Entity Neutering

Joseph Engelberg
UC San Diego

Asaf Manela
Wash. U.

William Mullins
UC San Diego

Luka Vulicevic
*UC San Diego**

June 24, 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, 2023, 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 1,056,053 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, ChatGPT 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, that is, 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 ChatGPT and two other leading LLMs, Llama 3 and Gemma 3. All LLMs struggle to correctly guess the subject firm in neutered articles, with ChatGPT guessing correctly 6.81% of the time and Llama and Gemma guessing correctly around 7% of the time. The LLM's ability to guess the date of publication is even lower. For example, ChatGPT is able to guess the date within a two week window only 0.19% of the time.

To recap, among the 1,056,053 news articles, after neutering, ChatGPT is only able to identify the firm or the date in 71,968 articles. We drop these "identifiable" articles from our sample and are left with 984,085 "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, and additionally ask for a measure of sentiment strength. We do this for both the raw and neutered articles. The sentiment

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

of the neutered text matches the sentiment of the raw text 90.1% 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 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 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 9 basis points (bps), while the corresponding return for stocks with positive sentiment measured from the *raw* text is 10 bps (Figure 5), 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 bps for the raw text and 10 bps for the neutered text. The corresponding figures among strongly negative articles are -19 and -14. 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 -37 bps when using the raw text, but -33 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*

[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.

sentiment are -45 bps and -39 bps; for *Strong positive* sentiment they are 24 bps and 13 bps.

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 24% when predicting next-day returns and 28% when predicting returns on days $t+2$ through $t+5$. The corresponding upper bounds are 19% and 15% when using the specifications where we group sentiment into the coarser positive and negative categories.

Finally, we perform our entity neutering procedure on another set of texts that have shown evidence of return predictability: over 280,000 Management Discussion and Analysis (MD&A) sections from firm 10-Q and 10-K filings between 2002 and 2023. Although these texts are, on average, nearly twenty times longer than a news article, we continue to find evidence that entity neutering is effective. ChatGPT recognizes the firm name only 12% of the time and recognized the date only 2% of the time after neutering. When using MD&A text to predict future returns we also find that the sentiment from the raw text has better predictability than the sentiment from neutered text, especially after event day 1. Depending on the specification, in the MD&A setting we calculate a bound on look-ahead bias of 32% to 42% for event day 1 and 67% to 71% for event days 2 through 5. These bounds suggest that the look-ahead bias problem may be more severe for MD&As than for news articles, especially when predicting returns beyond event day 1.

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, which can introduce lookahead 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 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 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 7% of the time in news text and 14% of the time in MD&A text, while in Sarkar and Vafa (2024) this number is 70% (and 65% in our sample using their name masking 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 6 concludes.

[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.

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 $Q\left(\cdot\right) \in \mathcal{Q}$ that summarizes the sampling distribution over economically relevant texts and beliefs about the LLM's sampling distribution over training datasets. $Q\left(\cdot\right)$ is a joint distribution over the sampling indicators $(D, \mathcal{T})$. Denote by $q_{it}^{\mathcal{T}|D}\left(\tau_{it}\right) = Q\left(\mathcal{T}_{it} = \tau_{it} | D_{it} = 1\right)$ 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 $q_{it}^{\mathcal{T}}\left(\tau_{it}\right) = Q\left(\mathcal{T}_{it} = \tau_{it}\right)$ and $q_{it}^{D}\left(d_{it}\right) = Q\left(\mathcal{D}_{it} = d_{it}\right)$.

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: $q_{it}^{\mathcal{T}|D}(\tau_{it}) > q_{it}^{\mathcal{T}}(\tau_{it})$.

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 $n(x_{it}; \tau)$.

[5]Suppose that the LLM $m\left(\cdot; \tau\right)$ 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 $q_{it}^{\mathcal{T}|D}\left(\tau_{it}\right)$ that text piece x_{it} appears in the training sample is zero, as is the marginal probability $q_{it}^{\mathcal{T}}\left(\tau_{it}\right)$ 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.

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 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 infor-

mation 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:

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

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 $m\left(\cdot;\tau\right)$ 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,*

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

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 from the Dow Jones archive from 2000 to 2009, and we filter the articles following the procedures in Ke, Kelly, and Xiu (2019). The number of articles remaining after each stage of article filtering is reported in Appendix Table A1. In order to explore the sensitivity of our results to the 2000-2009 period, we also take a random sample of 1 million articles from the Dow Jones archive through August 2, 2023.[6]

To test the robustness of our approach on alternative text, we also collect Management's Discussion and Analysis ("MD&A") sections from firm 10-K and 10-Q filings. We collect these sections from the University of Notre Dame repository provided by Loughran and McDonald (2016) and filter the sample according to the steps in Appendix Table A3. The sample has 283,660 observations and covers from March 2002 to September 30, 2023, the training cutoff date for GPT-4o mini.

[6]The 1M articles are collected using a stratified random sampling procedure. The procedure takes 175 randomly selected firms that had articles published about them on a given day. One random article is then sampled for each of these 175 firms. This is repeated for each day in the sample.

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 (such as names, dates, products, industries) but additionally instructs the LLM to use its best judgment in removing additional 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's name, its ticker, its industry, and the date of the article. We refer to this as the *neutering test* in Figure 1.

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 date). For the remaining articles, we 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 B.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 date 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 date it was written. Make your best guess based on information and context if you are unsure.

The full prompt is in Appendix B.2.

The results are presented in Table A4 and Figure 2. Among the 1,056,053 articles, ChatGPT is able to identify the subject firm in 71,968 of the cases, or 6.81% of the time and it is able to identify the date of the article correctly only 0.19% of the time. It is able to

identify *either* the date or the name of the firm only 7% of the time. Given these low rates of identification, we conclude the neutering process is relatively 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 12.23% of the time while small firms (bottom quartile) are guessed correctly 3.39% of the time. We also find a slight increase in guessing probability for value stocks compared to growth stocks (7.70% vs 5.90%).

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, ChatGPT 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 date of the article 90.8% 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 2 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 47% of the time.

Figure 3 suggests that the benefit of entity neutering compared with other methods comes from allowing the LLM to use its best judgement in de-identifying text. The figure plots the guessing accuracy as a function of *components* of our neutering prompt. For example, when we tell the LLM only, "*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 use the text to identify the company, the industry the company operates in*" its

[7]Figure A2 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 7% average as the number of characters in the article increases, it stabilizes and remains below 12% even as the number of words exceeds 1,000.

guessing accuracy with respect to firm name is 7.8% compared to 6.81% for the full prompt. This suggests that we do not necessarily need to specify anonymizing names, products or industries (as we do in the full prompt) because the LLM recognizes that these are clues to identification. However, the figure also demonstrates that, unsurprisingly, it is necessary to specify anonymizing dates in the article in order to lower the date guessing accuracy.

It is possible that ChatGPT is typically unable to guess the entity or date of neutered articles because of our 2000 – 2009 time period. 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 take a random sample of one million Dow Jones articles between 2000 and 2023, entity neuter these articles, and ask ChatGPT to guess the subject firm and date. We plot the frequency of correct guessing by year in Figure 4. For firm names, there is no discernible time-trend: overall, the LLM guesses correctly around 6% of cases. However, ChatGPT improves its guessing accuracy about dates in the final five years of the sample, peaking at an accuracy rate of around 2% in 2020 and 2023. This potentially reflects the fact that, when the LLM does not know the date of the article and must make a guess to satisfy our prompt, it guesses more recent periods, because text from those periods appear more frequently in its training corpus.

4.2.1 Robustness to alternate LLMs

A natural concern is that an 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 A4 demonstrate that LLMs other than of ChatGPT also display the same overall inability to identify the subject of a neutered article.

