et al. (1997) from event days $t+1$ to $t+10$. In all panels, the sentiment extracted from the raw text is denoted by a solid line and the sentiment extracted from the neutered text by a dotted line.

Panel (a) plots the average event time abnormal returns from positive sentiment articles in blue and negative sentiment in red. The plots indicate return predictability from the textual sentiment of the news articles, consistent with prior work such as Tetlock et al. (2008). The return predictability lasts until event day $t+5$. Critically, the dotted lines are nearly identical to the solid lines, indicating that the return predictability from the neutered text is very similar to that of the raw text in both magnitude and trend. For example, positive sentiment extracted from the neutered text predicts next-day abnormal returns of 9.1 bps, while from the raw text this number is 9.9 bps, with the difference statistically indistinguishable from zero.

Panel (b) plots the difference between the positive sentiment abnormal returns and the

negative sentiment abnormal returns in event-time. The difference is in a solid black line for the raw articles and in a dotted line for the neutered articles. Again, we find very similar patterns. For example, on event days $t+1$, $t+2$ and $t+3$ the returns using the raw articles are 21.4 bps, 4.8 bps and 1.8 bps while the corresponding returns using the neutered articles are 20.3 bps, 5.0 bps and 1.7 bps.

Panels (c) and (d) mirror those of (a) and (b) among the sample of articles with *strong sentiment*. These are articles for which GPT assigns an above-median score when asked to estimate the sentiment magnitude. While the neutered sentiment still tracks the raw sentiment in these figures, we begin to see a difference in predictability coming from the raw versus neutered texts, especially on event day $t+1$. For example, in the Positive-Negative difference among strong sentiment articles (panel (d)), the day $t+1$ return using the raw articles is 33 bps, compared to 24.7 bps for the neutered articles.

Table 2 presents the regression analogue of Figure 3 via a panel regression (stock i, day t) using the sample of observations for which GPT cannot identify either the subject firm or year, and can be matched to DGTW returns.[6] The controls are analogous to those in Tetlock et al. (2008) and include the contemporaneous DGTW return, lagged DGTW returns, and the logs of book-to-market, market capitalization and share turnover. The dependent variable in columns (1) - (4) is stock $i's$ abnormal return on day $t+1$, while in columns (5) - (8) it is the abnormal return from day $t+2$ to $t+5$.[7] The odd columns measure sentiment using the *raw* text, while the even columns use the *neutered* text.

The regression results, despite including controls, are similar to those in the figure, with the sharpest difference in coefficients coming from those that measure strong sentiment. For example, when predicting next-day abnormal return, the coefficient on *Negative sentiment* from the raw text is -10.5 bps, compared to -10.8 bps for the neutered text. These are statistically indistinguishable using a Wald test. However, when comparing *Strong negative*

[6] Appendix Table A2 uses a more restrictive sample, for which the LLM cannot recognize the firm, year, ticker or industry of the article. Results are robust.

[7] We do not go further than $t+5$ because, as Appendix Figure A3 shows, there is no predictability past $t+5$.

sentiment in columns (3) and (4) the comparison is now -20.5 bps for the raw versus -14.1 bps for the neutered, with the difference significant at the 1% level. In fact, across all specifications, the only coefficients which are statistically distinguishable from each other at the 1% level when comparing raw versus neutered sentiment are those with strong sentiment.

Table 3 presents the results from a calendar-time trading strategy based on sentiment, which complements the event-time results in Figure 3 and the panel regressions in Table 2. The trading strategy goes long positive sentiment stocks and short negative sentiment stocks, and rebalances daily. The strategy returns are regressed on the Fama and French (2018) six-factor model, with the factor betas and strategy alpha displayed in the table.

Columns (1) and (2) present the results from the raw and neutered text, respectively. The columns indicate the strategy has no factor exposure but significant alpha: 26 bps per day for the raw text and 25 bps for the neutered text. The difference (column 3) is insignificant. Only when we condition on strong sentiment (i.e., long *Strong positive* sentiment and short *Strong negative* sentiment) do we see that the 37 bps for the raw text is significantly different than the 30 bps per day for the neutered text. In short, our results in calendar time are consistent with those in event time.

To the extent that there are differences in predictability coming from the raw and neutered texts, Section 2.4 shows that we can use these differences in order to bound the look-ahead bias coming from the predictability of the raw text. Motivated by Equation (6), we calculate these bounds by comparing the mean squared error improvement from the regressions of Table 2 when adding the sentiment variables for each pair of regressions (columns 1 and 2, 3 and 4, 5 and 6 and 7 and 8). For example, the improvement in mean squared error when adding the raw sentiment variables in column (1) is 64, but some of this improvement may be attributable to look-ahead bias. The improvement in mean squared error when adding the neutered sentiment variables in column (2) is 53, which suggests that at most (64 - 53) / 64 = 17% of the predictability for $t + 1$ DGTW returns from the raw sentiment is coming from look-ahead bias. Similarly, if we predict $t + 2$ to $t + 5$ DGTW returns (using columns

5 and 6), at most 14% is coming from look-ahead bias.

The bound calculations generally indicate that the greatest possibility for look-ahead bias occurs among the models with strong sentiment, as this is where the predictability from the raw sentiment differs most from that of the neutered sentiment. For example, when predicting $t+1$ DGTW returns using strong sentiment, we conclude that at most 38% is coming from look-ahead bias. When predicting $t+2$ to $t+5$ DGTW returns at most 32% is coming from look-ahead bias.

5. Conclusion

Large Language Models are a remarkable technology, and many researchers and practitioners are using them to extract content from text to forecast future outcomes. Because state-of-the-art LLMs are trained with recent data, a natural concern arises: look-ahead bias. In this paper, we propose a solution called *entity neutering* which uses the LLM itself to remove all contextual clues from text while maintaining as much semantic content as possible.

