Exploring the Use of Large Language Models for Layoff Prediction

Abdullah Garra
Tel-Aviv University
206940777
Group 35
abdullahg@mail.tau.ac.il

Obayda Haj-Yehia
Tel-Aviv University
322674466
Group 35
obaidah@mail.tau.ac.il

Abstract

In recent years, many companies have announced layoffs, making it increasingly important to develop tools that can analyze and predict such events. Initially, we aimed to address the binary classification problem of determining whether a layoff will occur or not. However, this proved to be a challenging task due to the limited data available and difficulties in establishing a reliable baseline for comparison.

We shifted our attention to processing and integrating news data alongside technical financial metrics to tackle the percentage prediction task (i.e., predicting the percentage of employees being laid off, given that a layoff has occurred). Addressing the challenge of managing large volumes of news data, we experimented with filtering techniques, time window selection (7, 15, 30, and 90 days), and variations of FinBERT embeddings.

Our findings show that a hybrid approach combining technical and textual data can improve prediction accuracy. The best-performing model, which used average embeddings of unfiltered summaries, achieved a Test Mean Absolute Error (MAE) of 8.74, outperforming other configurations. This highlights the importance of retaining broader contextual information and suggests that advanced methods for integrating diverse data sources hold promise for further improving layoff prediction models.

1 Introduction

Layoffs are a significant event, impacting individuals, companies, and even the economy. For employees, job losses bring financial strain and uncertainty. For companies, layoffs can hurt reputation, disrupt productivity, and reduce morale. On a larger scale, high layoff numbers often signal economic downturns, making accurate predictions valuable for stakeholders like policymakers and investors.

Our literature review indicates that previous attempts for layoff prediction methods focus on structured data, such as financial indicators and company size. While useful, these data points might not capture all changes in economic sentiment or company-specific shifts. Unstructured data sources, like news articles, can provide real-time insights into market sentiment, but analyzing this data alongside structured indicators requires sophisticated techniques.

This is where Large Language Models (LLMs) like BERT and FinBERT come in. LLMs, trained on vast amounts of text, are effective at understanding language nuances, context, and sentiment, making them strong candidates for tasks that involve unstructured data. By analyzing news articles or financial reports, LLMs can detect subtle signals, like shifts in tone or specific company developments, that may hint at future layoffs and their volume. We hypothesized that when these insights are combined with traditional financial metrics, they can provide a more robust and nuanced layoff prediction model.

In this work, we explore a hybrid approach that combines structured financial data with LLM-powered analysis from news sources. This dual approach is an attempt to predict the potential scale of layoffs.

2 Related Work

Research on layoff prediction as a binary classification or percentage estimation task is relatively limited. However, several related areas provide useful insights, especially in the integration of structured and unstructured data for economic forecasting. Existing work on layoff prediction, stock price prediction, and financial sentiment analysis contributes foundational ideas that inform our approach.

2.1 Layoff Prediction Studies

Prior attempts at layoff prediction have largely focused on either predicting the likelihood of layoffs within the tech sector or estimating the scale of layoffs based on technical indicators. One example is an open-source project available on GitHub (Katulevskiy, 2023), which explores machine learning models for predicting tech layoffs by using structured financial data and technical indicators as primary inputs. This project provides a useful starting point for layoff-scale prediction but does not leverage recent advances in Large Language Models (LLMs) for handling unstructured data.

2.2 Stock Price Prediction with LLMs and Technical Data

A related line of research explores the combination of technical data with LLMs for stock price prediction. A project by Stanford students, for example, developed a model that integrates structured financial indicators with news sentiment to forecast stock prices (Naftchi-Ardebili and Singh, 2024). Although this study focuses on stock price prediction rather than layoffs, it provides valuable insights into how technical data can be combined with unstructured news text for financial forecasting. The approach they used—leveraging financial sentiment signals from news articles with technical indicators—informs our methodology of using LLMs to handle unstructured data alongside structured financial metrics for layoff prediction.