We feed the neutered articles to two different LLMs: Meta's Llama 3.2 class of models 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 5.6% of the time. Likewise, Google's Gemma 3 identifies the firm or the date correctly 6.7% 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 to what extent our entity neutering has "thrown out the baby with the bathwater." We begin with the subsample of articles for which ChatGPT was unable to guess either the firm or date in Section 4.2. Thus, we take our sample of 984,085 original articles and 984,085 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 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.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 467,751 original articles with positive sentiment according to ChatGPT, 91.5% of these also have positive sentiment in the corresponding neutered article. Likewise, among

the 375,074 original articles that ChatGPT 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 427,975 articles that ChatGPT assigns positive sentiment to both the original and neutered versions, the pair-wise rank correlation of their magnitude is 72.6%. Among negative sentiment articles, the rank correlation is 72.8%.

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 5. 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 returns for articles with sentiment extracted from the raw text is denoted by a solid line and by a dotted line for those for articles with sentiment extracted from the neutered text.

Panel (a) of Figure 5 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 bps, while from the raw text this number is 10 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 bps, 5 bps and 1 bps while the corresponding returns using the neutered articles are 20 bps, 5 bps and 1 bps.

Panels (c) and (d) of Figure 5 mirror those of (a) and (b) among the set of articles with *strong sentiment*. These are articles for which ChatGPT 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 32 bps, compared to 24 bps for the neutered articles.

Table 2 presents the regression analogue of Figure 5 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 be matched to DGTW returns (see Appendix Table A1 for additional details about this process). 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) through (4) is stock $i's$ abnormal return on day $t+1$, while in columns (5) to (8) it is the abnormal return from day $t+2$ to $t+5$. [8] The odd columns measure sentiment using 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 5, with the sharpest difference in coefficients between neutered and raw articles coming from those with positive sentiment and from those that indicate strong sentiment. For example, when predicting next-day abnormal return, the coefficient on *Negative sentiment* from both the raw text and the neutered text is almost identical, -10 bps. However, when comparing *Strong negative* sentiment in columns (3) and (4) the comparison is now -37 bps

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

for the raw versus -33 bps for the neutered, with the difference now statistically significant at the 5% level.

Table 3 presents the results from a calendar-time trading strategy based on sentiment, which complements the event-time results in Figure 5 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 statistically insignificant. Only when we condition on strong sentiment (i.e., long *Strong positive* sentiment and short *Strong negative* sentiment) do we see that the 38 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.

4.4.1 Bounding the lookahead 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 63, 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 51, which suggests that at most (63 - 51) / 63 = 19% of the predictability for $t+1$ DGTW returns from the raw sentiment is coming

from look-ahead bias.[9] Similarly, if we predict $t+2$ to $t+5$ DGTW returns (using columns 5 and 6), at most 15% is coming from look-ahead bias.

The bound calculations generally indicate that the greatest possibility for look-ahead bias occurs among the models incorporating the strength of 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 sentiment strength (columns 3 and 4), we conclude that at most 24% is attributable to look-ahead bias; when predicting $t+2$ to $t+5$ returns at most 28% is coming from look-ahead bias.

5. Entity Neutering in an alternate setting: Management Discussion and Analysis data

The effectiveness of entity neutering likely depends on the type of text being processed. News articles are typically short and are often written from the perspective of a reporter. In this section, we use our entity neutering procedure on text which is longer and is written from the point of view of managers: the Management Discussion and Analysis (MD&A) section of quarterly filings (10-Qs and 10-Ks). Like news content, the textual sentiment of MD&As has also been shown to predict returns (Feldman et al., 2010).

The effectiveness of entity neutering on MD&As could be different for a few reasons. First, the average number of words in an MD&A is much higher than that of a news article: 7,285 versus 399. Thus, even after the LLM removes entity-identifying clues, MD&As provide far more text that the LLM can use in order to identify the firm. In addition, because MD&As are written from the point of view of managers (not journalists) they tend to be more introspective about the firm's business, and this specificity may provide the LLM with more entity-identifying clues.

Our sample consists of 283,660 MD&As between 2002 and 2023 (see Appendix Table A3).

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

We neuter each MD&A using the same approach as in Section 4.1, asking ChatGPT to remove any entity-identifying information from the text. Then, we feed the neutered MD&A back to a new instance of ChatGPT and ask if it can identify the firm or date. Overall, ChatGPT correctly identifies 12.12% of firm names and 2.33% of dates among the MD&As. The identification rate for firms is higher than for news articles, likely due to the considerable length of the MD&A text providing more clues for the LLM. Indeed, when we plot firm identification as a function of MD&A length deciles in Appendix Figure A4, there is a strong upward slope: MD&As with lengths in the bottom three deciles are identified at a rate below 10% but those in the top decile are identified at a 19% rate. However, identification rates are similar for news and MD&As when word counts are also similar. For example, firms in the bottom MD&A decile by word length average 679 words and are identified at a 9% rate, while firms in the top news decile by word length average 1232 words and are identified at a 11.8% rate.

We also find a time trend in firm identification rates among the MD&A seen in Figure 6. Rates begin at 8% in 2002, climb to about 17% in 2020 and remain flat thereafter. Date identification rates remain low and flat throughout the sample, except for the COVID period when they spike to nearly 15%, which suggests that management discussions about the effects of the pandemic provided date-relevant clues to the LLM that neutering was unable to remove.

Figure 7 plots the event study returns as a function of both raw and neutered MD&A sentiment, analogous to Figure 5 for the news data. 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. Like Feldman et al. (2010), we find that MD&A sentiment predicts returns. Moreover, as was true for news sentiment, the raw sentiment slightly outperforms the neutered sentiment almost everywhere. However, we find a stronger asymmetry between positive and negative sentiment among the MD&A text, with negative sentiment having substantially more predictive power than positive sentiment. For example, the DGTW return

for event day $t+1$ is -13 bps for negative sentiment among the raw text and -12 bps among the neutered text (panel a). The analogous numbers are 2 and 1 for positive sentiment. Overall, Figure 7 suggests that return predictability lasts about 5 days (panel c) and is slightly larger for MD&As with strong sentiment (panel d).

Table 4 presents the regression analogue of Figure 7. The table uses the same controls as in Table 2 and follows the same construction, with odd columns using the sentiment from the raw text and even columns using the sentiment from the neutered text. Columns (1)-(4) predict next day DGTW returns, while columns (5)-(8) predict returns on event days 2 through 5. The regression generally confirms the patterns present in Figure 7, with positive sentiment generally insignificant throughout. Only when we consider *strong* positive sentiment extracted from the raw text (column 3) do we find a result significant at the 1% level for positive sentiment (8 bps with a t-stat of 2.5). However, negative sentiment has greater predictability, especially when it is strong. For example, the coefficient on strong negative sentiment is -20 bps (t-stat -4.1) for the raw text in column 3 and -17 bps (t-stat -3.4) for the neutered text in column (4).

When we compare the predictability of the raw versus neutered text for DGTW returns on days 2 through 5 (columns 5 - 8), we find a larger gap in predictability between the two, with the raw text outperforming the neutered text. For example, the coefficient on negative sentiment in column 5 is -21 bps (t-stat -3.6) compared to -13 bps (t-stat -2.7) in column (6). For strong negative sentiment (columns 7 and 8) the corresponding numbers are -29 bps (t-stat -3.5) versus -13 bps (t-stat -1.6).[10]

Feldman et al. (2010) argues that not only is MD&A sentiment relevant but also the *change* in sentiment is important. For example, if a management team is consistently optimistic or pessimistic in their discussions, the most useful information might come from detecting a change in their sentiment. With that in mind, we follow Feldman et al. (2010)

[10]Wald linear restriction tests comparing the coefficients on sentiment measured in the raw vs. in the neutered text cannot reject that they are equal for every pair of coefficients in the table; *Strong negative* on days 2 to 5 is the only exception, with equality rejected at the 2% level.

and also compute the change in MD&A sentiment from successive quarters and relate the change to future returns. The results are in Table 5.

We take two approaches to calculating sentiment change. The first approach is to convert positive sentiment magnitudes to positive numbers and negative sentiment magnitudes to negative numbers to generate a continuous sentiment measure ranging from -1 to 1, and then take the difference across quarters. For example, if the LLM determined that this quarter's MD&A was positive with a magnitude of 0.7 and last quarter's was negative with a magnitude of 0.6 then the change would be calculated as: +0.7 - (-0.6) = 1.3.