We demonstrate the effectiveness of entity neutering among 900,000 news articles by neutering each article and then confirming that the LLM cannot recognize the subject firm or time period for about 90% of the articles. In addition, when we compare the sentiment extracted from the raw text to the neutered text, we find them to be highly correlated with similar return predictability. This suggests that much of the information was preserved during the neutering process. Only when we condition on strong sentiment, do we see some significant differences in predictability between the raw and neutered sentiment. Finally, we show how one can use these differences to compute an upper bound on look-ahead bias.

If recent advances are any indication, LLMs will become more powerful over time. They will be trained on larger, recent datasets and they will have many more parameters. Almost certainly, this will amplify the risk of look-ahead bias. However, because entity neutering harnesses the LLM itself to neutralize the bias, its effectiveness should scale with the sophis-

tication of these models. Thus, as LLMs become more powerful, so too will the neutering process that we propose here.

References

Bybee, J Leland, ????, The ghost in the machine: Generating beliefs with large language models, 2023.

Bybee, Leland, Bryan Kelly, and Yinan Su, 2023, Narrative asset pricing: Interpretable systematic risk factors from news text, *Review of Financial Studies* 36, 4759–4787.

Chen, Yifei, Bryan T Kelly, and Dacheng Xiu, 2022, Expected returns and large language models, *Available at SSRN 4416687* .

Daniel, Kent, Mark Grinblatt, Sheridan Titman, and Russ Wermers, 1997, Measuring mutual fund performance with characteristic-based benchmarks, *Journal of Finance* 52, 1035–1058.

Eisfeldt, Andrea L, Gregor Schubert, and Miao Ben Zhang, 2023, Generative AI and firm values, National Bureau of Economic Research.

Fama, Eugene F, and Kenneth R French, 2018, Choosing factors, *Journal of Financial Economics* 128, 234–252.

Garcia, Diego, Xiaowen Hu, and Maximilian Rohrer, 2023, The colour of Finance words, *Journal of Financial Economics* 147, 525–549.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy, 2019, Text as data, *Journal of Economic Literature* 57, 535–574.

Glasserman, Paul, and Caden Lin, 2023, Assessing look-ahead bias in stock return predictions generated by GPT sentiment analysis, *arXiv preprint arXiv:2309.17322* .

He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu, 2025, Chronologically Consistent Large Language Models.

Hoberg, Gerard, and Asaf Manela, 2025, The natural language of finance, *Working paper* .

Jha, Manish, Jialin Qian, Michael Weber, and Baozhong Yang, 2024, ChatGPT and corporate policies, NBER working paper.

Ke, Zheng Tracy, Bryan T Kelly, and Dacheng Xiu, 2019, Predicting returns with text data, NBER working paper.

Kearney, Colm, and Sha Liu, 2014, Textual sentiment in finance: A survey of methods and models, *International Review of Financial Analysis* 33, 171–185.

Kelly, Bryan, Asaf Manela, and Alan Moreira, 2021, Text selection, *Journal of Business & Economic Statistics* 39, 859–879.

Lopez-Lira, Alejandro, 2023, Risk factors that matter: Textual analysis of risk disclosures for the cross-section of returns, Working paper.

Lopez-Lira, Alejandro, and Yuehua Tang, 2023, Can ChatGPT forecast stock price movements? Return predictability and large language models, arXiv preprint arXiv:2304.07619.

Loughran, Tim, and Bill McDonald, 2011, When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks, *Journal of Finance* 66, 35–65.

Ludwig, Jens, Sendhil Mullainathan, and Ashesh Rambachan, 2024, Large language models: An applied econometric framework, *arXiv preprint arXiv:2412.07031* .

Sarkar, Suproteem K., 2024, StoriesLM: A Family of Language Models With Sequentially-Expanding Pretraining Windows.

Sarkar, Suproteem K, and Keyon Vafa, 2024, Lookahead bias in pretrained language models, Working paper.

Tetlock, Paul C, 2007, Giving content to investor sentiment: The role of media in the stock market, *Journal of Finance* 62, 1139–1168.

Tetlock, Paul C, Maytal Saar-Tsechansky, and Sofus Macskassy, 2008, More than words: Quantifying language to measure firms' fundamentals, *Journal of Finance* 63, 1437–1467.