2.3 Financial Sentiment Analysis and LLMs

Recent work on financial sentiment analysis using LLMs has demonstrated the value of unstructured data for economic prediction tasks. The FinBERT model, a BERT-based LLM fine-tuned specifically on financial texts, has shown promising results in capturing nuanced sentiment from financial news articles (Araci, 2019). This model's ability to extract relevant sentiment indicators from news and financial reports makes it particularly suited for applications in layoff prediction, where market sentiment and company-specific news can provide early warnings of potential layoffs. In our study, we use variations of a FinBERT component to process news articles related to target companies, creating embeddings that can serve as inputs to our hybrid layoff prediction model.

Another language model, aMPnet, has shown strong performance in capturing semantic meaning from short texts (Song et al., 2020). We used aMPnet embeddings in our initial filtering stage to pre-select relevant news articles before passing them to FinBERT for fine-grained sentiment analysis. This was an attempt to filter and refine our data effectively, ensuring that only highly relevant news articles contribute to our model's predictions.

2.4 Stock Price Prediction with LSTMs

In stock price prediction, where structured financial data is combined with sentiment data, recurrent neural networks (RNNs) like LSTM have proven effective at capturing temporal dependencies (Adil Moghar, 2020). Studies have shown that LSTM models, which can process sequential data, outperform other traditional architectures in tasks that require an understanding of time series patterns, such as stock price forecasting. This success in stock price prediction points to the potential of LSTM for layoff prediction as well, as the financial metrics leading up to a layoff often follow temporal patterns that can be captured by such models. Inspired by this, we included an LSTM in our model experiments to capture sequential dependencies in financial data and understand how they correlate with layoff events.

3 Data Collection

3.1 Technical Data

Our technical data is based on structured financial indicators similar to those used in previous studies. Specifically, we included NASDAQ stock metrics, such as the difference in opening prices between 90 days prior and the date of the layoff. These metrics serve as baseline indicators of overall market conditions, which may correlate with layoffs across the tech sector. Additionally, we initially experimented with an alternative approach focused on per-company stock data instead of relying solely on NASDAQ trends. This approach aimed to capture more granular, company-specific signals potentially linked to layoff risk.

To gather per-company data, we used the Financial Modeling Prep (FMP) API, which provided access to historical stock prices, company financials, and other relevant metrics. However, after extensive experimentation, we decided to drop this per-company approach due to significant limitations in the data. The availability of historical data varied widely between companies, with many having only sparse records. Attempting to work with only well-documented companies resulted in a dataset too small for effective model training. On the other hand, including more companies required extensive filling of missing data, which posed significant challenges. Filling in missing values for time periods before a company's existence or was not realistic and risked introducing noise into the dataset. Additionally, oversampling to address the sparsity of per-company data would have broken the temporal dependencies inherent in the data, distorting the model's understanding of time-sensitive patterns and undermining its ability to generalize.

Ultimately, we found that neither scenario—working with a small, sparse dataset nor attempting to fill and oversample data—produced a model that performed well. This led us to abandon the per-company approach in favor of a broader, aggregated dataset better suited to the task.

The final technical dataset includes the following features:

- **`Date_layoffs`**: The date of each recorded layoff event.
- **`Company_Size_before_Layoffs`**: The size of the company prior to layoffs, used to contextualize the layoff scale.
- **`Money_Raised_in_millions`**: Funds raised by the company, as a measure of financial stability.
- **`Stock_delta`**: The stock price difference over the 90-day period leading up to the layoff event.
- One-hot encoded features for **`Industry`**, **`Stage`**, and **`Region`**: These capture categorical information relevant to layoff risk.

These structured indicators were standardized to allow for integration with the unstructured data features extracted from news articles. Chronological splitting was applied to prevent data leakage, ensuring that each training and test split represented a distinct time period.

3.2 Layoff Dataset

For the layoff data, we used the publicly available dataset the tech layoffs dataset, hosted on Kaggle (Herold, 2024). This dataset compiles all reported layoffs in the tech industry from 2020 to June 30, 2024, and originates from the website layoffs.fyi (lay), which tracks layoffs in real time. The dataset has also been referenced in a previous open-source project on GitHub (Katulevskiy, 2023), providing a historical record of layoffs across various companies and allowing us to model layoff patterns over time.