Columns (1) and (2) of Table 5 take this approach for the raw and neutered text, respectively, when predicting DGTW returns for event day 1. In the regressions we include both the level of the sentiment for the current quarter, as well as the change from the prior quarter (Δ *sentiment*). The coefficients on each suggest that Δ *sentiment* is nearly as important as the level of sentiment. For example, the coefficient estimates on *Continuous Sentiment* and *Delta Sentiment* are 9 bps and 7 bps, respectively, in the specification using raw sentiment (column 1). Using neutered sentiment (column 2) the corresponding estimates are both 7 bps.

For the second approach, we create indicator variables for all possible sentiment changes from the prior quarter to the current quarter. Given sentiment can be either Positive, Negative or Neither, there are 3 x 3 = 9 possible changes: Positive to Negative, Positive to Neither, Negative to Negative, Negative to Positive, and so on. Columns (3) and (4) present the coefficients on 8 of these (the omitted category is Positive to Positive) for next day DGTW returns. Among all sentiment change conditions, the largest effect exists for the *Positive to Negative* condition, with a coefficient of -20 bps (t-stat -6.5) for raw sentiment (column 3) and -18 bps (t-stat -5.7) for neutered sentiment.

The final four columns consider both approaches when predicting DGTW returns on event days 2 through 5. While the change in sentiment continues to be significant, we also find a larger spread between the coefficients on raw and neutered sentiment, as we did in

Table 4.

Because we find stronger predictability from the raw text when compared to the neutered text throughout Tables 4 and 5, we can once again calculate a bound on lookahead bias as we did in Section 4.4.1. Using the specifications in Table 4, the bound on lookahead bias is 41% when predicting next-day returns using the directions of sentiment (positive and negative, columns 1 and 2) and 42% when using sentiment magnitudes (columns 3 and 4). When predicting returns on days 2 through 5, the corresponding bounds are 71% and 67%. If we instead use the specifications in Table 5 that incorporate the change in sentiment, the corresponding bounds on lookahead bias are 32% and 35% (for columns 1-2 and 3-4, respectively) when predicting next day returns, and 70% and 67% when predicting returns for event days 2 through 5. The larger bounds when predicting returns on event days 2 to 5 reflect the sizable spread in coefficients between the raw and neutered text, and suggest that the look-ahead bias problem may be more severe for MD&As than for news articles, especially after event day 1.

6. 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 1 million news articles by neutering each article and then confirming that the LLM cannot recognize the subject firm or time period for over 90% 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. When we apply the entity neutering

procedure to over 280,000 Management Discussion and Analysis sections from 10-K and 10-Q filings we find broadly similar results, despite these texts being approximately 20 times the length of the news articles, on average. Finally, we show how one can use the difference between predictions from the raw text and the neutered text to compute an upper bound on look-ahead bias. These bounds indicate that the look-ahead bias when using MD&A sections to predict abnormal returns may be more of a problem than it is for news articles.

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 process 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.

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.

Feldman, Ronen, Suresh Govindaraj, Joshua Livnat, and Benjamin Segal, 2010, Management's tone change, post earnings announcement drift and accruals, *Review of Accounting Studies* 15, 915–953.

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.

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.

Loughran, Tim, and Bill McDonald, 2016, Textual analysis in accounting and finance: A survey, *Journal of Accounting Research* 54, 1187–1230.

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

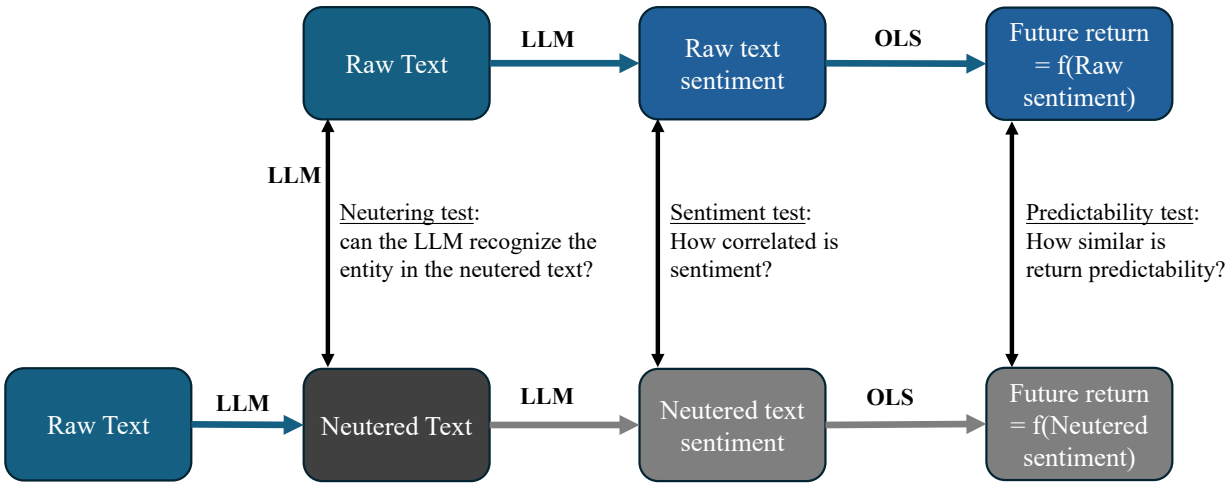
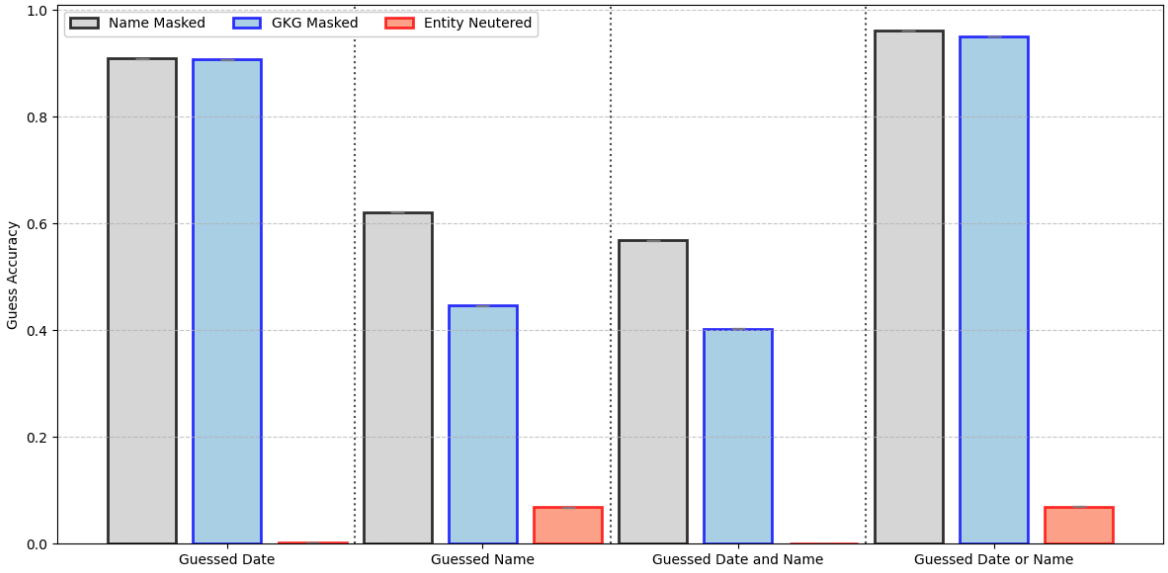
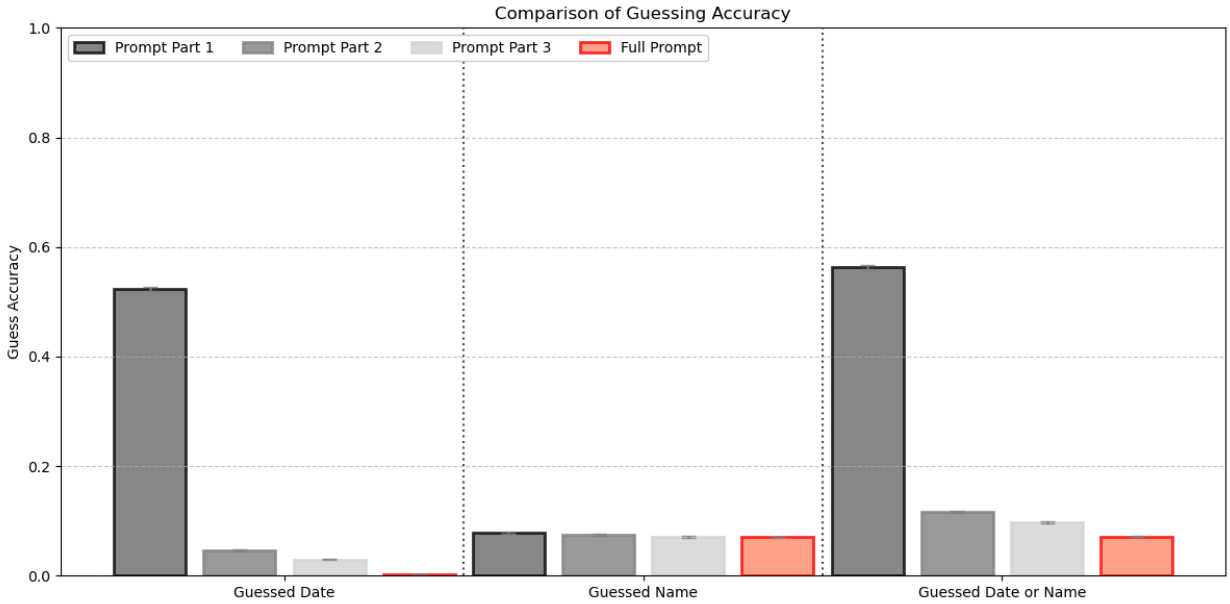