Figure 1: Procedure

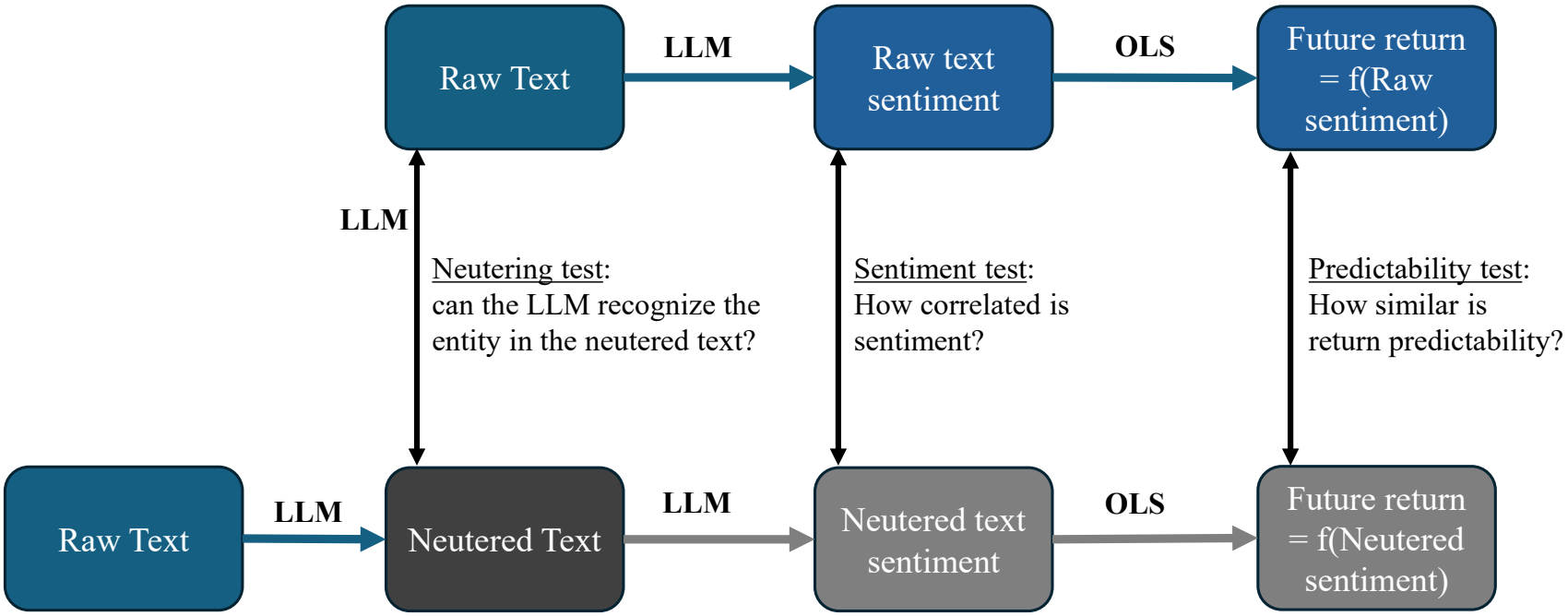
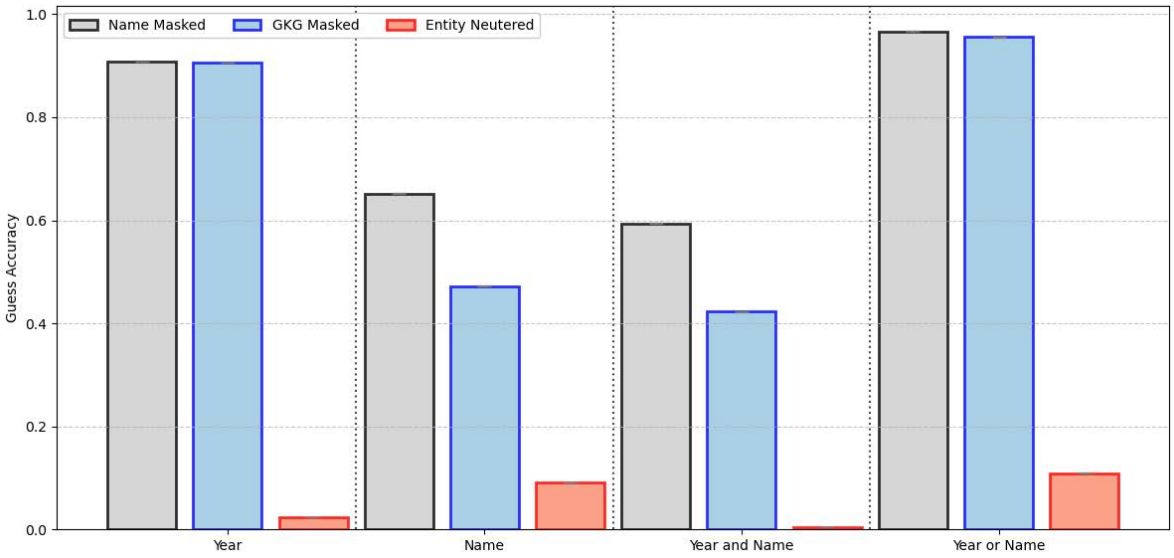
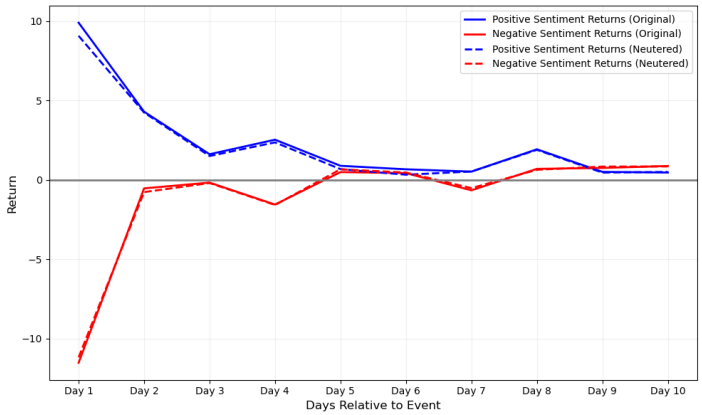


