Entity Neutering

Joseph Engelberg
UC San Diego

Asaf Manela
Wash. U.

William Mullins
UC San Diego

Luka Vulicevic
*UC San Diego**

September 4, 2025

Method updated & improved. Most recent version on GitHub

Abstract

Cutting-edge LLMs are trained on recent data, creating a concern about look-ahead bias. We propose a solution called *entity neutering*: using LLMs to find and remove all identifying information from text. We use an agent-based technique called adversarial anonymization, whereby an LLM agent iterates (i) masking identifying words and (ii) paraphrasing the text until the subject firm is unidentifiable. In a sample of 250,000 financial news articles we verify that, after neutering, ChatGPT and other LLMs identify the subject firm at the rate of random chance. Among the unidentified articles, the sentiment extracted from the raw text and the neutered text agree more than 95% 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, Agentic, Look-ahead Bias, Textual Analysis

*Contact: jengelberg@ucsd.edu, amanela@wustl.edu, wmullins@ucsd.edu, lvulicevic@ucsd.edu. We thank David Hirshleifer, Paul Tetlock, Gerard Hoberg, Suproteem Sarkar, and Ashesh Rambachan for valuable comments. We would also like to thank seminar and conference participants at LSE, CU Boulder, Rotman, NBER SI Asset Pricing, and the Inaugural Finance Research Revolution Conference. Finally, we are also grateful to the Rady Research Fund for valuable funding, 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, 2023, Jha et al., 2024). For example, if we ask GPT-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-5 is trained on data through September 2024. If, as is likely, GPT-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. In addition, most frontier LLMs 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, *entity neutering*, to address look-ahead bias when using LLMs to extract data from text. 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 consists of more than simply removing a firm's name from the text. As an example, consider the following Dow Jones Newswire story about Coca Cola (KO) on July 10, 2014:

[1]See for example, Tetlock, Saar-Tsechansky, and Macskassy (2008), Fang and Peress (2009), Loughran and McDonald (2011), Gurun and Butler (2012), 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
Coca-Cola (KO) hasn't been a popular stock of late because of weak **soda** sales (volume logged its first quarterly drop in **15 years** in **1Q**). But **RBC** says forthcoming **2Q** results could be the strongest in a while amid a **later Easter** versus **2013**, a **World Cup** boost and the **"Share A Coke"** campaign in which consumers personalize **Coke cans** with names. **RBC** is forecasting **4.4% volume growth**, including **3%** in the **US**, up from **2%** globally in **1Q** and **1%** in **4Q**.	*Entity_A (Ticker_A)* hasn't been a popular stock of late because of weak **soda** sales (volume logged its first quarterly drop in **15 years** in **1Q**). But **RBC** says forthcoming **2Q** results could be the strongest in a while amid a **later Easter** versus **2013**, a **World Cup** boost and the **"Share A** *Entity_A*" campaign in which consumers personalize *Entity_A* **cans** with names. **RBC** is forecasting **4.4% volume growth**, including **3%** in the **US**, up from **2%** globally in **1Q** and **1%** in **4Q**.	*Entity_A's* most recent performance has failed to gain momentum due to soft demand for *sku_alpha*, having their first fall in demand in *a_years*. Nevertheless, *Entity_B* indicates the upcoming *period_z* could surpass the *previous_time_performance*, helped by an *annual_event*, a *marketing_promo* uptick, and a program allowing users to customize *sku_beta* with a *tag_id*. *Entity_B* expects *x_volume_growth*, including pockets of growth across *region_x* and *region_y*.

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 "*Entity_A*" is not enough to assuage concerns about look-ahead bias because the LLM can identify Coca Cola from the other informational clues, like the fact that it sells soda, will be impacted by the World Cup, and is covered by RBC. Even if this additional information is masked, the LLM may recognize the article because it has memorized certain token sequences from its training data (e.g., Carlini et al., 2021,

2023). The third column not only masks identifying information, but also *paraphrases* the article, so that the LLM cannot use memorized token sequences or phrases to identify the entity. Importantly, the entity-neutered version preserves the article's generally positive sentiment about upcoming events having a positive impact on Coca Cola sales, but without using words which would identify Coca Cola as the subject of the article.

We perform this entity neutering procedure on a random sample of 255,328 news articles from the Dow Jones Archive between 2000 and 2022. What distinguishes entity neutering from prior work (Glasserman and Lin, 2023, Sarkar and Vafa, 2024) is that we task the LLM to anonymize the article so that the subject firm is unidentifiable.[2] It does so in an iterative process - masking key words and paraphrasing the text at each iteration - until a fresh instance of the LLM cannot identify the subject firm. In the language of computer science, the LLM acts as an agent (Wang et al., 2024), using tools to achieve a well-defined goal. Because the process by which it reaches that goal requires an independent LLM to check whether anonymization is achieved after each iteration, entity neutering is also analogous to the *adversarial anonymization* approach in computer science (e.g., Staab et al., 2024).

Using this agentic approach, GPT-5 nano ('ChatGPT') effectively neuters the subject Dow Jones articles so that, after a maximum of 9 iterations, ChatGPT can only recognize the subject firm 0.09% of the time.

We confirm that entity neutering is effective by resubmitting the neutered articles to two other leading LLMs, Llama 3 and Gemma 3. All of the LLMs struggle to correctly guess the subject firm in neutered articles. Compared to ChatGPT's guessing rate of 0.09%, Llama and Gemma guess correctly around 0.05% of the time. The ability of the LLMs to guess the date of publication is even lower. For example, ChatGPT is able to guess the date within a two week window only 0.002% of the time.

These rates of identification are similar to random guessing. For example, if an LLM simply guessed the most popular firm in our sample (Microsoft), it would have been correct

[2]We use GPT-5 nano via the API, which does not have access to the internet.

0.15% of the time. If it had chosen a firm randomly from our distribution of articles, it would have been correct 0.1% of the time. This is similar to the 0.09% we find as the identification rate after entity neutering.[3]

To recap, among the 255,328 news articles, ChatGPT is only able to identify the firm or the date in 243 articles after neutering. We drop these "identifiable" articles from our sample and are left with 255,085 "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 ChatGPT to identify whether each article constitutes a positive or negative signal for the future returns of the subject firm. We do this for both the raw and neutered articles. The sentiment of the neutered text matches the sentiment of the raw text 95% of the time. 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. We extract both a binary measure of sentiment (positive or negative) as well as a magnitude. If the return predictability coming from the raw text is powered by look-ahead bias, then we would expect to find a sharp fall 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 ability of the neutered text to predict returns is similar to

[3]While we use our distribution of articles to calculate a random guessing benchmark, a more precise benchmark would be the distribution of articles that the LLM was trained on, but as Ludwig et al. (2024) notes, this distribution is unknown. However, when we ask ChatGPT for its top 5 guesses for the subject firm in an unspecified Dow Jones Newswire article it provided the following firms: Apple, Microsoft, Tesla, Google/Alphabet, and Amazon. In our sample, these occur with the following frequencies: 0.14%, 0.15%, 0.15%, 0.13%, and 0.13% respectively.

that of the raw text. For example, stocks with positive sentiment on a given day – when this sentiment is extracted from the *neutered* text – have an average next-day abnormal return of 19 basis points (bps), while the corresponding return for stocks with positive sentiment measured from the *raw* text is 19 bps (Figure 6), 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 the magnitude of sentiment do we see differences between the predicted returns from the raw and neutered texts. For example, positive articles with *Strong sentiment* (i.e., sentiment magnitude greater than one standard deviation above the mean) predict next day abnormal returns of 46 bps for the raw text and 33 bps for the neutered text. The corresponding figures among strongly negative articles are -57 and -35. 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 -66 bps when using the raw text, but -39 bps when using the neutered text. The coefficient on strong positive sentiment using the raw text is 43 bps, and 32 bps on the neutered text. This wedge generally holds even on event days after $t+1$.