We conducted further data analysis on this dataset to better understand trends and characteristics that might signal layoffs. Specifically, we explored factors such as company size, industry, and funding history. To assess the value of additional data now available, we re-ran the models from the GitHub repository, comparing their performance with the expanded more updated dataset.

3.3 News Data Collection

For our unstructured data, we used news articles from the Common Crawl News archive, specifically the archived version provided by the Stanford-Oval community (OVAL, 2024) dataset on Hugging Face. This dataset archives a collection of news articles' clean text, following the Datatrove (Penedo et al., 2024) methodology for filtering news. Various Common Crawl News datasets (e.g. (Blagojevic, 2024)) exist on HuggingFace, but Stanford's stood out since Datatrove uses Trafilatura for text extraction. Emperical evidence shows trafilatura performed best out of all extractors out there in the context of news articles. More on this in the next section.

To focus on the most relevant articles for our prediction task, we filtered the dataset by domain, selecting only those from Yahoo Finance (financial.yahoo.com), as Yahoo is ranked as the number one news website according to (Scheitle et al., 2019) using their API (API) to conclude this insight. Due to resource constraints, we limited our data processing to this single source, balancing quality with manageable data size. This filtering yielded approximately 120,000 articles spanning from January 2020 to June 2024. These articles were then stored in a MySQL database provided by the university.

3.3.1 Text Extraction from HTML - Benchmark

To extract text from HTML, we compared several popular tools to ensure we selected the one that provided the highest accuracy and consistency in capturing news content. Our evaluation followed the methodology from the Fundus benchmark (Dallabetta et al., 2024), which compares various HTML-to-text extraction tools. We created a list of the top 26 English news websites from the Tranco list, selecting those that appeared in the top 1000 websites (Scheitle et al., 2019). A full list of these websites and their rankings can be found in Appendix A.

3.3.2 Experiment Setup

For each of the top 26 news sites, we selected two articles available in the Common Crawl News (CC-News) archive and accessible to us (i.e, not paywalled). We manually extracted the text from each article to create a ground truth. We then compared each tool's extraction output against the ground truth using the Word Error Rate (WER), a standard metric for evaluating text extraction quality. The tools tested included:

- Trafilatura
- NewsPlease
- Boilerpipe
- Boilernet
- BTE
- Justext

The Word Error Rate (WER) averages for each tool are presented in Table 1 below.

Tool	WER Average
Trafilatura	0.292
NewsPlease	0.279
Boilerpipe	0.488
Boilernet	0.717
BTE	1.111
Justext	0.509

Table 1: WER Averages for Different HTML-to-Text Extraction Tools.

While both NewsPlease and Trafilatura performed well, achieving lower WER scores than other tools, it was a close competition between them. To gain further insight, we plotted the WER distribution for these two tools (see Figure 1) using box plots.

The box plot shows that although NewsPlease and Trafilatura had closer average WER scores, Trafilatura displayed less variance and fewer outliers in its performance across different sites. NewsPlease had a wider spread, indicating that while it sometimes achieved very low WER scores, it was less consistent and occasionally performed poorly on certain articles. In contrast, Trafilatura maintained a more stable extraction quality across the diverse set of websites. This stability and lower variance made Trafilatura a more reliable choice for our dataset.

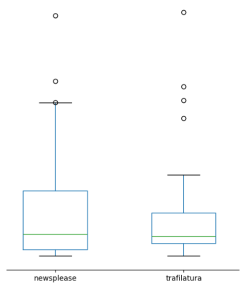

Figure 1: BoxPlot for Trafilatura and newsplease - WER performance

Based on this empirical evidence, we favored stanford's Huggingface dataset since it uses it for text extraction, besides pre and post extraction filtering, ensuring high-quality and consistent content extraction from HTML across various news domains.

4 Methodology

4.1 Data Pre-processing

For each layoff report, we aimed to create a feature representing the textual content from news articles within different time windows leading up to the layoff event. We experimented with 7-day, 15-day, 30-day, and 90-day windows to compare how each period's content affects the model's performance.

To capture relevant information from the articles, we aggregated news data for each layoff report, grouping them into various time windows (7, 15, 30, and 90 days prior to each layoff date). However, due to tokenization limits (with a maximum length of 512 tokens for FinBERT), we needed to carefully aggregate the texts to better fit within these constraints.