Figure 2: LLM guessing accuracy by de-identification method

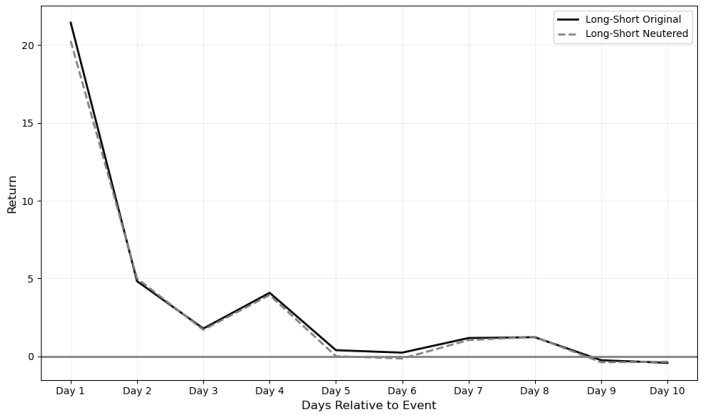


The columns labeled *Year* report the share of cases in which the LLM correctly guessed the year of the article. The *Name* columns correspond to correctly guessing the firm name. The columns labeled *Year and Name* correspond to the LLM correctly guessing both year and firm name, whereas *Year or Name* corresponds to the LLM correctly guessing either the year or the firm name. Each color bar represents a different method of de-identification. The leftmost (gray) bars mask all the firm name; the middle (blue) bars mask up to 20 firm-associated terms using Google Knowledge Graph (GKG) (following Glasserman and Lin, 2023); the rightmost (red) bars correspond to our entity-neutering procedure.

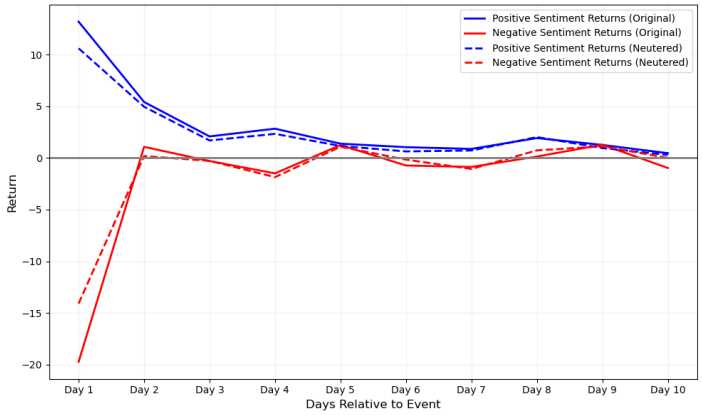
Figure 3: Event study returns by news sentiment



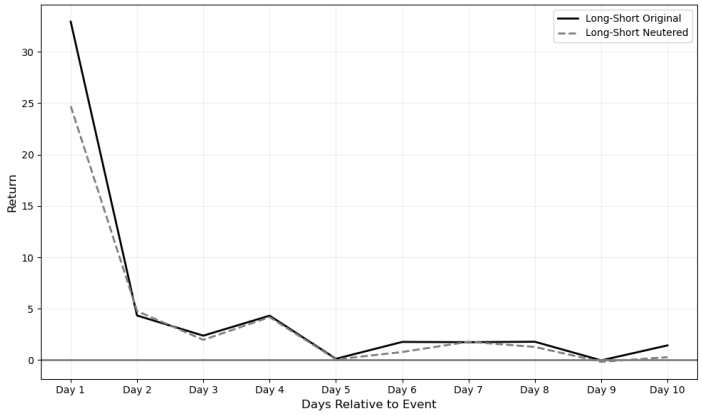
(a) Positive and negative



(b) Positive−negative



(c) Strong positive and strong negative



(d) Strong positive−strong negative

The figures plot average DGTW returns in event time. The blue lines (in the top half of the figure) in panels (a) and (c) indicate the return of articles with positive sentiment on day 0, while the red lines (in the bottom half of the figure) correspond to negative sentiment articles. The black lines in panels (b) and (d) are the difference between the two. Solid lines indicate sentiment measured with the raw text, dotted lines indicate sentiment measures with the neutered text. The top two panels include the sentiment of all articles, while the bottom two only include above-median (strong) sentiment.

Table 1: Firm identification rate by LLM

Firm	Model	# parameters	identification rate	# obs.
OpenAI	GPT-4o-mini	unknown	9.1%	911,283
Meta	Llama 3.2	3B	8.2%	21,100
DeepSeek	R1	1.5B	7.2%	5,000
Google	Gemma 3	4B	XXX%	XXX

The table presents the rate of correct guesses of firm names after being fed neutered articles for a variety of LLMs. The first row uses OpenAI's GPT-4o-mini model. Subsequent rows use other LLMs, with a random subsample of articles.