We also find similar results in a calendar-time trading strategy which goes long positive-sentiment articles and short negative-sentiment articles. We regress the trading strategy returns on the Fama and French (2018) six factors, finding a positive and significant alpha and no significant factor loadings. This last finding suggests that look-ahead bias is not entering indirectly, via the LLMs' ability to identify neutered firms' factor-based characteristics.

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, 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 25% when predicting next-day returns with binary sentiment, and 43% when predicting returns with strong and weak sentiment.

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 LLM look-ahead bias. First, we recommend our entity neutering approach (i.e., masking, paraphrasing and iterating) so that the subject entity cannot be identified by a separate LLM. As we show, if entity neutering is successful, the identification rate for the LLM should be small. Secondly, we recommend dropping the observations for which the LLM correctly identifies the entity, and performing the analysis using both the neutered and the raw text among the unidentifiable observations. 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 developed here to obtain an upper bound on the contribution of look-ahead bias to predictability.

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 these 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 a concern (e.g., 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, which can introduce look-ahead bias (Glasserman and Lin, 2023, Sarkar and Vafa, 2024, Levy, 2024, Ludwig et al., 2024). 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* (He, Lv, Manela, and Wu, 2025).[4] While some papers have implemented this approach by applying LLMs exclusively to the limited 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, 2023, Jha, Qian, Weber, and Yang, 2024).

The second approach to the look-ahead bias problem - rather than switching out a cutting-edge LLM for a chronologically consistent *model*, is to *pre-process the text* so that the LLM cannot identify the entity.

The second approach to the look-ahead bias problem is to pre-process the text so that the LLM cannot identify the entity. That is, while the first approach addresses the problem by *changing the model*, the second approach addresses the problem by *changing the text*. The advantage of the second approach is that researchers can continue to use frontier LLMs (like ChatGPT and Gemini) rather than the less-powerful ones under the first approach. However,

[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.

thus far, the papers which have used this second approach have found little success. For example, Sarkar and Vafa (2024) masks firm names, and Glasserman and Lin (2023) masks names and 20 related terms. When we apply the Sarkar and Vafa (2024) approach to our data we find identification rates of 62% (similar to what they find); applying the Glasserman and Lin (2023) approach leads to identification rates of 45%. Our innovation is to task the LLM itself to iteratively use masking and paraphrasing to de-identify the text. To the best of our knowledge, entity neutering is the first approach which pre-processes the text in such a way that frontier LLMs can identify the subject firm at chance levels, allowing researchers to use these LLMs while also addressing the look-ahead bias problem. 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 main dataset is a random sample of 255,328 texts from the Dow Jones archive

between 2000 and 2022, and we filter the articles following the procedures in Ke, Kelly, and Xiu (2019).[6] The number of articles remaining after each filtering stage is reported in Appendix Table A1.

4. Empirical analysis

Figure 2 provides a roadmap for our analyses. We perform entity neutering in an iterative process - masking key words and paraphrasing the text at each iteration - until a fresh instance of the LLM cannot identify the subject firm. We describe this process in greater detail below.

We then drop from the sample all articles in which the LLM can identify either the firm or the date (within a window of seven days before and seven days after the publication date of the article) after nine iterations. We then extract the sentiment from the raw and neutered text and evaluate how correlated the sentiment scores are – which we call the *sentiment test* – as illustrated in Figure 2.

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 using GPT-5 nano ('ChatGPT'). Figure 1 is a flowchart that illustrates the entity neutering process, which extends the adversarial anonymization approach from computer science (Staab et al., 2024). The first step is masking. Specifically, for each article, we prompt the LLM with the following:

Your role is to ANONYMIZE all text that is provided by the user. The text you

[6] We take a random sample of approximately 10% of all texts in the archive (after filtering), rather than process the entire archive because ChatGPT-5 nano API calls cost approximately $1,200 per 100,000 articles.

need to anonymize will be about firm {NAME}, its ticker, {TICKER}, and its industry, {INDUSTRY}. Any instance of the firm, their products, the industry or industry related areas they are involved in should be anonymized. Absolutely no mention of key words related to that industry can be left in the text. After you have anonymized a text, NOBODY, not even an expert financial analyst should be able to use the text to identify the company or the industry the company operates in.

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

Also ANONYMIZE any other identifiers related to companies including 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.

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 to make an educated guess about its identity.

The full prompt is in Appendix B.1.

The next step is paraphrasing. Here the goal is to replace the tokens to prevent identification from LLM memorization while maintaining information content. For each article, we prompt the LLM with the following objective:

You are to paraphrase text ensuring that its core information content is unchanged, such that someone who has memorized the text could not recognize it. Make sure to replace all unique or identifying words and phrases. Also change

the sentence structure so that the resulting text is completely different from the original. This includes changing quotes and other highly recognizable structures in the text. The text provided will be about firm {NAME}, its ticker, {TICKER}, and its industry, {INDUSTRY}. ... Anonymity is the prime objective, leave zero hints for someone who has memorized the text! Ensure that the paraphrased text has approximately the same number of words as the original text.

The full prompt is in Appendix B.1.

After masking and paraphrasing, a new instance of the LLM checks to see if the article is anonymized. Specifically, we give it the masked and paraphrased article, along with the following prompt:

You will receive a body of text which has been anonymized. You are omniscient. Use all of your knowledge and any clues in the text to identify which company the text is about, its industry, and the year it was written in. Make your best guess if you are unsure.

The full prompt is in Appendix B.5.

If the LLM is able to identify the firm after this step, the entire process continues to iterate and stops only when the LLM is unable to identify the subject firm or after nine iterations, whichever comes first. The specific prompts for re-masking and re-paraphrasing are given in Appendix B.5.

4.2 Identification Rates after Entity Neutering

Identification rates after entity neutering are presented in Figure 3. Among the 255,328 articles in our sample, ChatGPT is able to identify the subject firm in 241 of the cases, or 0.09% of the time and it is able to identify the date of the article correctly only 0.003% of the time. It is able to identify *either* the date or the name of the firm only 0.1% of the time. These identification rates are similar to those of random guessing. For example, if the LLM

guessed the firm from a randomly chosen article in our sample, it would be correct 0.1% of the time. Given our rates of identification are low and similar to that of random chance, we conclude the neutering process is successful.[7]

In Appendix Table A2, we explore whether the identification rate is related to firm characteristics. For example, larger firms might be identified more often because they are more likely to appear in the training dataset. This is precisely what we find: large firms (those in the top quartile) are guessed correctly 0.22% of the time while small firms (bottom quartile) are guessed correctly only 0.02% of the time. We also find a higher guessing probability for growth stocks compared to value stocks (0.15% vs. 0.04%).

We next explore the added value of neutering versus a simpler approach, such as masking only the firm's name, in Figure 3. When we only mask the firm's name, ChatGPT is able to use the remaining contextual clues in the articles to identify the subject firm correctly 62.2% of the time, in line with the findings in Sarkar and Vafa (2024). In addition, it is able to correctly guess the date of the article 92.3% of the time. We also 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 3 shows that entity neutering strongly outperforms this approach as well. For example, ChatGPT was able to correctly guess the subject firm of articles de-identified with GKG 44.8% of the time.

Figure 4 displays the guessing accuracy of firm name and date after each iteration. It shows that most of the work of entity neutering is done by the first iteration, after which the LLM guesses the firm correctly in only 4.4% of the cases. The guessing rate decreases exponentially, reaching 0.09% after nine iterations. This rate of identification is similar to random guessing. For example, if an LLM simply guessed the firm that appears most

[7]Figure A3 shows that the entity neutering approach is robust to substantially increasing news 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 date rises above the 0.10% average as the number of characters in the article increases, it stabilizes and remains below 0.25% even as the number of words exceeds 1,000.

frequently in our sample (Microsoft), it would have been correct 0.15% of the time. If it had chosen a firm randomly from our distribution of articles, it would have been correct 0.1% of the time.

