Bank of England

EcoFinBench – a natural language processing benchmark for economics and finance

Staff Working Paper No. 1,163
December 2025

Max Ahrens, Dragos Gorduza and Michael McMahon

Bank of England

Staff Working Paper No. 1,163

EcoFinBench – a natural language processing benchmark for economics and finance

Max Ahrens[1], Dragos Gorduza[2] and Michael McMahon[3]

Abstract

We introduce EcoFinBench, a natural language processing (NLP) benchmark suite for the domains of economics and finance. We comprehensively test a large array of NLP models across multiple domain-specific data sets for sentence classification. Specifically, we evaluate dictionary models, word count models, topic models, and modern transformer models. Furthermore, we introduce two new data sets to the research community. The Bluebook data set for text-only sentiment analysis in monetary policy, and the Greenbook data set for multimodal (text and numeric) sentiment analysis. We focus on data sets that require the models to work with relatively few data points and long average text lengths – typical characteristics of data sets in the economic and financial domain. We find that, dictionary models – still widely used as a default text analysis tool in economics and finance – underperform substantially across all evaluated data sets. From our findings, we conclude that given the underperformance of existing solutions in the multimodal domain, future modelling work is needed. With our benchmark suite we aim to lay the foundation for a systematic assessment on the most commonly used NLP models in economics and finance. To our knowledge, we are the first to provide such holistic benchmarking assessment for economics and finance.

Key words: Machine learning, natural language processing, artificial intelligence, benchmark.

JEL classification: C45, Y10, C55, C88, G17.

---

(1) Maihem.ai – work conducted while at the University of Oxford
(2) Bank of England. Email: dragos.gorduza@bankofengland.co.uk
(3) University of Oxford

The views expressed in this paper are those of the authors, and not necessarily those of the Bank of England or its committees. The authors wish to extend their thanks to the University of Oxford and the Economic and Social Research Council Grand Union DTP (project ES/P000649/1). Furthermore, the authors thank staff at the Bank of England, as well as the reviewers and the discussant during the ECONDAT 2025 Summer Conference, for their comments.

The Bank’s working paper series can be found at www.bankofengland.co.uk/working-paper/staff-working-papers

Bank of England, Threadneedle Street, London, EC2R 8AH
Email: enquiries@bankofengland.co.uk

ISSN 1749-9135 (on-line)

1 Introduction

Text analysis has a long tradition in economics and finance. Tasks such as financial sentiment analysis, text classification, document similarity assessment, and information extraction from text documents play a pivotal role in these domains. Financial dictionary models, in particular Loughran and Mcdonald (2011), have for a long time been a workhorse model used by both practitioners and researchers in this domain, despite or perhaps because of its relatively simplistic conceptual framework.

As the field of natural language processing (NLP) progressed, new text analysis methods found increasing application in economics and finance. Topic models witnessed and still witness substantial popularity (Hansen and McMahon, 2016; Larsen and Thorsrud, 2019). With the introduction of transformers (Vaswani et al., 2017), deep neural network models for NLP found their way into economics and finance research. Transformers such as BERT (Devlin et al., 2018) became the foundational models for financial domain adaptations such as FinBERT (Araci, 2019) and FLANG-BERT (Shah et al., 2022).

Advancements in NLP have benefited from concerted actions to establish common benchmark datasets, tasks, and evaluation metrics in the research community – prominent examples being the General Language Understanding Evaluation (GLUE) (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a). However, there is a lack of an established NLP benchmark for the domains of economics and finance, as pointed out by Ash et al. (2021). Shah et al. (2022) might be the first paper to address this benchmark vacuum, suggesting the Financial Language Understanding Evaluation (FLUE) benchmark tasks. But their work considers transformer architectures only. They compare BERT (Devlin et al., 2018) and ELECTRA (Clark et al., 2020) to the authors' domain adapted equivalents. What is needed is a holistic benchmark, comparing the classes of NLP models that have been introduced into and adopted by the economics and finance domain over the past years. It is important to take stock, systematically evaluate, and produce visibility about which NLP models work best under which dataset characteristics in these fields.