Table 2: Return predictability – raw vs. neutered text

	Dep. var.: $DGTW_{i,t+1}$				Dep. var.: $DGTW_{i,(t+2) \to (t+5)}$			
	(1) Raw	(2) Neutered	(3) Raw	(4) Neutered	(5) Raw	(6) Neutered	(7) Raw	(8) Neutered
Positive sentiment$_{i,t}$	7.18***	5.69***			7.31***	5.17***		
	[5.66]	[4.76]			[3.65]	[2.84]		
Negative sentiment$_{i,t}$	-10.48***	-10.80***			-13.96***	-14.94***		
	[-6.38]	[-5.95]			[-5.79]	[-6.89]		
Strong positive$_{i,t}$			10.14***	7.07***			11.20***	6.96***
			[6.81]	[5.62]			[4.96]	[3.61]
Weak positive$_{i,t}$			1.06	-0.49			-0.62	-2.85
			[0.79]	[-0.32]			[-0.28]	[-1.13]
Weak negative$_{i,t}$			-1.40	-3.16			-4.37*	-6.06**
			[-0.69]	[-1.22]			[-1.67]	[-2.16]
Strong negative$_{i,t}$			-20.49***	-14.11***			-24.54***	-18.78***
			[-9.62]	[-7.72]			[-5.93]	[-6.99]
$DGTW_{i,t}$	0.03**	0.03**	0.03**	0.03**	-0.06***	-0.06***	-0.06***	-0.06***
	[2.28]	[2.30]	[2.19]	[2.27]	[-3.36]	[-3.35]	[-3.41]	[-3.36]
$DGTW_{i,t-1}$	-0.04**	-0.04*	-0.04**	-0.04**	-0.12***	-0.12***	-0.12***	-0.12***
	[-1.96]	[-1.96]	[-1.99]	[-1.97]	[-2.77]	[-2.76]	[-2.78]	[-2.77]
$DGTW_{i,t-2}$	-0.07**	-0.07**	-0.07**	-0.07**	-0.08**	-0.08**	-0.08**	-0.08**
	[-2.53]	[-2.53]	[-2.55]	[-2.54]	[-2.26]	[-2.26]	[-2.27]	[-2.26]
$DGTW_{i,(t-21) \to (t-3)}$	-0.01**	-0.01**	-0.01**	-0.01**	-0.01*	-0.01*	-0.01*	-0.01*
	[-2.24]	[-2.23]	[-2.29]	[-2.25]	[-1.92]	[-1.91]	[-1.96]	[-1.93]
$DGTW_{i,(t-252) \to (t-22)}$	0.00	0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00
	[0.70]	[0.75]	[0.33]	[0.63]	[-1.26]	[-1.22]	[-1.45]	[-1.29]
Log($BM_{i,t}$)	0.40	0.31	0.84	0.51	-1.43	-1.54	-0.92	-1.30
	[0.28]	[0.22]	[0.60]	[0.36]	[-0.59]	[-0.64]	[-0.39]	[-0.54]
Log($ME_{i,t}$)	-1.31**	-1.28**	-1.11*	-1.18*	-1.33	-1.29	-1.13	-1.18
	[-2.01]	[-1.97]	[-1.67]	[-1.80]	[-1.31]	[-1.27]	[-1.10]	[-1.15]
Log($ST_{i,t}$)	1.10	1.06	1.69	1.22	8.32***	8.29***	8.92***	8.47***
	[0.57]	[0.55]	[0.85]	[0.63]	[2.91]	[2.91]	[3.02]	[2.94]
Constant	22.93***	23.03***	21.20***	21.93***	37.06***	37.43***	35.39***	36.15***
	[2.82]	[2.81]	[2.60]	[2.67]	[2.88]	[2.91]	[2.74]	[2.81]
# observations	834,804	834,804	834,804	834,804	834,789	834,789	834,789	834,789
# clusters (date)	2,219	2,219	2,219	2,219	2,219	2,219	2,219	2,219
Adj. R^2	0.011	0.011	0.011	0.011	0.012	0.012	0.012	0.012

Sentiment is extracted via LLM as described in section 4. Strong (weak) sentiment is an indicator for articles with sentiment magnitude $\geq$ ($<$) 0.7, the median sentiment. Controls are analogous to Tetlock et al. (2008). Coefficients are in basis points. t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Table 3: Trading strategy returns

	Dep. var.: Long-short$_{t+1}$					
	(1)	(2)	(3)	(4)	(5)	(6)
	Raw	Neutered	Difference	Raw & strong	Neutered & strong	Difference & strong
α	25.69***	24.72***	0.97	37.66***	29.69***	7.97***
	[9.87]	[8.98]	[1.38]	[10.43]	[9.61]	[6.97]
Market	-0.04*	-0.03	-0.01	-0.07*	-0.04	-0.03*
	[-1.65]	[-1.31]	[-0.68]	[-1.76]	[-1.24]	[-1.88]
HML	-0.10	-0.10	-0.00	-0.14	-0.13	-0.00
	[-1.51]	[-1.41]	[-0.20]	[-1.27]	[-1.45]	[-0.11]
SMB	-0.02	-0.03	0.01	-0.06	-0.05	-0.02
	[-0.28]	[-0.42]	[0.53]	[-0.69]	[-0.56]	[-0.63]
UMD	-0.01	-0.00	-0.01	0.04	0.01	0.02
	[-0.40]	[-0.09]	[-0.95]	[0.77]	[0.32]	[1.51]
RMW	-0.02	0.00	-0.02	-0.00	0.02	-0.03
	[-0.33]	[0.05]	[-1.11]	[-0.01]	[0.44]	[-0.97]
CMA	-0.02	-0.05	0.03	-0.07	-0.04	-0.04
	[-0.26]	[-0.56]	[1.22]	[-0.59]	[-0.34]	[-1.04]
# observations	2,219	2,219	2,219	2,219	2,219	2,219
Adj. R^2	0.002	0.001	0.001	0.005	0.003	0.007

Note: Trading strategy returns are constructed by equal weighting on day t and holding through day $t+1$. Strategy returns are regressed on the six factors in Fama and French (2018). In columns 1 and 2 we go long stocks with positive sentiment and short stocks with negative sentiment. In columns 3 and 4 we go long stocks with strong positive sentiment and short stocks with strong negative sentiment. Columns 3 and 6 report the differences between 1 and 2, and 4 and 5 respectively. All regressors are measured at $t+1$. Coefficients are in basis points. Robust t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