While we use our distribution of articles to calculate a random guessing benchmark, a more precise benchmark would be the distribution of articles that the LLM was trained on, but as Ludwig et al. (2024) notes, this distribution is unknown. However, when we ask ChatGPT for its top 5 guesses for the subject firm in an unspecified Dow Jones Newswire article it provided the following firms: Apple, Microsoft, Tesla, Google/Alphabet and Amazon. In our sample, these occur with the following frequencies: 0.14%, 0.15%, 0.15%, 0.13%, and 0.13%, respectively.

The guessing accuracy of the date of each article, which we score as correct if the guess is within 7 days on either side of the true date, is extremely low after the first iteration – 0.13% – falling to 0.04% after the second, and to 0.003% by the ninth.

If recent text makes up a disproportionate share of GPT's training corpus, it may be guessing the firms and dates that appear more frequently in recent texts. To explore this possibility, we plot the frequency of correct guessing by year in Figure 5. For firm names, there is no discernible time-trend: overall, the LLM guesses correctly in about 0.10% of cases most years. For date guessing, the rate is close to zero throughout.

4.2.1 Robustness to alternate LLMs

A natural concern is that an LLM may retain some memory of past interactions, even when they are run in different instances and on different days. If this is a concern, then the LLM that performed the neutering should not be the one to test its validity.

We feed the neutered articles to two different LLMs: Meta's Llama 3.2 with 3 billion parameters and Google's Gemma 3 model with 4 billion parameters. Across all LLMs we find low rates of entity identification. For example, Llama identifies the firm or the date correctly 0.04% of the time. Likewise, Google's Gemma 3 identifies the firm or the date

correctly 0.06% 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 versus 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 whether our entity neutering has destroyed the information in the text, i.e., whether it has "thrown out the baby with the bathwater." We begin with the 255,085 articles for which ChatGPT was unable to guess either the firm or date in Section 4.2. Thus, we then take the raw and neutered versions of each of the 255,085 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 ChatGPT 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 B.6.

To ascertain how much information content has been lost in the neutering process, we compare the textual sentiment of the raw articles with that of the neutered articles. Overall, when sentiment is detected in both the raw and neutered versions of the same article, they share the same sentiment 96% of the time when positive and 97% when negative. However, some information is lost in the neutering process – potentially a combination look-ahead bias

and direct information loss. Specifically, 21% of raw articles with either positive or negative sentiment have no detected sentiment in the neutered text.

4.4 Return Predictability: How similar is raw versus 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 6. Sentiment from each article is measured on the publication date, day t, 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 returns for articles with sentiment extracted from the raw text is denoted by a solid line and by a dotted line for sentiment extracted from the neutered text.

The plots indicate return predictability from the 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, the raw text predicts abnormal returns of 19 bps and -22 bps for positive and negative articles, respectively, on day $t+1$; for the neutered text, the corresponding numbers are 19 bps and -20 bps.

Panel (b) plots the difference between the positive sentiment abnormal returns and the negative sentiment abnormal returns in event-time. The difference is shown as a solid black line for the raw articles and as a dashed 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 41 bps, 6 bps and 5 bps while the corresponding returns using the neutered articles are 38 bps, 5 bps and 5 bps. In Appendix Figure A1 we show that the sentiment from either the raw or neutered text outperforms that of leading dictionary-based approaches (Loughran and McDonald, 2011, Garcia et al., 2023) by more than three times, highlighting the advantage of using frontier LLMs to extract sentiment from text.

Panels (c) and (d) of Figure 6 mirror those of (a) and (b) among the set of articles with *strong sentiment*. These are articles for which ChatGPT assigns a sentiment score at least one standard deviation above the mean. 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 103 bps, compared to 68 bps for the neutered articles.

Table 2 presents the regression analogue of Figure 6 via a panel regression (stock i, day t) using the sample of observations for which ChatGPT cannot identify either the subject firm or date, and can also be matched to DGTW returns (see Appendix Table A1). The controls are analogous to those in Tetlock et al. (2008) and include the contemporaneous DGTW return, lagged DGTW returns, and the natural logarithms of book-to-market, market capitalization and share turnover. The dependent variable in columns (1) through (4) is stock $i's$ DGTW return on day $t+1$, while in columns (5) to (8) it is the DGTW return from day $t+2$ to $t+5$.[8] The odd columns measure sentiment in the *raw* text, while the even columns use the *neutered* text.

The regression results in Table 2, despite including controls, are similar to those in Figure 6, with the sharpest difference in coefficients between neutered and raw articles coming from those with either strong positive or strong negative sentiment. For example, when predicting next-day abnormal return, the coefficient on *Negative sentiment* from the raw text is -28 bps and from the neutered text is -23 bps. However, when comparing *Strong negative* sentiment in columns (3) and (4) the comparison becomes -66 bps for the raw versus -39 bps for the neutered.

Table 3 presents the results from a calendar-time trading strategy based on sentiment, which complements the event-time results in Figure 6 and the panel regressions in Table 2. The trading strategy goes long positive sentiment stocks and short negative sentiment stocks,

[8]We do not go further than $t+5$ because, consistent with Figure 6, there is no predictability past $t+5$.

and rebalances daily. The strategy returns are regressed on the Fama and French (2018) six factors, 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 that the strategy has significant alpha: 48 bps per day for the raw text and 43 bps for the neutered text. The difference (column 3) is statistically significant at the 5% level. When we condition on strong sentiment (i.e., long *Strong positive* sentiment and short *Strong negative* sentiment), we see much bigger differences: 70 bps for the raw text versus 50 bps for the neutered text, with the 20 bps difference significant at the 1% level. Overall, our results in calendar time are consistent with those in event time.

Moreover, Table 3 indicates that the trading strategy has no exposure to any of the six Fama and French (2018) factors. This suggests that look-ahead bias is not entering indirectly, via the LLMs' ability to identify neutered firms' factor-based characteristics.

4.4.1 Bounding the look-ahead bias

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 374, 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 281, which suggests that at most (374 - 281) / 374 = 25% of the predictability for $t+1$ DGTW returns from the raw sentiment is coming from look-ahead bias.[9] Similarly, if we predict $t+1$ DGTW returns conditioning on

[9]The bound is calculated as the MSE improvement from using the raw sentiment .. using the neutered sentiment, scaled by the MSE improvement from using the raw sentiment versus no sentiment. That is, $Bound = (MSE_{raw\ sentiment} - MSE_{neutered\ sent.})\ /\ (MSE_{raw\ sent.} - MSE_{no\ sent.})$.

sentiment strength (using columns 3 and 4), at most 43% is coming from look-ahead bias.

We can also calculate bounds for the $t+2$ to $t+5$ period by comparing columns 5 and 6 as well as 7 and 8. Both bound calculations indicate that at most 39% of the return predictability for this period is coming from look ahead bias.

4.4.2 Guidance for researchers

The analysis in this article provides a recipe for dealing with concerns about LLMs' look-ahead bias. First, we recommend our entity neutering approach (masking, paraphrasing and iterating) so that the subject entity cannot be identified by a separate LLM. The prompts for this approach can be tailored to the specific research setting. As we show, if entity neutering is successful, the LLM should be able to identify the subject entity in only a small share of texts. Second, we recommend dropping observations for which the LLM identifies the entity, and performing the analysis using both the neutered and the raw text among the unidentifiable observations.

If the researcher's conclusions are generally unchanged when using the neutered text relative to the conclusions from the raw text, this is evidence that their results are not driven by look-ahead bias. If the conclusions differ, we recommend calculating the bounds suggested here, as an upper bound on the contribution of look-ahead bias to predictability.