As one specific case in point, in economics and finance, text data often comes accompanied with important numeric metadata. For instance, economic reports and financial statements are usually written in the context of current market conditions. Without the economic and financial context that is often encoded in numeric data, text passages can be ambiguous in their meaning. Take for example the statement of a central bank that forecasts the inflation rate to increase by 2 percentage points until the end of the year. Is this good, bad, or neutral news for the financial markets? It depends. It depends on the current economic conditions, such as the current inflation rate. If such statement was made in a deflationary economic environment, in which the consumer price index (CPI) is substantially below the central bank's target (of usually 2%), a signal to the markets that CPI is about to pick up could be good news. On the other hand, the same statement given in an economic environment of hyperinflation would be seen as bad market news. In short, we need multimodal benchmark evaluations in which we test models on how well they can make predictions that rely on the joint assessment of text and numeric data. In the future, we might want to consider adding further modalities such as audio, image, or video.

With this paper, we aim to provide a starting ground for building a systematic NLP benchmark evaluation in economics and finance. We establish an initial NLP model benchmark suite that includes dictionary models, word count models, topic models, and Transformer-based models. An overview on the model suite is given in Section 4. We actively welcome contributions and additions from the research community. We chose some of the most common NLP datasets used in economics and finance and added an additional multimodal NLP dataset to establish a dataset foundation to which the community is invited to contribute. The current benchmarking task is sentence classification of which sentiment analysis is likely the most widely applied use case in economics and finance. An overview on the dataset suite is provided in Section 3. We aim to extend our benchmark to include additional tasks such as document similarity or named-entity-recognition (NER) in the future.

Finally, we suggest that it might be time for the economics and finance community to move on from using hand-crafted dictionary models such as Loughran and Mcdonald (2011) (LMdict) as their default methods. Our findings show that LMdict methods underperform substantially across all evaluated datasets.

So far, there exist no theoretical foundations as to which NLP model is optimal in which use case. We believe it would be good practice for any NLP-centered work in the economics and finance domain to consider an array of NLP models to add further robustness to research findings whilst reducing model bias in empirical findings.

2 Related Work

In their recent paper, Ash et al. (2021) point out the lack of established NLP benchmarks in economics and finance. Recently, several papers compared particular subsets of NLP models used in economics and finance on various

downstream tasks. Das et al. (2022) compare the workhorse dictionary model from Loughran and Mcdonald (2011) against the authors' machine learning based financial dictionary models. Overall results don't show substantial performance increases by the new approaches. Boukes et al. (2020) and van Atteveldt et al. (2021) find that dictionary model performance is often close to chance and inter-dictionary consistency is low. Such findings suggest that it might be time to default to more performant NLP model alternatives. Our paper features a dedicated assessment of the performance of the workhorse dictionary models by Loughran and Mcdonald (2011) against data-driven machine learning methods. This is especially important today as transformer models like RoBERTa (Liu et al., 2019) or FinBERT Araci (2019) have become easier to train and LLMs like the GPT family are increasingly available in an off-the-shelf fashion, without requiring any retraining. Moreover, these models are seeing increasing use in economic and financial practice as they demonstrate substantial performance in analysing sentiment for among other tasks automated trading Kirtac and Germano (2024).

Our work aligns with other benchmarking initiatives in the field such as Nasiopoulos et al. (2025) who benchmark traditional machine learning models and more modern text-specific transformer models on two financial NLP tasks or Li et al. (2023) who test transformer models GPT4 and Chat-GPT on 8 datasets and 5 tasks including sentiment classification but also named entity recognition and question-answering. Shah et al. (2022) propose FLUE, a transformer-only benchmark for the financial domain. However, what is particularly needed in this domain is an evaluation that compares the entire spectrum of NLP models, from simple dictionaries and word count models to state-of-the-art (SOTA) LLMs. We are laying the foundation for such a benchmark suite. Leippold (2023) also performs a benchmark evaluation of adversarial attacks on financial texts created with GPT-3 (Brown et al., 2020) and finds that standard dictionary models are substantially less robust to such attacks than context-aware transformers such a models from the BERT (Devlin et al., 2018) family. However, we also expand on these initiatives in several ways. First, we compare a large array of NLP models that includes dictionary models, transformer models, and various other NLP models popular in economics and finance, aiming for a broad survey of the field of tools available off the shelf and after fine tuning to economists. Secondly, although we focus our assessment of performance on sentiment classification, we use two datasets focused on monetary policy documents (the Bluebook and Greenbook corpora) which traditional sentiment classification benchmarks have not investigated before. This is in service to the mission of creating an economists' economics and financial NLP benchmark. Thirdly, we expand previous approaches by assessing model performances in a multimodal (text + numeric data) setting going beyond the text-only approaches highlighted elsewhere in economics and finance NLP. The importance of multimodal task benchmarks has been recently emphasized by Ash et al. (2021).