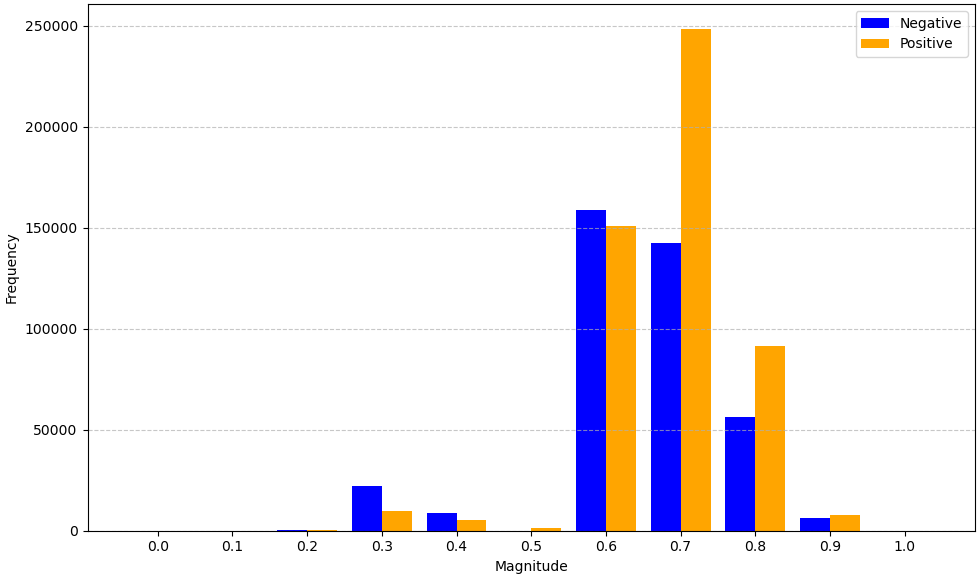
Internet Appendix:

Entity Neutering

by Joseph Engelberg, Asaf Manela, William Mullins & Luka Vulicevic[1]

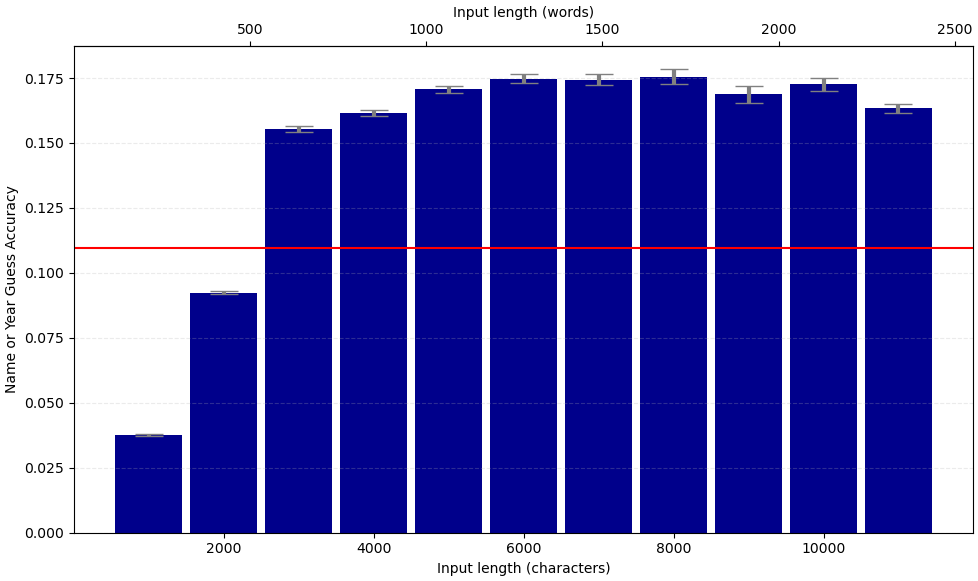
[1]jengelberg@ucsd.edu, amanela@wustl.edu, wmullins@ucsd.edu, lvulicevic@ucsd.edu.

Figure A1: Distribution of sentiment magnitude



The figure plots the distribution of the magnitude of sentiment generated by the LLM. Sentiment magnitude is on the horizontal axis and ranges from 0 to 1, where higher values represent stronger sentiment. For each horizontal-axis value, the left-hand bar (blue) represents negative sentiment, and the right hand bar (orange) represents positive sentiment.

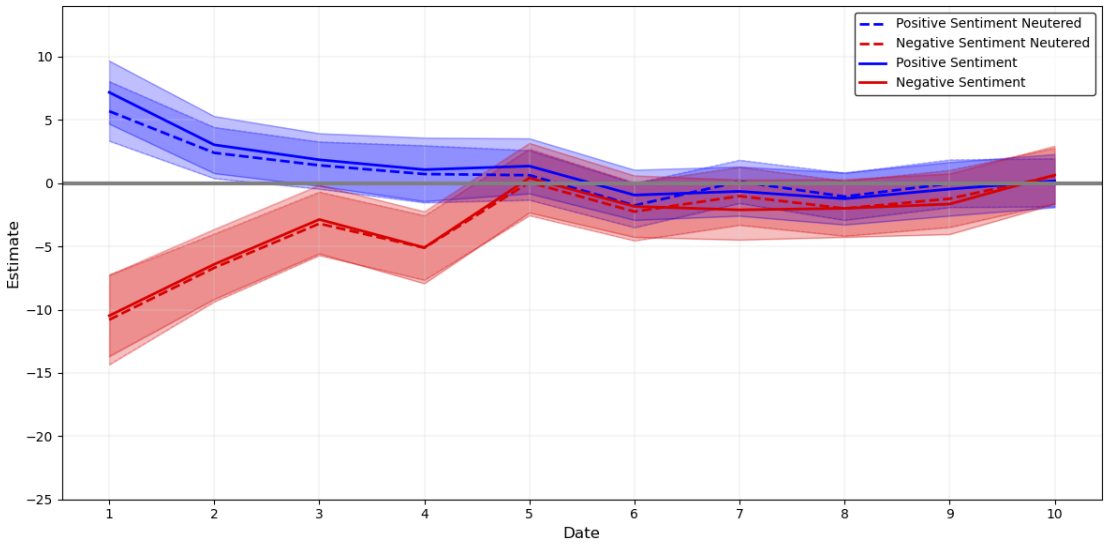
Figure A2: Firm or year identification rate by article length