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, look-ahead bias is a natural concern. 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 over 250,000 news articles by neutering each article and then confirming that the LLM cannot recognize the subject

firm or time period for well over 99% of the articles. In addition, when we compare the sentiment extracted from the raw text to the sentiment from 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.

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

References

Bybee, J Leland, 2023, 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.

Carlini, Nicholas, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang, 2023, Quantifying memorization across neural language models, in *The Eleventh International Conference on Learning Representations*.

Carlini, Nicholas, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al., 2021, Extracting training data from large language models, in *30th USENIX security symposium (USENIX Security 21)*, 2633–2650.

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.

Fang, Lily, and Joel Peress, 2009, Media coverage and the cross-section of stock returns, *The journal of finance* 64, 2023–2052.

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* .

Gurun, Umit G, and Alexander W Butler, 2012, Don't believe the hype: Local media slant, local advertising, and firm value, *The Journal of Finance* 67, 561–598.

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.

Levy, Bradford, 2024, Caution ahead: Numerical reasoning and look-ahead bias in AI models, Working paper.

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.

Staab, Robin, Mark Vero, Mislav Balunović, and Martin Vechev, 2024, Large language models are advanced anonymizers, *arXiv preprint arXiv:2402.13846* .

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.

Wang, Lei, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al., 2024, A survey on large language model based autonomous agents, *Frontiers of Computer Science* 18, 186345.

Figure 1: The entity neutering process

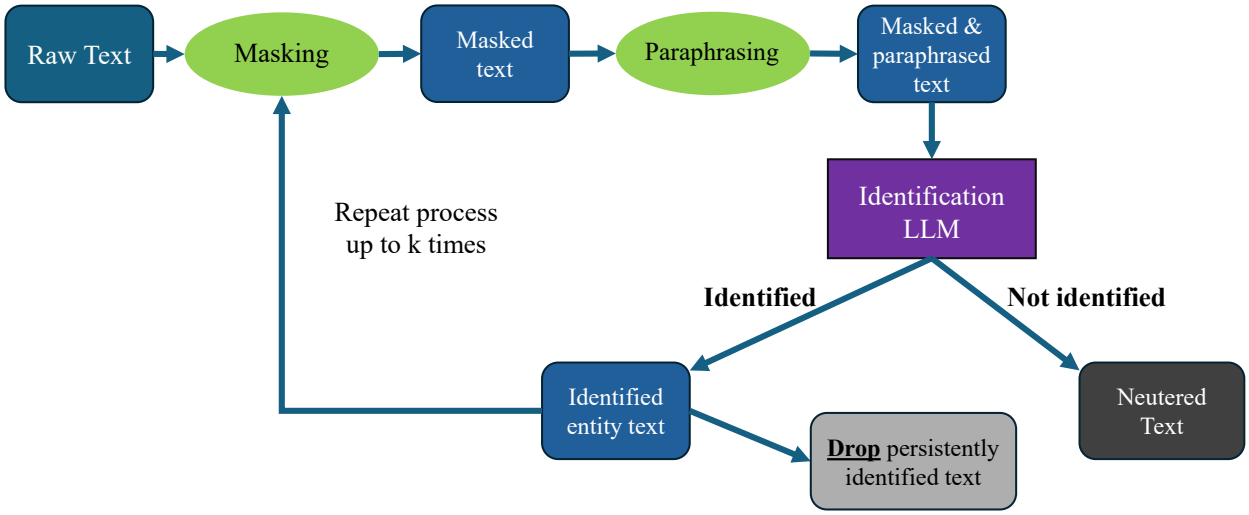


Figure 2: Procedure

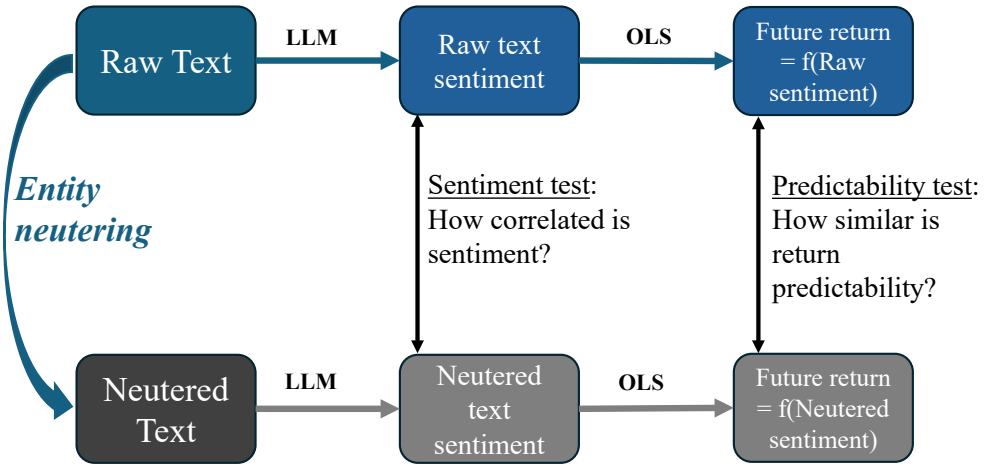
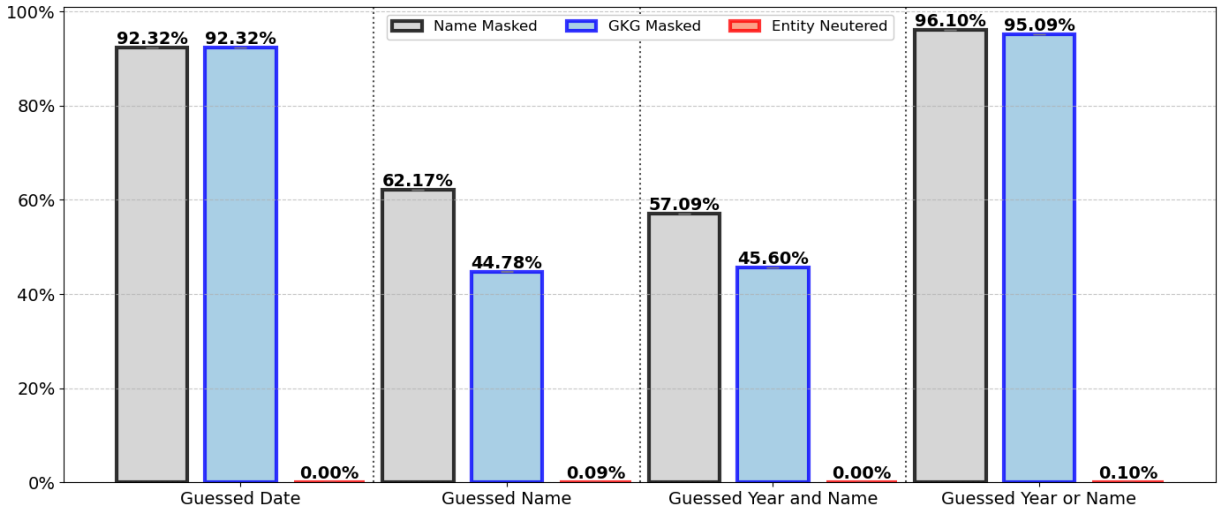
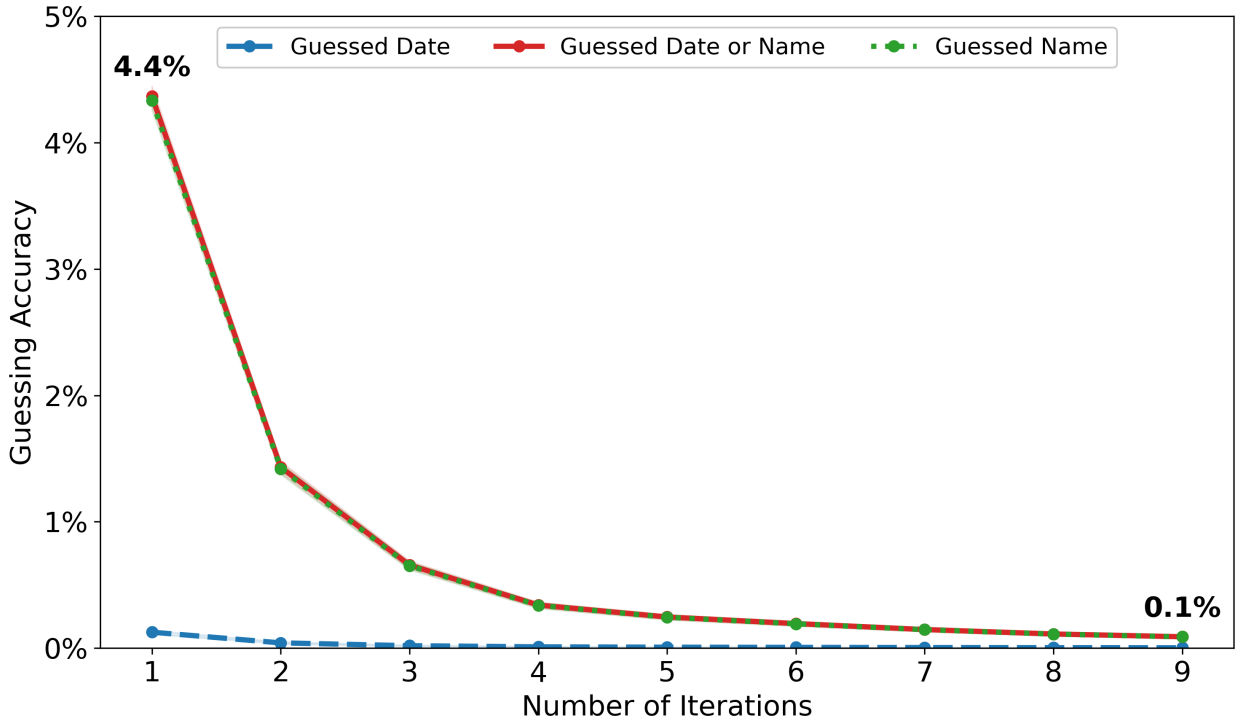