Figure 2: LLM guessing accuracy for News articles by de-identification method



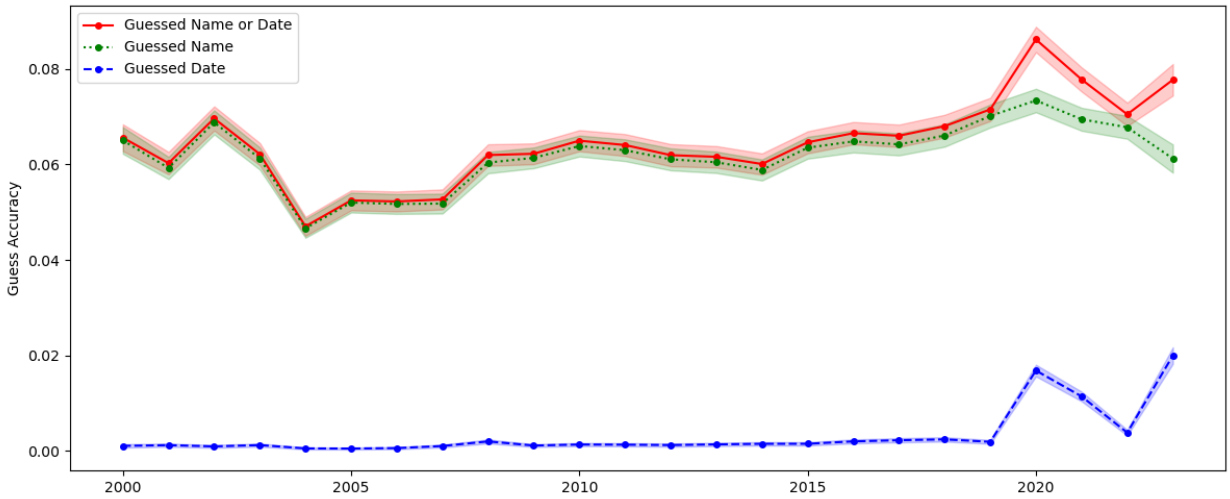
The columns labeled *Guessed Date* report the share of cases in which the LLM correctly guessed the date of the article within one week around the actual publication date. The *Guessed Name* columns correspond to correctly guessing the firm name. The columns labeled *Guessed Date and Name* correspond to the LLM correctly guessing both year and firm name, whereas *Guessed Date or Name* corresponds to the LLM correctly guessing either the year or the firm name. Each color bar represents a different method of de-identification. The leftmost (gray) bars mask all the firm names; 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: Identification rate for Entity neutered News articles by prompt parts



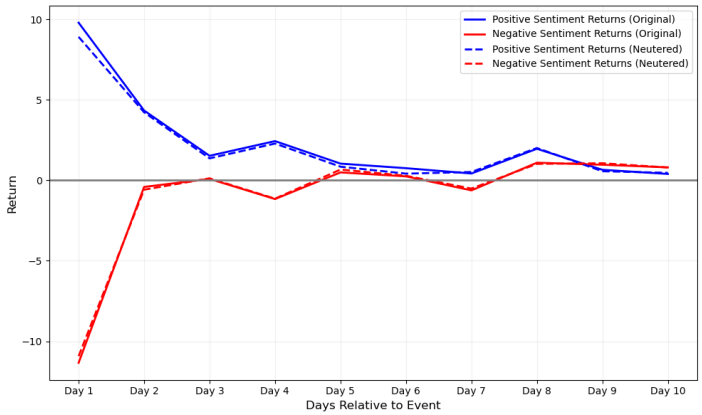
The above examines guessing accuracy of entity neutered text using different cumulative parts of the prompt seen in B.1. The sample used for the above graph is 100 thousand articles. These articles are random sampled from the 2000-2009 corpus and are the same across all versions of the prompt.

Figure 4: Entity neutered News article guessing accuracy over time

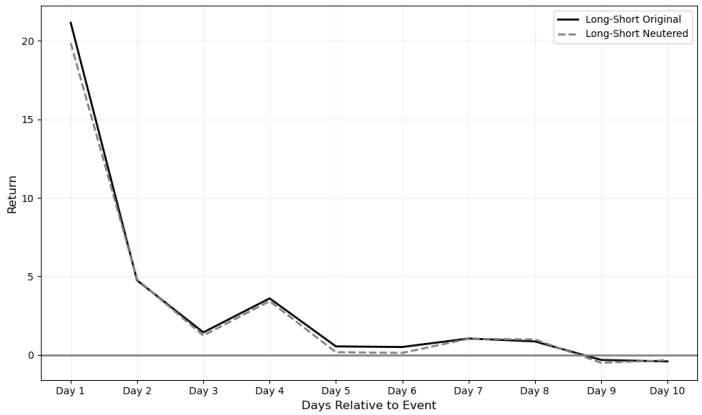


The figure shows guessing accuracy over time for the extended news article corpus, which ranges from 2000 to 2023.

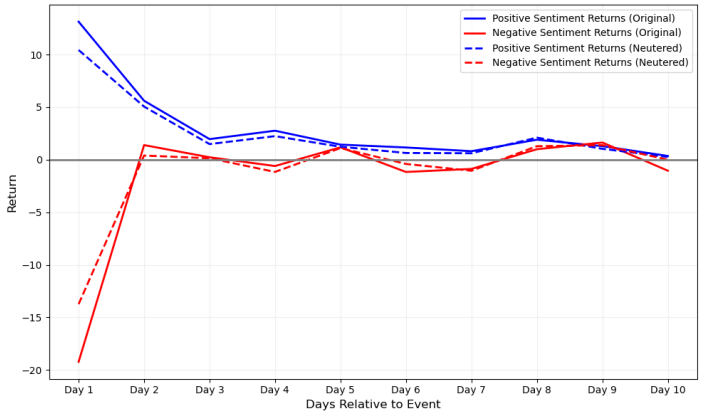
Figure 5: Event study returns by news sentiment