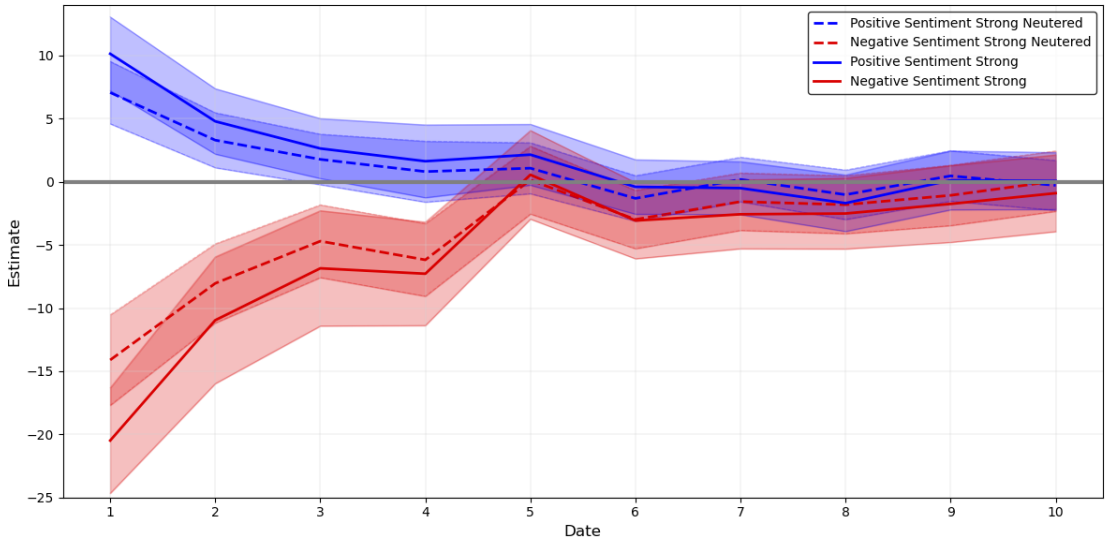
This figure plots the identification rate of firm or year as a function of article length (measured in characters). The vertical axis represents the share of all articles for which LLM was able to correctly identify the firm or year. Average accuracy is 11% and is plotted with a red horizontal line.

Figure A3: Regression coefficients in event time

Panel A: Full sample



Panel B: Strong sentiment only



This figure plots coefficients and 95% confidence intervals on positive and negative sentiment from a regression of DGTW returns mirroring the specification in Table 2, columns 1 and 2, for event days $t+1$ through $t+10$. The blue lines (in the top half of the figure) correspond to positive sentiment, while the red lines (in the bottom half) correspond to negative sentiment. Solid lines correspond to sentiment extracted from raw text; dotted lines correspond to sentiment extracted from neutered text.

Table A1: Sample filtering

Filter	**Sample size**	**Obs. removed**
Dow Jones Archive (2000-2009)	6,425,043	
Only one stock tagged	4,939,431	1,485,612
First article in chain	4,136,881	802,550
Body longer than 500 characters	2,096,540	2,040,341
Uniquely mapped to daily CRSP tickers	1,056,053	1,040,487
Sentiment can be extracted	911,283	144,770
Un-identifiable neutered articles	811,501	99,782

This table shows the number of remaining observations at each step of the data filtering process, which follows Ke et al. (2019).

Table A2: Return predictability – robustness

	Dep. var.: $DGTW_{i,t+1}$				Dep. var.: $DGTW_{i,(t+2) \to (t+5)}$			
	(1) Raw	(2) Neutered	(3) Raw	(4) Neutered	(5) Raw	(6) Neutered	(7) Raw	(8) Neutered
Positive sentiment$_{i,t}$	6.81***	5.39***			7.52***	5.35***		
	[5.25]	[4.50]			[3.66]	[2.85]		
Negative sentiment$_{i,t}$	-10.06***	-10.05***			-13.48***	-14.11***		
	[-5.75]	[-5.11]			[-5.72]	[-6.56]		
Strong positive$_{i,t}$			9.79***	6.70***			11.28***	7.21***
			[6.28]	[5.28]			[4.84]	[3.62]
Weak positive$_{i,t}$			0.92	-0.37			0.14	-2.84
			[0.67]	[-0.24]			[0.06]	[-1.08]
Weak negative$_{i,t}$			-1.23	-2.54			-3.90	-5.22*
			[-0.57]	[-0.93]			[-1.43]	[-1.77]
Strong negative$_{i,t}$			-20.59***	-13.49***			-24.90***	-18.17***
			[-9.30]	[-6.86]			[-6.03]	[-6.89]
$DGTW_{i,t}$	0.03**	0.03**	0.03*	0.03**	-0.06***	-0.06***	-0.06***	-0.06***
	[2.03]	[2.06]	[1.94]	[2.03]	[-3.21]	[-3.20]	[-3.26]	[-3.21]
$DGTW_{i,t-1}$	-0.04*	-0.04*	-0.04*	-0.04*	-0.13***	-0.13***	-0.13***	-0.13***
	[-1.85]	[-1.84]	[-1.87]	[-1.85]	[-2.82]	[-2.82]	[-2.83]	[-2.82]
$DGTW_{i,t-2}$	-0.07**	-0.07**	-0.07**	-0.07**	-0.08**	-0.08**	-0.08**	-0.08**
	[-2.34]	[-2.34]	[-2.35]	[-2.34]	[-2.23]	[-2.23]	[-2.24]	[-2.23]
$DGTW_{i,(t-21) \to (t-3)}$	-0.01**	-0.01**	-0.01**	-0.01**	-0.01*	-0.01*	-0.01*	-0.01*
	[-2.09]	[-2.09]	[-2.14]	[-2.10]	[-1.81]	[-1.81]	[-1.86]	[-1.82]
$DGTW_{i,(t-252) \to (t-22)}$	0.00	0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00
	[0.45]	[0.49]	[0.09]	[0.38]	[-1.22]	[-1.19]	[-1.40]	[-1.25]
Log($BM_{i,t}$)	-0.17	-0.25	0.25	-0.07	-1.73	-1.84	-1.25	-1.60
	[-0.12]	[-0.18]	[0.18]	[-0.05]	[-0.75]	[-0.80]	[-0.55]	[-0.70]
Log($ME_{i,t}$)	-1.48**	-1.46**	-1.28*	-1.36**	-1.69	-1.66	-1.48	-1.54
	[-2.20]	[-2.16]	[-1.88]	[-2.01]	[-1.63]	[-1.60]	[-1.42]	[-1.48]
Log($ST_{i,t}$)	0.56	0.52	1.16	0.67	8.59***	8.54***	9.21***	8.71***
	[0.28]	[0.26]	[0.56]	[0.33]	[3.00]	[2.99]	[3.10]	[3.02]
Constant	24.22***	24.18***	22.54***	23.10***	42.71***	42.98***	40.97***	41.68***
	[2.90]	[2.89]	[2.70]	[2.76]	[3.25]	[3.27]	[3.12]	[3.17]
# observations	748,648	748,648	748,648	748,648	748,633	748,633	748,633	748,633
# clusters (date)	2,219	2,219	2,219	2,219	2,219	2,219	2,219	2,219
Adj. R^2	0.011	0.011	0.011	0.011	0.013	0.013	0.013	0.013