Figure 3: LLM guessing accuracy for news articles by de-identification method



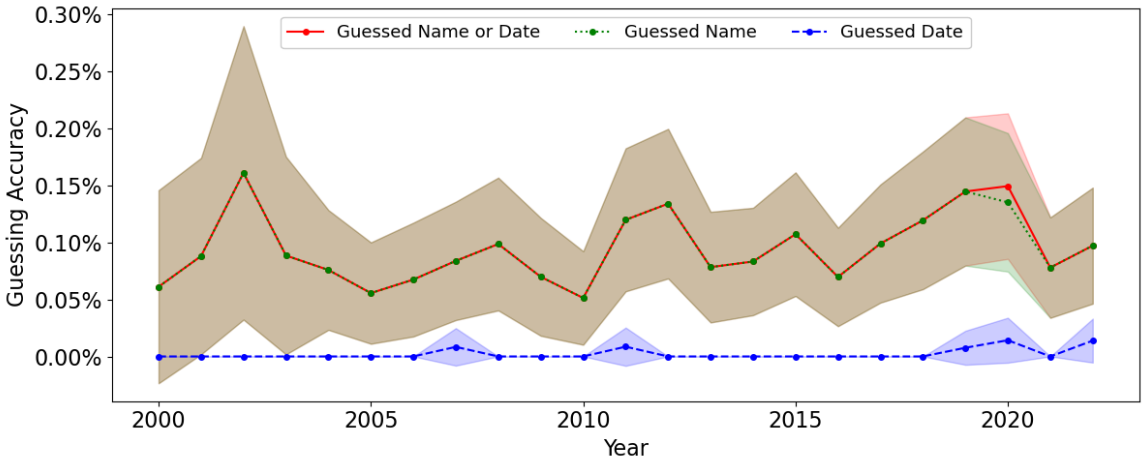
The columns labeled *Guessed Date* report the share of cases in which the LLM correctly guessed the date of the article within one week of the actual publication date (i.e., +/- 7 days of the publication day). The *Guessed Name* columns correspond to correctly guessing the firm name. The columns labeled *Guessed Date and Name* correspond to the LLM correctly guessing both year and firm name, whereas *Guessed Date or Name* corresponds to the LLM correctly guessing either the date or the firm name. Each color bar represents a different method of de-identification. The leftmost (gray) bars mask firm names; the middle (blue) bars mask up to 20 firm-associated terms using Google Knowledge Graph (GKG) along with the name and ticker (following Glasserman and Lin, 2023); the rightmost (red) bars correspond to our entity-neutering procedure after 9 iterations.

Figure 4: The neutering process over each iteration



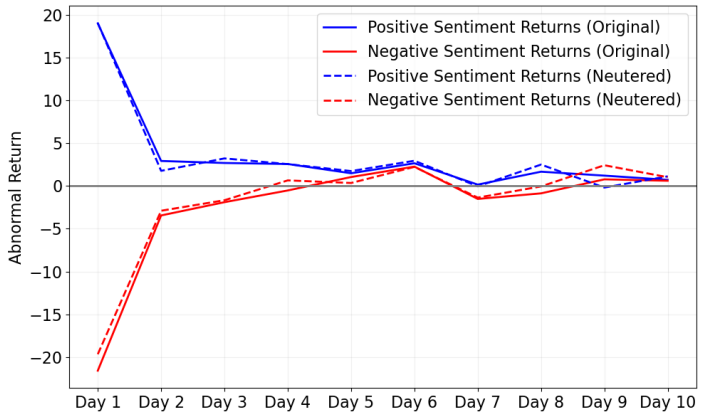
The horizontal axis shows how many times the sample has been through the neutering process (Figure 1). Only unidentified articles are included in each iteration. For each iteration, we plot the guessing rate for date, name, and name or date.

Figure 5: Identification accuracy over time

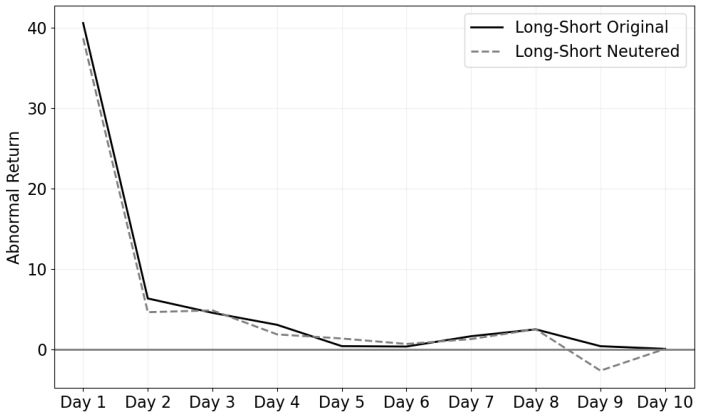


The figure shows ChatGPT's guessing accuracy for each year on neutered text, along with confidence intervals at the 5% level.