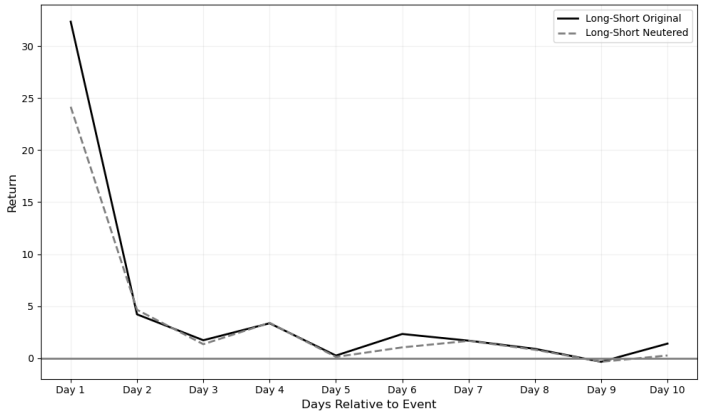
(a) Positive and negative



(b) Positive−negative



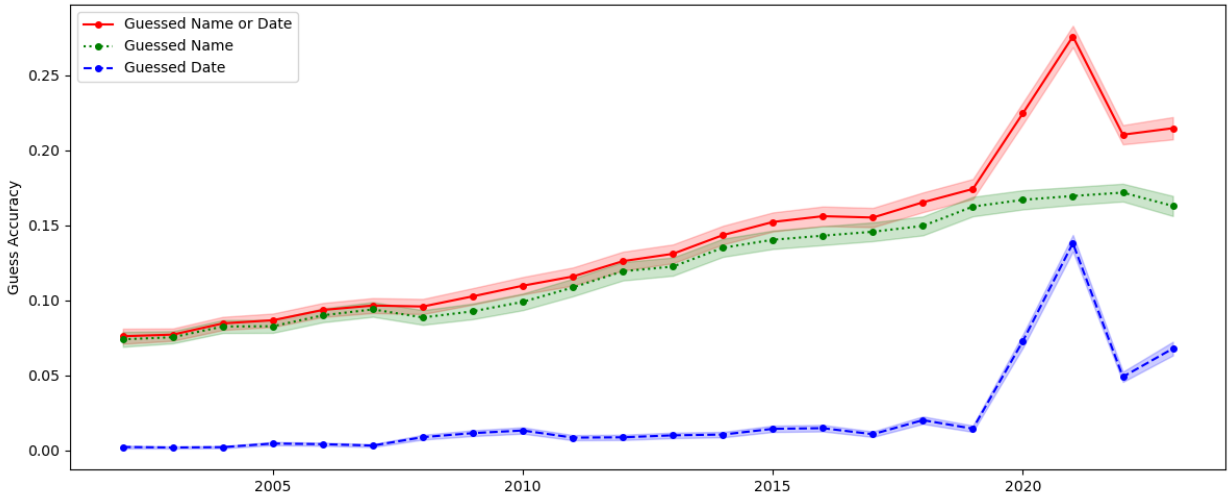
(c) Strong sentiment



(d) Strong sentiment long−short

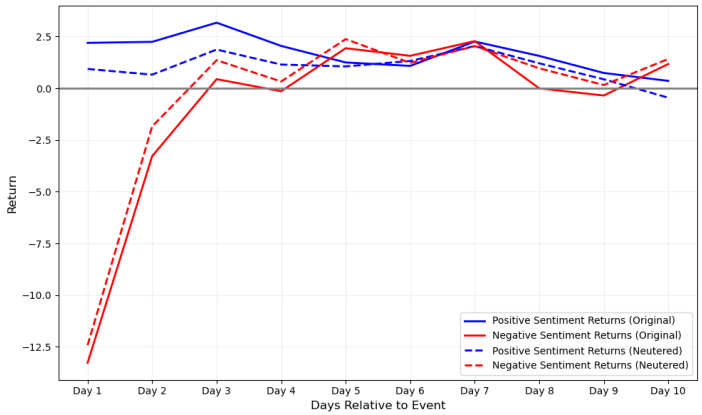
The figures plot average DGTW returns in event time by sentiment, which is extracted from DJNW news articles. The blue lines (in the top half of panels (a) and (c)) indicate the return of articles with positive sentiment on day 0, while the red lines (in the bottom half of the figures) correspond to 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.

Figure 6: Entity neutered MD&A guessing accuracy over time

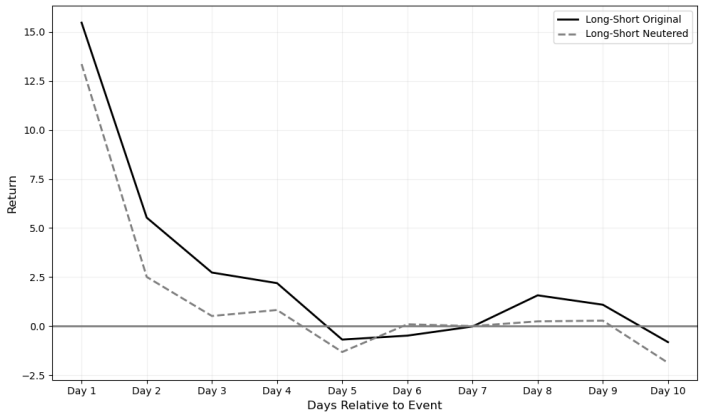


The figure shows guessing accuracy over time for MD&A text from 2002 to 2023.

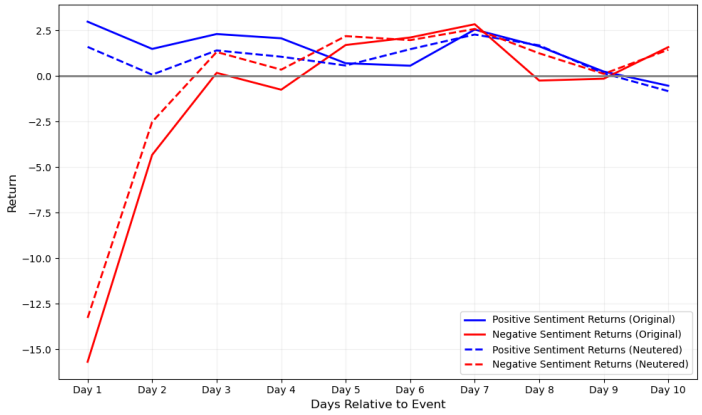
Figure 7: Event study returns by MD&A sentiment



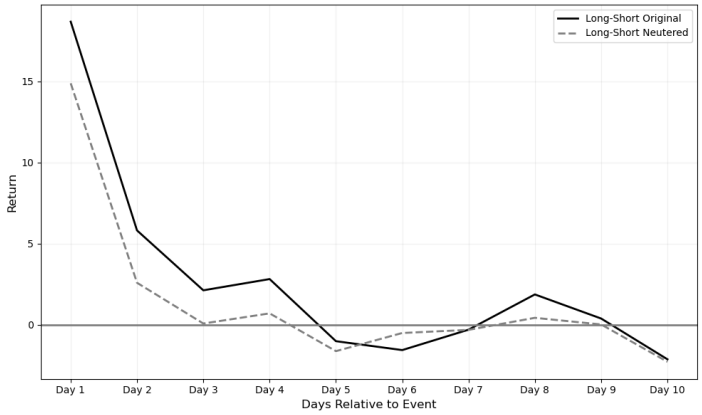
(a) All MD&A



(b) All MD&A long-short



(c) Strong sentiment



(d) Strong sentiment long-short

The figures plot average DGTW returns in event time by sentiment, which is extracted from Management's Discussion and Analysis (MD&A). 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 measured 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-4o-mini	unknown	6.81%	1,056,053
Meta	Llama 3.2	3B	5.6%	295,400
Google	Gemma 3	4B	6.7%	211,000