This is a robustness test for Table 2 in which we drop all articles for which at least one of the following are correctly guessed by the LLM from the neutered text: 1) firm name, 2) year, 3) ticker, 4) firm industry. Coefficients are in basis points. t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

A. Prompts

A.1 Neutering prompt

We use the following LLM prompt to neuter the original text of each article:

> *Your role is to ANONYMIZE all text that is provided by the user. After you have anonymized a text, NOBODY, not even an expert financial analyst should be able to read the text and know the identity of the company nor the industry the company operates in. For example, if the text is: The country's largest phone producer Apple had great phone related earnings but Google did not in 2024 likely because of Apple's slogan Think Different, then you should ANONYMIZE it to: The country's largest product_type_1 producer Company_1 had great product_type_1 related earnings but Company_2 did not in time_1 likely because of Company_1's slogan slogan_1. You should also ANONYMIZE any other information which one could use to identify the company or make an educated guess at its identity. Stock tickers are identifiers and are usually four capitalized letters or less (consider TIK as a stand-in for an arbitrary ticker) and are sometimes referenced in the text in the following formats; SYMBOL:TIK, $TIK, >TIK, $ TIK, SYMBOL TIK, SYMBOL: TIK, > TIK. Make sure you censor TIK to ticker_x, and any other identifiers related to companies. This includes the names of individuals, locations, industries, sectors, product names and types and generic product lines, services, times, years, dates and all numbers and percentages in the text including units. These should be replaced with name_x, location_x, industry_x, sector_x, product_x, product_type_x, product_line_x, service_x, time_x, year_x, date_x and number a, b, c, respectively. Also replace any website or internet links with link_x. You should never just delete an identifier; instead, always replace it with an anonymous analog. After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.*

A.2 Neutering test prompt

After neutering a text, we wish to examine whether the LLM can identify the entity (firm) and time (year). We use the following LLM prompt.

> *You will receive a body of text which has been anonymized. You are omniscient. Use all your knowledge and the context to identify which company the text is about, as well as the year it was written. Make your best guess based on information and context if you are unsure. Please only provide the ticker of the company you have identified. Provide your estimate exactly in the following format, with no other text at all (TIK is your estimate of the ticker, Y is your estimate of the year): **Company Estimate: TIK**,**Year Estimate: Y***

A.3 Sentiment prompt

To measure sentiment, we ask the LLM to provide (1) the direction (positive/negative) of the sentiment in the article, and (2) the magnitude or intensity of the sentiment $\in (0, 1)$. When prompted to extract a number in this manner, the LLM generates the value in a structured sequence: first '0' with probability approaching 1, then '.' with probability approaching 1, followed by the first decimal place, which provides the magnitude of the estimate. Constraining the LLM's output in this manner allows for precise control over tokenization. For example, when prompted to generate a magnitude from 1 to 5 without restrictions, the LLM may produce ambiguous tokens such as ' 4'. This ambiguity arises because the assigned probability of ' 4' conflates the likelihood of the magnitude being 4 with the probability of including a preceding space. Within the LLM's state space, tokens like '4', ' 4', '4 ', and similar variations coexist, making direct inference from log probabilities challenging, if not impossible. We use the following LLM prompt:

> *You are an expert Financial Analyst. Your task is to read news articles and infer if the reported news is a good or bad signal for the future returns of the*

*security being written about. Based on your answer, I will choose to buy or sell that security. After reading the news, provide a bull or bear signal as two numbers: the direction and the magnitude. For direction, output only 0,1, or NA, where 0 is bearish, 1 is bullish, and NA means there is no information relevant to security prices. For magnitude, output a number ranging from 0 to 1 ONLY, where numbers near 0 indicate slightly bearish (if the direction is 0) or slightly bullish (if the direction is 1), and numbers near 1 indicate highly bearish (if the direction is 0) or highly bullish (if the direction is 1). Provide your output exactly in the following format, with no other text at all: **Direction Estimate: DIRECTION**,**Magnitude Estimate: MAGNITUDE**"*