Our first attempt to deal with this challenge was summarizing each article using ChatGPT-4o's API to condense the text while retaining key information. These summaries were saved, along with the original publication dates, in a separate table within our MySQL database. Concatenating the article summaries within each of the specified windows to form a single text input for each layoff report. While concatenation reduced the length of the resulting text, the concatenated summaries still often exceeded the desired token limit. For example, tokenizing 30-day summaries with Longformer's tokenizer for a sample of 200 dates, yielded the statistics in Table 2

As shown, even with summaries, the concatenated text often significantly exceeded the target

Statistic	30-day Summaries (Token Count)
Count	200
Mean	937.26
Minimum	3
Maximum	4716

Table 2: Token Count Statistics for 30-Day Summaries.

token range for FinBERT, with extreme variability in length. For instance, the maximum token count reached over 4,700 for some entries, which far exceeds FinBERT's processing capabilities.

Given these limitations, we explored alternative approaches to better represent the articles. One approach involved applying a filtering process (details forthcoming) to reduce input length while preserving the most informative content within each window. Another approach utilized pre-calculated embeddings of all summaries within a given time window. This embedding-based method, while resource- and time-intensive, achieved the best performance. Notably, it required days to compute the embeddings—significantly more time than was needed to generate the summaries using ChatGPT's API or to apply the filtering process.[1]

4.2 Summary Filtering and Embedding

We used all-mpnet-base-v2 (Sentence-Transformers, 2020) from (Song et al., 2020) as our embedding model due to its decent performance in Semantic Textual Similarity (STS) tasks, where it ranked 8th in the MTEB benchmark. And so it is relatively effective in capturing semantic relationships, while also being manageable within our system's resource constraints. Furthermore, all-mpnet-base-v2 is highly popular, with over 366 million downloads on Hugging Face last month, indicating its widespread use and reliability in real-world applications.

4.2.1 Summary Filtering and Ranking Pipeline

We implemented a pipeline that processes article summaries from our database for each layoff date. For each layoff, we defined 7-day, 15-day, 30-day, and 90-day windows, where we extracted and embedded the article summaries using our embedding model. To assess the relevance of these summaries, we used a set of 15 reference sentences related to layoffs, stock, and macroeconomics (See Appendix B). The article summaries within each window were ranked by their average cosine-similarity to these reference sentences.

We analyzed the similarity scores to determine a threshold that would capture "highly relevant" articles without missing key information. Based on this analysis, we decided to take a minimum of the top 5 articles for each layoff date. Using a progressive method, we continued adding articles as long as the gap between the cosine similarity scores of two consecutively ranked articles remained below 0.035, ensuring the inclusion of as many relevant articles as possible.

4.2.2 Choosing the Threshold

We conducted an experiment focusing on a 30-day window of summaries for 100 layoff dates. Each article within this 30-day window had already been embedded using the all-mpnet model, with an average of 764 articles per layoff date and a maximum of 1002. For each layoff date, we ranked the summaries by their cosine similarity to the reference sentences and analyzed the largest gap in similarity values among the top-ranked articles. In our analysis, we examined the top ranked articles for each layoff date. We specifically looked at two aspects:

- The index at which the largest gap (or drop) in similarity scores occurred, as this gap indicates a potential point where relevance sharply decreases.
- The value of this largest similarity gap.

We observed that the largest gap in similarity typically appeared early in the rankings 1 and 2 (See Figure 2). This distribution suggests that the first few articles are generally the most relevant, with larger drops in similarity occurring soon after.

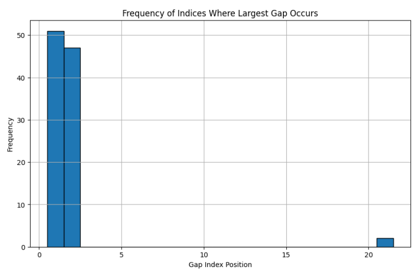

Figure 2: Largest Gap Index Frequency

Regarding the gap values, the statistics of the largest gaps are visualized in the boxplot 3 with the following values:

[1]A GitHub repository containing the code and partial implementation of our methods is available at: GitHub