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

Table 2: Return predictability – DJNW articles

	Outcome: $DGTW_{i,t+1}$				$DGTW_{i,(t+2) \to (t+5)}$			
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	Raw	Neutered	Raw	Neutered	Raw	Neutered	Raw	Neutered
$\mathbb{1}$ Positive sentiment$_{i,t}$	7.57***	5.85***			8.22***	5.91***		
	[5.89]	[4.90]			[3.97]	[3.15]		
$\mathbb{1}$ Negative sentiment$_{i,t}$	-9.80***	-10.23***			-14.93***	-15.83***		
	[-5.91]	[-5.73]			[-5.86]	[-7.08]		
$\mathbb{1}$ Strong positive$_{i,t}$			20.43***	13.15***			24.15***	12.59***
			[9.10]	[7.86]			[6.20]	[4.57]
$\mathbb{1}$ Weak positive$_{i,t}$			4.54***	2.90**			4.49**	3.20*
			[3.55]	[2.40]			[2.18]	[1.66]
$\mathbb{1}$ Weak negative$_{i,t}$			-5.47***	-5.82***			-10.17***	-11.37***
			[-3.54]	[-3.26]			[-4.45]	[-5.21]
$\mathbb{1}$ Strong negative$_{i,t}$			-37.37***	-33.13***			-45.22***	-39.02***
			[-7.67]	[-8.08]			[-5.07]	[-5.23]
$DGTW_{i,t}$	0.03**	0.03**	0.03**	0.03**	-0.07***	-0.07***	-0.08***	-0.08***
	[2.24]	[2.27]	[2.08]	[2.15]	[-3.48]	[-3.47]	[-3.55]	[-3.51]
$DGTW_{i,t-1}$	-0.04*	-0.04*	-0.04*	-0.04*	-0.14***	-0.14***	-0.15***	-0.15***
	[-1.85]	[-1.84]	[-1.90]	[-1.88]	[-3.00]	[-3.00]	[-3.02]	[-3.01]
$DGTW_{i,t-2}$	-0.08***	-0.08***	-0.08***	-0.08***	-0.09**	-0.09**	-0.09**	-0.09**
	[-2.72]	[-2.72]	[-2.75]	[-2.74]	[-2.33]	[-2.33]	[-2.35]	[-2.34]
$DGTW_{i,(t-21) \to (t-3)}$	-0.01**	-0.01**	-0.02**	-0.02**	-0.01*	-0.01*	-0.01*	-0.01*
	[-2.43]	[-2.43]	[-2.51]	[-2.49]	[-1.80]	[-1.80]	[-1.87]	[-1.85]
$DGTW_{i,(t-252) \to (t-22)}$	0.00	0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00
	[0.67]	[0.72]	[0.09]	[0.31]	[-1.20]	[-1.17]	[-1.51]	[-1.35]
Log($BM_{i,t}$)	0.58	0.49	1.19	0.94	-1.66	-1.78	-0.95	-1.35
	[0.40]	[0.34]	[0.82]	[0.64]	[-0.63]	[-0.67]	[-0.36]	[-0.51]
Log($ME_{i,t}$)	-1.41**	-1.38*	-1.26*	-1.15	-1.57	-1.52	-1.42	-1.29
	[-1.96]	[-1.91]	[-1.75]	[-1.59]	[-1.42]	[-1.38]	[-1.28]	[-1.16]
Log($ST_{i,t}$)	0.89	0.86	1.51	1.42	8.74***	8.71***	9.39***	9.29***
	[0.42]	[0.41]	[0.72]	[0.67]	[2.74]	[2.74]	[2.89]	[2.87]
Constant	23.81***	24.00***	23.12***	21.86**	40.97***	41.24***	40.35***	38.98***
	[2.74]	[2.74]	[2.66]	[2.48]	[3.00]	[3.02]	[2.96]	[2.86]
# observations	858,088	858,088	858,088	858,088	858,073	858,073	858,073	858,073
# clusters (date)	2,219	2,219	2,219	2,219	2,219	2,219	2,219	2,219
Adj. R^2	0.012	0.012	0.013	0.013	0.016	0.016	0.016	0.016
RMSE	314.811	314.829	314.709	314.758	525.239	525.255	525.160	525.213

Sentiment is extracted via LLM as described in section 4. Strong (weak) sentiment is an indicator for articles with sentiment magnitude $\geq$ ($<$) 0.7, which is the median sentiment for both raw and neutered articles. Coefficients are in basis points. t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Table 3: Trading strategy returns using DJNW sentiment

	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

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. Robust t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Table 4: Return predictability – MD&A

	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$_{i,t}$	5.06*	2.79			-2.05	-3.77		
	[1.70]	[1.08]			[-0.37]	[-0.82]		
$\mathbb{1}$ Negative sentiment$_{i,t}$	-9.40***	-8.33***			-20.92***	-12.97***		
	[-3.01]	[-3.05]			[-3.63]	[-2.74]		
$\mathbb{1}$ Strong positive$_{i,t}$			8.35**	4.51			2.03	7.44
			[2.49]	[1.30]			[0.32]	[1.14]
$\mathbb{1}$ Weak positive$_{i,t}$			4.29	2.63			-3.11	-5.30
			[1.41]	[1.00]			[-0.56]	[-1.15]
$\mathbb{1}$ Weak negative$_{i,t}$			-6.28**	-6.18**			-18.51***	-13.00***
			[-2.02]	[-2.28]			[-3.16]	[-2.67]
$\mathbb{1}$ Strong negative$_{i,t}$			-19.68***	-17.10***			-29.01***	-13.32
			[-4.10]	[-3.44]			[-3.45]	[-1.61]
$DGTW_{i,t}$	-0.02***	-0.02***	-0.02***	-0.02***	-0.01*	-0.01*	-0.01*	-0.01*
	[-3.73]	[-3.69]	[-3.77]	[-3.71]	[-1.86]	[-1.78]	[-1.89]	[-1.78]
$DGTW_{i,t-1}$	-0.01**	-0.01**	-0.01**	-0.01**	-0.01*	-0.01*	-0.01*	-0.01*
	[-2.55]	[-2.51]	[-2.57]	[-2.52]	[-1.88]	[-1.83]	[-1.89]	[-1.83]
$DGTW_{i,t-2}$	-0.02***	-0.02***	-0.02***	-0.02***	-0.02**	-0.01**	-0.02**	-0.01**
	[-3.09]	[-3.07]	[-3.10]	[-3.08]	[-2.04]	[-2.02]	[-2.05]	[-2.02]
$DGTW_{i,(t-21) \to (t-3)}$	-0.00*	-0.00*	-0.00*	-0.00*	-0.01***	-0.01***	-0.01***	-0.01***
	[-1.71]	[-1.65]	[-1.75]	[-1.65]	[-2.89]	[-2.84]	[-2.90]	[-2.84]
$DGTW_{i,(t-252) \to (t-22)}$	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
	[-0.69]	[-0.47]	[-0.89]	[-0.56]	[-1.44]	[-1.22]	[-1.53]	[-1.24]
Log($BM_{i,t}$)	3.54**	3.47**	3.75***	3.53**	3.38	3.11	3.56	3.16
	[2.50]	[2.45]	[2.64]	[2.49]	[1.32]	[1.21]	[1.39]	[1.23]
Log($ME_{i,t}$)	2.22***	2.36***	2.00***	2.22***	-2.84***	-2.43**	-3.04***	-2.52**
	[3.84]	[4.09]	[3.49]	[3.89]	[-2.72]	[-2.34]	[-2.93]	[-2.45]
Log($ST_{i,t}$)	-0.98	-1.14	-0.68	-0.93	3.66**	3.33*	3.90**	3.33*
	[-1.17]	[-1.35]	[-0.81]	[-1.11]	[2.11]	[1.92]	[2.27]	[1.95]
Constant	-28.94***	-30.56***	-25.57***	-28.49***	59.77***	50.49***	62.80***	51.80***
	[-3.42]	[-3.69]	[-3.04]	[-3.46]	[3.94]	[3.48]	[4.18]	[3.61]
# observations	227,963	227,963	227,963	227,963	227,745	227,745	227,745	227,745
# clusters (date)	5,280	5,280	5,280	5,280	5,280	5,280	5,280	5,280
Adj. R^2	0.002	0.002	0.002	0.002	0.001	0.001	0.001	0.001
RMSE	408.877	408.899	408.857	408.889	721.784	721.825	721.779	721.823

Strong (weak) sentiment is an indicator for articles with sentiment magnitude $\geq$ ($<$) 0.7, which is the median sentiment for both raw and neutered articles. Coefficients are in basis points. t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Table 5: Return predictability – MD&A with continuous sentiment