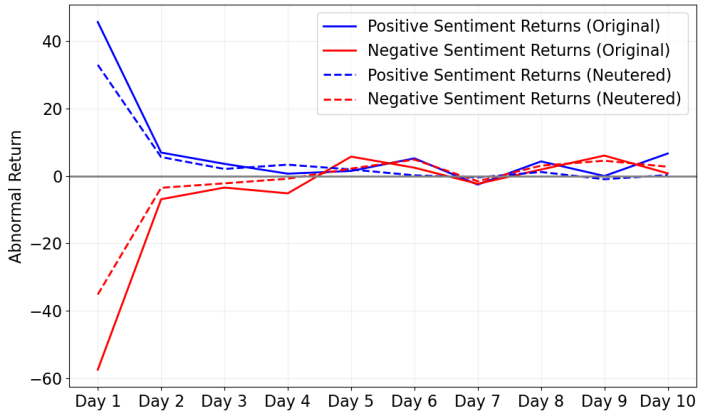
Figure 6: Event study returns by news sentiment



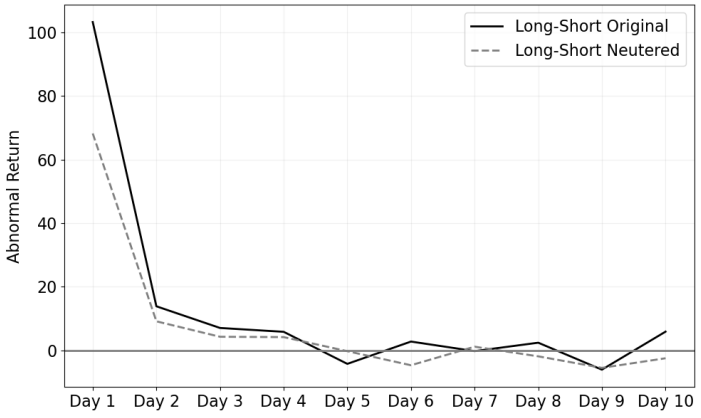
(a) Positive and negative



(b) Positive − negative



(c) Strong sentiment



(d) Strong sentiment long − short

The figures plot average DGTW returns in event time by sentiment, which is extracted from DJNW news articles. The blue lines (in the top half of 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 figures) correspond to negative sentiment articles. The black lines in panels (b) and (d) are the difference between the two. Solid lines indicate sentiment extracted from the raw text, dotted lines indicate sentiment extracted from 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 name identification rate for DJNW news articles by LLM

Firm	Model	# parameters	identification rate	# obs.
OpenAI	GPT-5 nano	unknown	0.09%	255,328
Meta	Llama 3.2	3B	0.04%	150,000
Google	Gemma 3	4B	0.06%	150,000

The table presents LLMs' rate of correct guesses of firm names based on entity neutered articles. The first row is the main result of the paper, using OpenAI's GPT-5 nano model. Subsequent rows use other LLMs, with the same random subsample of articles.

Table 2: Return predictability for DJNW news articles

	Outcome: $DGTW_{i,t+1}$				$DGTW_{i,(t+2) \to (t+5)}$			
	(1) Raw	(2) Neutered	(3) Raw	(4) Neutered	(5) Raw	(6) Neutered	(7) Raw	(8) Neutered
$\mathbb{1}$ Positive sentiment$_{i,t}$	14.46***	17.56***			3.07	5.33*		
	[7.38]	[9.69]			[0.80]	[1.69]		
$\mathbb{1}$ Negative sentiment$_{i,t}$	-28.29***	-22.99***			-9.77**	-5.76*		
	[-13.59]	[-11.90]			[-2.56]	[-1.82]		
$\mathbb{1}$ Strong positive$_{i,t}$			43.25***	31.91***			11.54*	13.17**
			[10.42]	[8.42]			[1.71]	[2.25]
$\mathbb{1}$ Weak positive$_{i,t}$			9.97***	15.13***			1.75	4.03
			[5.10]	[8.12]			[0.45]	[1.23]
$\mathbb{1}$ Weak negative$_{i,t}$			-18.95***	-17.61***			-8.79**	-5.75*
			[-9.37]	[-8.74]			[-2.27]	[-1.74]
$\mathbb{1}$ Strong negative$_{i,t}$			-65.58***	-39.24***			-13.57**	-5.67
			[-15.32]	[-11.82]			[-2.15]	[-1.08]
$DGTW_{i,t}$	-0.04***	-0.04***	-0.04***	-0.04***	-0.02**	-0.02**	-0.02***	-0.02**
	[-6.22]	[-6.15]	[-6.56]	[-6.24]	[-2.54]	[-2.52]	[-2.58]	[-2.53]
$DGTW_{i,t-1}$	-0.01**	-0.01**	-0.01**	-0.01**	-0.04***	-0.04***	-0.04***	-0.04***
	[-2.27]	[-2.29]	[-2.30]	[-2.30]	[-4.24]	[-4.24]	[-4.24]	[-4.24]
$DGTW_{i,t-2}$	-0.01**	-0.01**	-0.01**	-0.01**	-0.04***	-0.04***	-0.04***	-0.04***
	[-2.23]	[-2.26]	[-2.27]	[-2.26]	[-4.67]	[-4.68]	[-4.68]	[-4.68]
$DGTW_{i,(t-21) \to (t-3)}$	-0.00*	-0.00**	-0.00**	-0.00**	-0.01**	-0.01**	-0.01**	-0.01**
	[-1.90]	[-2.03]	[-2.00]	[-2.10]	[-2.31]	[-2.34]	[-2.32]	[-2.34]
$DGTW_{i,(t-252) \to (t-22)}$	0.00	0.00	-0.00	-0.00	-0.00***	-0.00***	-0.00***	-0.00***
	[0.14]	[0.07]	[-0.06]	[-0.03]	[-2.75]	[-2.77]	[-2.77]	[-2.78]
Log($BM_{i,t}$)	-1.85*	-1.62	-1.57	-1.64	-4.73***	-4.61***	-4.65***	-4.58**
	[-1.72]	[-1.50]	[-1.46]	[-1.52]	[-2.66]	[-2.60]	[-2.62]	[-2.57]
Log($ME_{i,t}$)	-1.62***	-1.65***	-1.55***	-1.58***	-3.48***	-3.52***	-3.50***	-3.57***
	[-3.66]	[-3.72]	[-3.50]	[-3.56]	[-4.44]	[-4.47]	[-4.47]	[-4.54]
Log($ST_{i,t}$)	-0.70	-0.76	-0.30	-0.68	-0.23	-0.26	-0.24	-0.30
	[-0.75]	[-0.81]	[-0.32]	[-0.73]	[-0.14]	[-0.17]	[-0.15]	[-0.19]
Constant	27.15***	24.92***	27.07***	24.04***	53.28***	51.64***	53.64***	52.26***
	[4.24]	[3.91]	[4.22]	[3.77]	[4.39]	[4.43]	[4.42]	[4.49]
# observations	242,234	242,234	242,234	242,234	242,145	242,145	242,145	242,145
# clusters (date)	5,781	5,781	5,781	5,781	5,781	5,781	5,781	5,781
Adj. R^2	0.006	0.005	0.007	0.005	0.002	0.002	0.002	0.002
RMSE	362.413	362.542	362.155	362.486	621.553	621.563	621.550	621.563

Strong (weak) sentiment is an indicator for articles with Z-scored sentiment ≥ 1 standard deviation above the mean, and is calculated separately among positive and negative sentiment articles, using information up to $t-1$. Coefficients are in basis points, t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Table 3: Trading strategy returns using DJNW news article sentiment

	Dep. var.: Long-short$_{t+1}$					
	(1)	(2)	(3)	(4)	(5)	(6)
	Raw	Neutered	Difference	Raw & strong	Neutered & strong	Difference & strong
α	48.34***	43.14***	5.20**	70.09***	49.83***	20.26***
	[18.18]	[15.89]	[2.50]	[16.01]	[14.55]	[5.28]
Market	-0.04	-0.04	-0.00	-0.02	-0.04	0.02
	[-1.29]	[-1.18]	[-0.08]	[-0.46]	[-0.89]	[0.36]
HML	0.12*	0.05	0.07	0.15	0.09	0.06
	[1.65]	[0.70]	[1.21]	[1.21]	[1.22]	[0.54]
SMB	0.15**	0.13**	0.02	0.09	0.15**	-0.06
	[2.35]	[1.97]	[0.43]	[0.92]	[1.99]	[-0.61]
UMD	0.04	0.00	0.04	0.10*	0.05	0.05
	[1.11]	[0.03]	[1.46]	[1.71]	[1.11]	[1.03]
RMW	0.02	-0.04	0.06	-0.14	-0.02	-0.12
	[0.25]	[-0.51]	[0.96]	[-1.12]	[-0.18]	[-1.14]
CMA	-0.17	-0.01	-0.16*	-0.17	0.01	-0.19
	[-1.44]	[-0.06]	[-1.77]	[-1.04]	[0.10]	[-1.17]
# observations	5,702	5,702	5,702	5,702	5,702	5,702
Adj. R^2	0.003	0.001	0.001	0.001	0.002	0.000