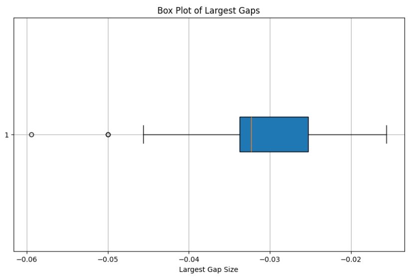

Figure 3: Largest Gaps boxplot distribution

- Mean: -0.0307
- Median: -0.0323
- Standard Deviation: 0.0077
- Minimum: -0.0595
- Maximum: -0.0156

Based on this analysis we set the gap threshold at 0.035 to allow a bit more flexibility in capturing relevant articles, as our resources could support a slightly larger selection. This threshold aligns closely with the observed median gap (0.0323) and is slightly above the average gap (0.0307), falling within approximately one standard deviation (0.0077) from the mean.

4.3 Baseline Model and Evaluation Metric

We started by running the models from the original GitHub repository and added a bidirectional LSTM to their lineup. This BiLSTM model, with layers of 1024 and 512 units followed by dense layers of 256, 128, and 64 units, outperformed their models and became our baseline. Unlike their approach, which used Mean Squared Error (MSE) as the evaluation metric, we used the Mean Absolute Error (MAE). Given that we rely on generic metrics, such as the NASDAQ index and general financial news (rather than per-company data), MAE is more robust against outliers and better captures the overall prediction quality. Its robustness ensures that individual outliers do not disproportionately affect the evaluation, offering a clearer measure of the model's average performance. We trained the model for 15 epochs, as running it longer led to overfitting. Simpler architectures did not perform well even with extended training. This model achieved an MAE of 16.12 on the test set.

4.4 Models

The general architecture of the models we tested combines a BiLSTM, which processes technical data, each time with a different variation of FinBERT, which handles news-related data. The BiLSTM is designed to capture temporal patterns in the technical data, while FinBERT encodes textual information from the news. The outputs of both components are concatenated and passed through a regression head to predict layoff percentages.

For the FinBERT component, we tested three variations:

1. Filtered Summaries (Not Fine-Tuned): FinBERT embeddings were generated from filtered summaries, and the model was not fine-tuned. These embeddings were not frozen during the training of the whole model, allowing them to be updated as part of the overall training process.

2. Filtered Summaries (Fine-Tuned and Frozen): FinBERT was fine-tuned on the filtered summaries in a separate step and frozen during the training of the combined model.

3. All Summaries (Average Embedding): In this configuration, we did not train on the filtered summaries. Instead, each summary was tokenized and passed through FinBERT. The [CLS] token from the FinBERT output was extracted as the embedding for each summary. We then computed the average of these embeddings across all summaries within the time window to represent the unfiltered textual data.

Each of the first two variations was tested with 7-day, 15-day, 30-day, and 90-day time windows. The third variation was calculated only with a 90-day window in an attempt to get the maximum amount of context.

The outputs of the BiLSTM's projection layer and the FinBERT embeddings are concatenated to form a unified representation. This combined representation is then passed through a series of dense layers. The first layer reduces the concatenated input dimension (128 from the BiLSTM and 768 from FinBERT) to 128 units, followed by another dense layer with 64 units, both utilizing ReLU activations and dropout for regularization. The final layer outputs a single value corresponding to the

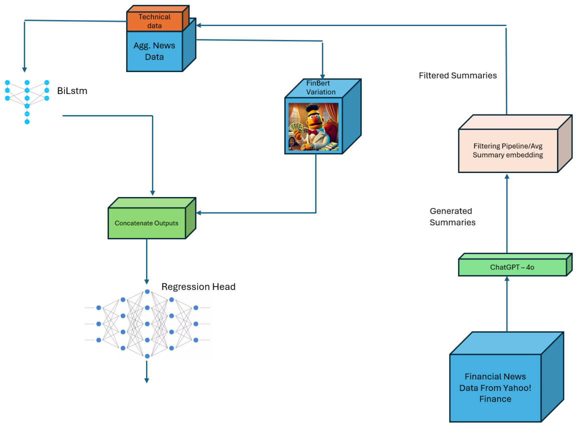

Figure 4: Experiment Architecture

predicted layoff percentage, making this a regression model.