	Outcome: $DGTW_{i,t+1}$				$DGTW_{i,(t+2) \rightarrow (t+5)}$			
	(1) Raw	(2) Neutered	(3) Raw	(4) Neutered	(5) Raw	(6) Neutered	(7) Raw	(8) Neutered
Continuous $sentiment_{i,t}$	8.91***	6.93***			11.33***	5.71**		
	[6.28]	[4.74]			[4.59]	[2.26]		
Δ $sentiment_{i,t}$	6.79***	6.67***			7.05**	5.06*		
	[4.03]	[4.14]			[2.49]	[1.68]		
$\mathbb{1}$ Positive to $negative_{i,t}$			-19.81***	-17.62***			-21.65***	-10.77*
			[-6.46]	[-5.69]			[-3.95]	[-1.76]
$\mathbb{1}$ Positive to $neither_{i,t}$			0.11	-1.78			3.57	8.41
			[0.02]	[-0.44]			[0.34]	[1.15]
$\mathbb{1}$ Negative to $negative_{i,t}$			-10.56***	-8.22***			-13.38***	-4.29
			[-4.39]	[-3.24]			[-3.15]	[-0.96]
$\mathbb{1}$ Negative to $positive_{i,t}$			7.92**	7.59**			11.49**	7.33
			[2.45]	[2.28]			[2.22]	[1.38]
$\mathbb{1}$ Negative to $neither_{i,t}$			-14.52**	-7.24			5.16	0.48
			[-1.98]	[-1.34]			[0.41]	[0.05]
$\mathbb{1}$ Neither to $neither_{i,t}$			-0.67	0.03			6.04	7.91
			[-0.18]	[0.01]			[0.82]	[1.25]
$\mathbb{1}$ Neither to $positive_{i,t}$			1.08	-1.41			4.71	5.15
			[0.44]	[-0.58]			[1.17]	[1.27]
$\mathbb{1}$ Neither to $negative_{i,t}$			-12.39***	-9.92***			-15.54***	-8.17
			[-3.99]	[-3.13]			[-2.71]	[-1.48]
$DGTW_{i,t}$	-0.02***	-0.02***	-0.02***	-0.02***	-0.01*	-0.01*	-0.01*	-0.01*
	[-4.14]	[-4.09]	[-3.76]	[-3.71]	[-1.77]	[-1.68]	[-1.88]	[-1.80]
$DGTW_{i,t-1}$	-0.01***	-0.01**	-0.01**	-0.01**	-0.01*	-0.01*	-0.01*	-0.01*
	[-2.60]	[-2.55]	[-2.57]	[-2.53]	[-1.74]	[-1.68]	[-1.89]	[-1.83]
$DGTW_{i,t-2}$	-0.02***	-0.02***	-0.02***	-0.02***	-0.01*	-0.01*	-0.02**	-0.01**
	[-3.05]	[-3.04]	[-3.09]	[-3.08]	[-1.85]	[-1.82]	[-2.05]	[-2.02]
$DGTW_{i,(t-21) \rightarrow (t-3)}$	-0.00*	-0.00*	-0.00*	-0.00*	-0.01***	-0.01**	-0.01***	-0.01***
	[-1.88]	[-1.80]	[-1.75]	[-1.68]	[-2.62]	[-2.56]	[-2.91]	[-2.85]
$DGTW_{i,(t-252) \rightarrow (t-22)}$	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
	[-0.69]	[-0.50]	[-0.62]	[-0.44]	[-1.43]	[-1.22]	[-1.39]	[-1.20]
$Log(BM_{i,t})$	3.92***	3.84***	3.44**	3.36**	3.37	3.12	3.23	3.01
	[2.66]	[2.61]	[2.42]	[2.37]	[1.27]	[1.18]	[1.26]	[1.17]
$Log(ME_{i,t})$	2.28***	2.43***	2.34***	2.45***	-2.81***	-2.42**	-2.66**	-2.32**
	[3.82]	[4.07]	[4.04]	[4.26]	[-2.60]	[-2.25]	[-2.54]	[-2.23]
$Log(ST_{i,t})$	-0.96	-1.11	-1.07	-1.21	3.63**	3.31*	3.56**	3.29*
	[-1.12]	[-1.29]	[-1.26]	[-1.43]	[2.05]	[1.87]	[2.05]	[1.90]
Constant	-31.73***	-33.84***	-27.39***	-30.03***	47.74***	42.04***	51.80***	42.55***
	[-3.91]	[-4.20]	[-3.42]	[-3.76]	[3.30]	[2.92]	[3.61]	[2.98]
# observations	213,006	213,006	227,963	227,963	212,799	212,799	227,745	227,745
# clusters (date)	5,275	5,275	5,280	5,280	5,275	5,275	5,280	5,280
Adj. R^2	0.002	0.002	0.002	0.002	0.001	0.001	0.001	0.001
RMSE	413.036	413.062	408.869	408.893	726.233	726.279	721.785	721.829

Continuous sentiment $\in [-1, 1]$. Δ sentiment is current quarter minus preceding quarter sentiment. Positive to positive is the omitted baseline for columns (3),(4),(7) and (8). Coefficients are in basis points. t-statistics in brackets. * $p < .1$; ** $p < .05$; *** $p < .01$.

Internet Appendix:

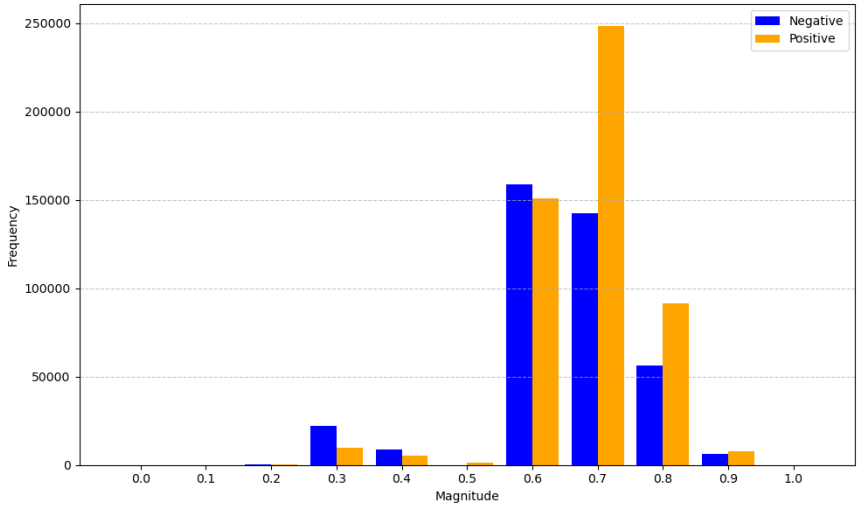
Entity Neutering

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

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

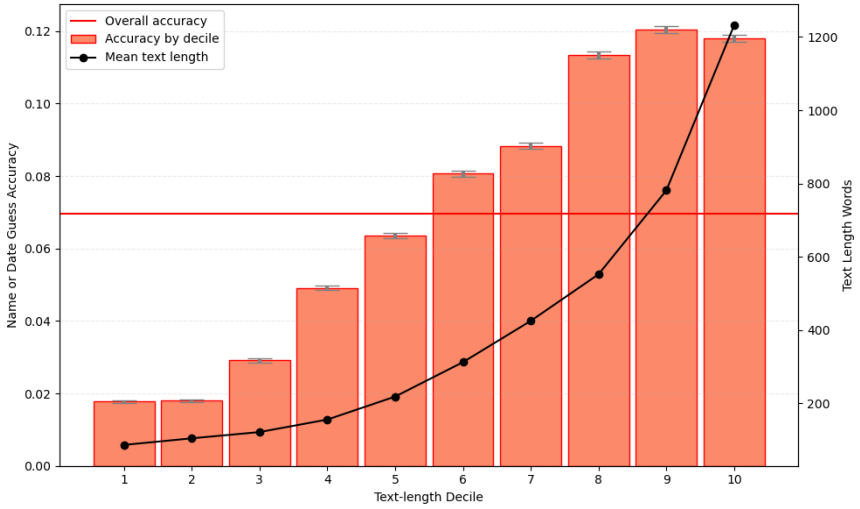
A. News Articles

Figure A1: Distribution of sentiment magnitude for DJNW articles



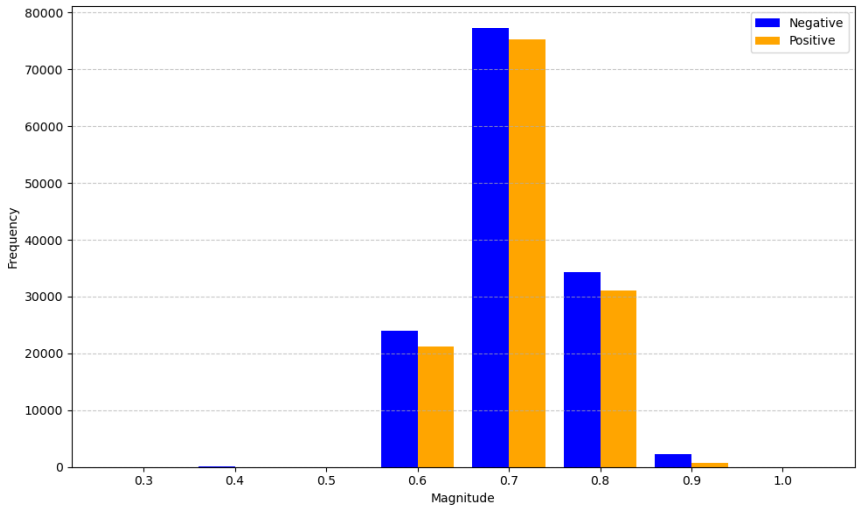
The figure plots the distribution of the magnitude of sentiment generated by the LLM from DJNW articles. Sentiment magnitude is on the horizontal axis and ranges from 0 to 1, where higher values represent stronger sentiment. For each horizontal-axis value, the left-hand bar (blue) represents negative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

Figure A2: Firm or date identification rate by DJNW article length



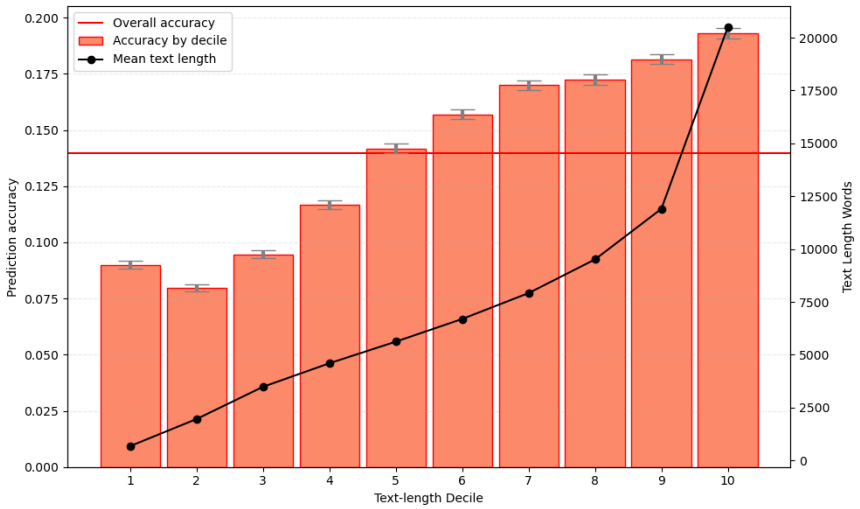
This figure plots the identification rate of firm or date as a function of DJNW article length (measured in characters). The vertical axis represents the share of all articles for which LLM was able to correctly identify the firm or date. Average accuracy is 7% and is plotted with a red horizontal line.

Figure A3: Distribution of sentiment magnitude for MD&A



The figure plots the distribution of the magnitude of sentiment generated by the LLM from MD&A. 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 A4: Firm or date identification rate by MD&A length



This figure plots the identification rate of firm or date as a function of MD&A length (measured in characters). The vertical axis represents the share of all MD&As for which LLM was able to correctly identify the firm or date. Average accuracy is 14% and is plotted with a red horizontal line.

Table A1: DJNW news sample filtering steps

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
Un-identifiable articles	984,085	71,968
Control variables exist	858,088	125,997

This table shows the number of remaining observations at each step of the sample filtering process. All but the final two steps follow Ke, Kelly, and Xiu (2019). "Un-identifiable articles" refers to articles for which 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 guess that is within a ±7-day window of the true date). "Control variables" refer to those used in Table 2. We perform the entity neutering procedure on 1,056,053 DJNW news articles from January 3, 2000 through January 2, 2009.

Table A2: Identification rates for DJNW news articles

Data	Guess name	Guess date	Guess date or name	Guess date and name
Small	3.39%	0.15%	3.51%	0.02%
Medium	5.82%	0.19%	5.98%	0.03%
Large	12.23%	0.21%	12.39%	0.05%
Value	5.90%	0.22%	6.07%	0.04%
Growth	7.70%	0.15%	7.82%	0.02%
All	6.81%	0.19%	6.96%	0.04%

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), measured yearly. 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), measured yearly.

Table A3: MD&A sample filtering steps

Data Stage	Obs.	Obs. lost	% Lost	Unique firms
Notre Dame 10-Q & 10-K repo.	641,527			
MD&A extracted	506,080	-135,447	-21.11%	
CIK link to PERMNO exists	292,231	-213,849	-42.26%	
DGTW returns exist	283,660	-8,571	2.93%	
Un-identifiable articles	244,080	39,613	13.96%	
Control variables exist	227,963	-16,084	6.59%	10,163

This table shows the number of remaining observations at each step of the sample filtering process for the MD&A dataset. To extract an MD&A section from an SEC filing in the Notre Dame repository, the text must have a valid CIK, a valid DateTime stamp, and an MD&A section of over 500 characters. The majority of observations are excluded at this stage are older filings lacking a valid DateTime stamp. "Un-identifiable articles" are described in the note to Appendix Table A1. "Control variables" refer to those used in Table 4. We perform the entity neutering procedure on 283,660 MD&A sections from 19 March, 2002 through September 30, 2023.

Table A4: Identification rates for MD&A text

Data	Guess Name	Guess Date	Guess Date or Name	Guess Date and Name
10Q MD&A	12.08%	3.0%	14.45%	0.63%
10K MD&A	13.30%	0.31%	13.53%	0.08%
Small	5.71%	1.14%	6.72%	0.12%
Medium	9.74%	2.49%	11.85%	0.37%
Large	23.29%	3.22%	25.43%	1.09%
Value	8.62%	2.17%	10.47%	0.31%
Growth	15.62%	2.38%	17.36%	0.64%
All MD&A	12.12%	2.33%	13.96%	0.49%

The table presents the rate of correct identification (guesses) of MD&A firm names and dates. The first two rows represent the guessing rates for MD&As extracted from 10-Q and 10-K filings respectively. The next three rows show guessing rates for the bottom quartile, middle two quartiles, and top quartiles of firm size respectively. Finally, the unconditional identification rate is presented in the final row.

B. Prompts

B.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 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 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 to make an educated guess about its identity. Stock tickers are identifiers, are usually four or fewer capitalized letters (consider TIKR as a stand-in for an arbitrary ticker), and can 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 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.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**"*

B.4 Neutering prompt split into 4 component parts

Below we present the neutering prompt split into its four component parts. In Figure 3 we show results from neutering using (i) only the first part, (ii) both parts 1 and 2, (iii) parts 1–3, and (iv) the full prompt. In each case we append the following to the prompt to ensure the output's format: *After you read and ANONYMIZE the text, you should output the anonymized text and nothing else.*

Prompt part 1:

1. *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 use the text to identify the company, the industry the company operates in or the date.*

Prompt part 2:

2. *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.*

Prompt part 3:

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

Prompt part 4:

4. *Stock tickers are identifiers, are usually four or fewer capitalized letters (consider TIKR as a stand-in for an arbitrary ticker), and can 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.