Trading strategy returns are constructed by equal weighting on day t and holding through day $t+1$. In columns 1 and 2 we go long stock with positive sentiment news and short stocks with negative sentiment news; in columns 3 and 4 we go long stocks with strong positive sentiment news and short stocks with strong negative sentiment news. Columns 3 and 6 report the differences between 1 and 2, and 4 and 5, respectively. All regressors measured at $t+1$. Coefficients are in basis points, with robust t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Internet Appendix:

Entity Neutering

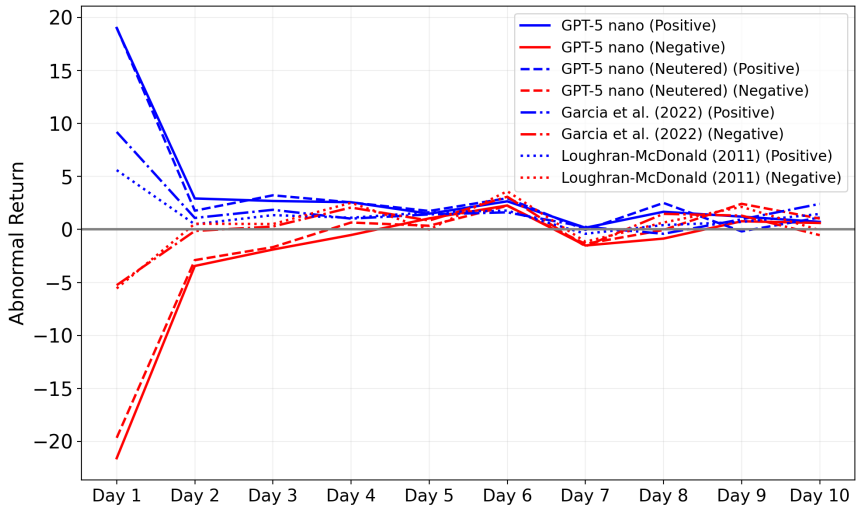
by Joseph Engelberg, Asaf Manela, William Mullins & Luka Vulicevic[1]

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

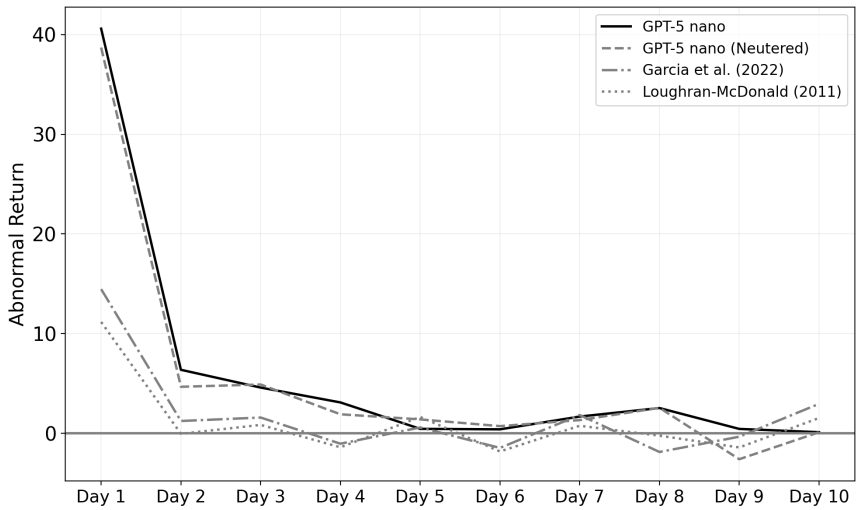
A. Additional Results

Figure A1: Event study returns by sentiment: LLM vs Dictionary approach

Panel A: Positive and negative

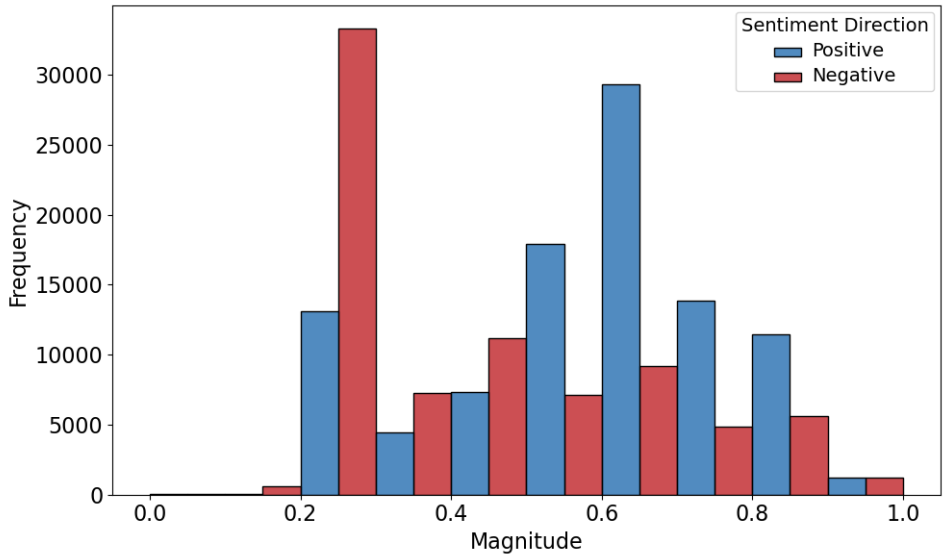


Panel B: Positive − negative



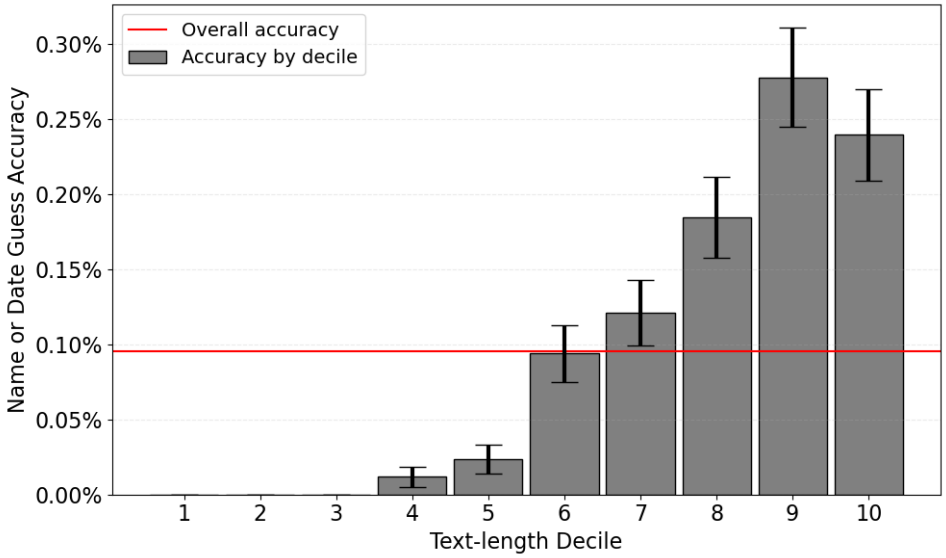
The figures plot average DGTW returns, with positive or negative sentiment extracted from DJNW news articles. As in Figure 6, the solid and dashed lines show returns where sentiment is extracted using ChatGPT from the raw and neutered text, respectively. This figure differs from Figure 6 by including dotted lines showing returns where sentiment is measured using the Loughran and McDonald (2011) dictionary approach and the Garcia et al. (2023) unigram dictionary. For both dictionary approaches we determine a text's sentiment by calculating the number of negative unigrams over the sum of positive and negative unigrams, and assigning negative sentiment to texts with a below-median share in an expanding time window, and positive sentiment to those above the median. The blue lines (in the top half of Panel A) indicate the return of articles with positive sentiment on day 0, while the red lines (in the bottom half of Panel A) correspond to negative sentiment articles. The black lines in Panel B show the difference in returns between positive and negative sentiment articles.