This architecture, illustrated in Figure 4, is designed to leverage both structured and unstructured data combining insights from technical and news-based sources into a predictive framework.

5 Results

Table 3 presents the test Mean Absolute Error (MAE) for the baseline (BiLSTM only) model and different variations of the FinBERT-based models combined with BiLSTM across different time windows.

The baseline BiLSTM model, trained only on technical data, performs the worst with an MAE of 16.12, meaning leveraging the textual data, though a small portion of the sumamries, improved the model's ability to make predictions, thus highlighting the benefits of combining technical and textual features, with shorter windows like 15 days offering a balance between relevance and context.

The average embeddings model, which used the mean of all summaries' embeddings within a time window, achieved the best performance, with a Test MAE of 8.74 on the 90-day window.

The results indicate that the finetuned and frozen FinBERT consistently outperforms the non-finetuned (free) version, achieving the lowest MAE of 9.91 on the 15-day window. In comparison, the non-finetuned (free) model performs best on the same window with an MAE of 10.80 but shows a significant drop in accuracy for longer windows, such as 90 days (MAE: 14.88). This suggests that longer time windows introduce noise from less relevant articles, particularly when FinBERT is not finetuned.

These results indicate that our models can predict the percentage of layoffs a company is likely to experience with an average error of below 9%.

Model Configuration	**Time Window**	**Test MAE**
BiLSTM	N/A	16.12
Free FinBERT	7 Days	11.83
	15 Days	10.80
	30 Days	11.1
	90 Days	14.88
Finetuned & Frozen	7 Days	10.5
	15 Days	9.91
	30 Days	10.30
	90 Days	10.27
Avg. Embeddings	90 Days	8.74

Table 3: Test MAE for Combined BiLSTM and FinBERT Models Across Time Windows.

6 Conclusions

This study investigated methods for predicting the percentage of layoffs in companies by integrating structured technical data with unstructured news data. Among the tested approaches, the average embeddings model, which calculated the average embeddings of the contextualized representation of FinBERT using unfiltered summaries, achieved the best performance with a Test MAE of 8.74. This suggests that retaining a broader context, rather than filtering articles, allows the model to capture more comprehensive patterns. However, the computational cost of this method underscores the need for scalable solutions when working with larger datasets.

The finetuned and frozen FinBERT model performed well on shorter time windows, with the best Test MAE of 9.91 for the 15-day window. This highlights the effectiveness of domain-adapted embeddings and the importance of selecting an appropriate time window to balance context and noise. In contrast, the non-finetuned FinBERT and the baseline BiLSTM struggled to achieve comparable accuracy. For the non-finetuned FinBERT, the issue may lie in its need for significantly more training steps and data to better adapt, as its learning process and weight updates is influenced by both the technical data and the BiLSTM.

While challenges remain, including computational efficiency and data availability, this study demonstrates the potential of hybrid approaches for tackling layoff prediction tasks. The findings are an attempt to provide a foundation for further exploration into leveraging textual and financial data for predictive modeling in this underexplored area.

7 Limitation & Future work

A key limitation of our study is the reliance on generic financial metrics and aggregated news data rather than company-specific data. While this approach provides a broad understanding of layoff trends, it dilutes the model's ability to make precise predictions for individual companies. Incorporating company-specific data could significantly enhance prediction accuracy. Platforms like the Financial Modeling Prep (FMP) API (Prep) offer rich technical data which could improve the relevance and quality of news inputs.

In addition, more advanced methods, such as retrieval-augmented generation (RAG), could enhance the filtering process by dynamically retrieving the most relevant news articles for each layoff event. We believe this approach is superior to the simple filtering process applied in this study, as it can provide a more comprehensive and contextually relevant dataset. However, due to time constraints, we were unable to explore this method further. Incorporating RAG in future work could significantly improve the model's ability to identify and focus on the most pertinent information, thereby enhancing prediction accuracy. Such a targeted approach could also make the binary classification problem—predicting whether a layoff will occur—more feasible by providing deeper insights into the nuances of individual companies.

Additionally, while fine-tuning pre-trained models like FinBERT has proven effective, we believe that further pretraining (and not fine-tuning) on a large corpus of stock and layoff-related news from diverse sources could make these models more specialized in layoff prediction.

Future work could also explore the integration of dynamic data sources to create models capable of online learning. For example, the FINVIZ API (FINVIZ) provides a continuous stream of financial and stock-related news from various sources. Leveraging this API within a real-time pipeline would allow a model to continuously update its knowledge, adapt to evolving trends, and improve prediction capabilities over time. This online learning approach could be a valuable extension to static models trained on pre-collected datasets.

Despite these limitations, our work lays the foundation for further exploration of better models, advanced pretraining techniques, and dynamic data pipelines, for more robust and adaptable layoff prediction systems. Moreover, given the almost complete absence of prior work in this area, we hope that our study sparks additional interest and future research into the challenges and opportunities of layoff prediction.

References

Layoffs.fyi.

Adil Moghar. 2020. Stock market prediction using lstm recurrent neural network.

Tranco API. Tranco list. https://tranco-list.eu/.

Dogu Tan Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Vladimir Blagojevic. 2024. Cc news: Common crawl news dataset with newsplease extraction.

Max Dallabetta et al. 2024. Fundus: Benchmarking information extraction from html. *arXiv preprint arXiv:2403.15279*.

FINVIZ. Finviz news aggregator. https://finviz.com/news.ashx.

Ulrike Herold. 2024. Tech layoffs 2020-2024.

Nikita Katulevskiy. 2023. Tech layoffs prediction project.

Kasra Naftchi-Ardebili and Karanpartap Singh. 2024. Predicting stock market trends from news articles and price trends using transformers.

Stanford OVAL. 2024. Ccnews: Common crawl news dataset.

Guilherme Penedo, Hynek Kydlíček, Alessandro Cappelli, Mario Sasko, and Thomas Wolf. 2024. Datatrove: large scale data processing.

Financial Modeling Prep. Financial modeling prep api. https://site.financialmodelingprep.com/developer/docs.

Sebastian Scheitle et al. 2019. A comprehensive toplist approach for the web: Tranco.

Sentence-Transformers. 2020. all-mpnet-base-v2.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*.

A Appendix A: Top 26 English News Websites from Tranco

Table 4 presents the top news websites appearing in the top 1k websites according to Tranco from August 2024.

#	Website	Tranco Rank
1	yahoo.com	46
2	forbes.com	157
3	theguardian.com	173
4	reuters.com	268
5	washingtonpost.com	273
6	dailymail.co.uk	281
7	go.com (abcnews.go.com)	294
8	businessinsider.com	339
9	cnbc.com	368
10	telegraph.co.uk	372
11	npr.org	394
12	foxnews.com	399
13	time.com	430
14	cnet.com	461
15	timesofindia.indiatimes.com	487
16	independent.co.uk	503
17	cbsnews.com	537
18	latimes.com	607
19	pbs.org	807
20	whitehouse.gov	808
21	ndtv.com	839
22	hindustantimes.com	855
23	apnews.com	925
24	weforum.org	960
25	businesswire.com	974
26	theconversation.com	979

Table 4: Top 26 English News Websites from Tranco List Ranked in the Top 1000.

B Appendix B: Reference Sentences for Similarity Calculation

The following 15 sentences were used as reference points to measure the similarity of article summaries in relation to layoff-related content:

- "The company announced a massive layoff affecting thousands of employees."
- "Due to financial difficulties, several departments will face job cuts."
- "The stock price plunged following reports of an earnings miss."
- "A downturn in the market has severely impacted the company's profits."
- "The company is undergoing a restructuring that will lead to layoffs."
- "Quarterly losses have forced the company to reduce its workforce."
- "A significant drop in revenue is pushing the company to lay off workers."
- "The board decided to cut jobs to manage the company's declining earnings."
- "Economic challenges are driving layoffs in multiple sectors."
- "The company announced a hiring freeze and potential job cuts."
- "Due to unforeseen financial challenges, layoffs have become necessary."
- "Market volatility has led to a decline in the company's stock value."
- "Facing financial instability, the company is downsizing."
- "The firm is closing down divisions due to underperformance."
- "Shares fell sharply after the company issued a profit warning."