Figure A2: Distribution of sentiment magnitude for DJNW news articles



The figure displays the distribution of the magnitude of sentiment generated by the LLM from DJNW news articles. 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 A3: Firm or date identification rate by DJNW news article length



This figure plots the identification rate of firm or date as a function of DJNW 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 date. Average accuracy after nine iterations is 0.1% and is plotted with a red horizontal line. The average number of words for each decile bin is: 92 (decile 1), 109, 119, 151, 217 (decile 5), 301, 411, 518, 712, and 1,143 (decile 10).

Table A1: Filtering steps for the DJNW news sample

Filter	Sample size	N. of obs. dropped
DJNW archive	40,954,779	
First article in chain	27,276,473	13,678,306
One stock tagged	13,756,026	13,520,447
Body longer than 500 char.	9,240,090	4,515,936
Unique map to CRSP	2,554,091	6,685,999
Random sample	255,410	2,298,681
ChatGPT refusal	255,328	82
Entity neutered and unidentified	255,085	243
Control variables exist	242,234	12,851

This table shows the number of observations remaining at each step of the sample filtering process, as well as the number of observations dropped when moving from the preceding step. We follow Ke, Kelly, and Xiu (2019), but with an additional filter on article length. After taking a 10% random sample, we then apply the entity neutering procedure. "ChatGPT refusal" refers to cases in which GPT-5 nano's safety protocol prevented the entity neutering procedure from starting, forcing us to drop those 82 observations. "Unidentified" articles are those that the LLM was unable to correctly identify from the neutered text either the entity name or the date of the article (we count as correct any date guess that is within a $\pm$7-day window of the true date). 243 articles were still identified after the entity neutering procedure described by Figure 1, and were dropped. "Control variables" refer to those used in Table 2. The final line shows the number of observations in the regression sample.

Table A2: Identification rates for DJNW news articles by firm characteristic

Data	Guess name	Guess date	Guess date or name	Guess date and name
Small	0.02%	0.00%	0.02%	0.00%
Medium	0.07%	0.00%	0.07%	0.00%
Large	0.22%	0.01%	0.22%	0.00%
Value	0.04%	0.00%	0.04%	0.00%
Growth	0.15%	0.00%	0.15%	0.00%
All	0.09%	0.00%	0.10%	0.00%

The table reports the entity identification rate for DJNW news articles by firm characteristics. The first three rows split firms into market capitalization bins (the bottom quartile, middle two quartiles, and top quartile). The last two rows split firms into bins by book to market and represent the bottom half of firms (Value) and the top half of firms (Growth). All firm characteristics are measured yearly.

B. Prompts

B.1 Masking prompt

We use the following LLM prompt to mask the original text of identifying features:

> *Your role is to ANONYMIZE all text that is provided by the user. The text you need to anonymize will be about firm {NAME}, its ticker, {TICKER}, and its industry, {INDUSTRY}. Any instance of the firm, their products, the industry or industry related areas they are involved in should be anonymized. Absolutely no mention of key words related to that industry can be left in the text. After you have anonymized a text, NOBODY, not even an expert financial analyst should be able to use the text to identify the company or 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 2022, 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 to make an educated guess about its identity. Stock tickers are identifiers, and are usually four or fewer capitalized letters (use TIKR as a stand-in for an arbitrary ticker), and may be referenced in the text in the following formats; SYMBOL:TIKR, $TIKR, >TIKR, $ TIKR, SYMBOL TIKR, SYMBOL: TIKR, > TIKR. Make sure you ANONYMIZE TIKR to ticker_x. Also ANONYMIZE any other identifiers related to companies including 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 anonymized analog. After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.

B.2 Paraphrasing prompt

We use the following LLM prompt to paraphrase the original text:

You are to paraphrase text ensuring that its core information content is unchanged, such that someone who has memorized the text could not recognize it. Make sure to replace all unique or identifying words and phrases. Also change the sentence structure so that the resulting text is completely different from the original. This includes changing quotes and other highly recognizable structures in the text. The text provided will be about firm {NAME}, its ticker, {TICKER}, and its industry, {INDUSTRY}. Any information which could ever be used to identify the firm that the text is about or when the text was written MUST be anonymized. Absolutely no mention of key words related to that industry or any other industries can be left in the text. If you observe any names, proper nouns, aliases, acronyms, dates, industry references (related and unrelated to the industry of the firm in question), government agency names, product names or types, etc., replace them with anonymous strings. Anonymity is the prime objective, leave zero hints for someone who has memorized the text! Ensure that the paraphrased text has approximately the same number of words as the original text. After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.

B.3 Iterated Masking prompt

We use the following LLM prompt to mask the text for each iteration after the initial round of masking an paraphrasing. For brevity, we only show how the prompt is different compared to the original masking prompt:

> *Your role is to ANONYMIZE all text that is provided by the user. You have already tried to anonymize the following text and failed, leaving details which were used to identify the text by an expert. Be more aggressive in your efforts to anonymize and pay more attention, so that you do not leave even the most minute details in the anonymized text. The text you need to anonymize will be about firm {NAME}, its ticker, {TICKER}, and its industry, {INDUSTRY}.*
>
> ...
>
> *You have already failed me before, take no chances this time and be aggressive in censoring the text. After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.*

B.4 Iterated Paraphrasing prompt

We use the following LLM prompt to paraphrase the text for each iteration after the initial round of masking an paraphrasing. For brevity, we only show how the prompt is different compared to the original paraphrasing prompt:

> *You are to paraphrase text ensuring that the core information content is unchanged, such that someone who has memorized the text could not recognize it. You have already failed to paraphrase the text such that it is anonymous. Be more aggressive in your paraphrasing and anonymizing of the text. Make sure to replace all unique or identifying words and phrases.*
>
> ...

You have already failed once before, take no chances this time, be aggressive in your effort to anonymize the text. After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.

B.5 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 of your knowledge and any clues in the text to identify which company the text is about, its industry, and the year it was written in. Make your best guess if you are unsure. Your options for industry guesses are the following: Aerospace, Agriculture, Automobiles and Trucks, Banking, Beer and Liquor, Construction Materials, Printing and Publishing, Shipping Containers, Business Services, Chemicals, Computers, Apparel, Construction, Coal, Pharmaceutical Products, Electrical Equipment, Fabricated Products, Financial Services, Food Products, Recreation, Precious Metals, Defense, Computer Hardware, Healthcare, Consumer Goods, Insurance, Measuring and Control Equipment, Machinery, Restaurants, Hotels, and Motels, Medical Equipment, Non-Metallic and Industrial Mining, Petroleum and Natural Gas, Paper, Personal Services, Real Estate, Retail, Rubber and Plastic Products, Shipbuilding, Railroad Equipment, Tobacco Products, Candy and Soda, Software, Steel Works, Communication, Entertainment, Transportation, Textiles, Utilities, Wholesale. Provide your estimate EXACTLY in the following format, with NO other text at all (TIKR is your estimate of the ticker, NAME is your estimate of the firm name, IND is your estimate of the firm's industry, Y is your estimate of the year): **Ticker Estimate: TIKR**,**Name Estimate: NAME**,**Industry Estimate: IND**,**Year Estimate: Y***

B.6 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**"*