In summary, we compare a large array of NLP models that includes dictionary models, transformer models, and various other NLP models popular in economics and finance.

3 Datasets and Evaluation

3.1 Datasets

Our current benchmark suite contains the following datasets for which Table 1 shows the descriptive statistics:

1. The Financial Phrase Bank (FPB) dataset[1] (Malo et al., 2014), contains sentences from financial news that are annotated as reflecting either positive, negative, or neutral sentiment. We use only sentences that have 100% annotator agreement.
2. The Twitter Financial News (TFN) dataset[2] contains annotated finance-related tweets classified as either bearish, bullish, or neural.
3. The FOMC Bluebook Alternatives (FBA) dataset[3] contains FOMC statement alternatives and their respective fed funds rate (FFR) decision. The original labels are the proposed rates in each alternative and we categorise the label as negative if the FOMC decision leads to a decrease in the FFR, neutral if the FFR is unchanged, and positive if the FFR has been increased.
4. The FOMC Greenbook CPI forecasts (FGC) dataset[4] contains the Greenbook paragraphs about inflation (consumer price index) and the associated one-quarter-ahead Greenbook forecast that we use as a label. We define the forecast as either increasing, decreasing, or unchanged. Furthermore, the dataset contains numeric metadata which consists of the one-quarter-ahead forecasts on CPI, GDP growth, and unemployment of the previous FOMC meeting.

[1] https://huggingface.co/datasets/financial_phrasebank
[2] https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment
[3] https://github.com/mcmahonecon/FOMC-Alt-Statements
[4] Author Michael McMahon. Available upon request

Table 1: Descriptive statistics of datasets in EcoFinBench

Dataset	Data Type	Total	Train Split	Val Split	Test Split	Neg (%)	Neut (%)	Pos (%)	Mean Len	Max Len	Min Len
Fin Phrase Bank	text	2,264	60%	15%	25%	13%	61%	25%	122	315	9
Fin. Twitter	text	11,931	64%	16%	20%	15%	20%	65%	86	227	2
Bluebooks	text	418	64%	16%	20%	6%	75%	17%	2,716	5,934	666
Greenbooks	text+tab	144	64%	16%	20%	48%	6%	47%	3,940	13,063	292

The datasets we chose reflect what we think are some of the key characteristics of text sources in the domain of economics and finance:

1. Economics and finance text datasets can often be rather small, containing only hundreds or thousands of data points compared to millions or billions of data points in 'classic' NLP datasets.
2. Economics and finance text datasets can often contain rather long text sequences, as reflected in the Bluebook and Greenbook dataset which have mean sequence lengths of around 3,000 and 4,000 words, and maximum sequence lengths of 6,000 to 13,000 words.
3. Economics and finance text datasets can often contain potentially relevant numeric metadata that is vital to gauge the context in which a statement is to be interpreted, as reflected in the Greenbook dataset.

We aim to consistently expand the selection of economic and financial datasets and are actively welcoming contributions from the research community. Finally, a note on models in the benchmark suite that have used some of those datasets for training. FinBERT and FLANG-BERT have been fine-tuned on FPB. As we don't have the exact train-test split that was used by the authors, we report the test results on FPB from the original papers (where available). We cross-check those results with our own test results – being mindful that those results might have been subject to training data leakage. Our test results for FinBERT and FLANG-BERT, however, show very comparable results to the figures reported in the respective original papers.

3.2 Evaluation Metrics

We use the F1 score as the overall performance metric for all classification tasks. For classification tasks with more than two classes, we report the macro F1 score. The F1 score is widely used as an evaluation metric for machine learning and NLP classification tasks. Particularly in unbalanced datasets it provides a more meaningful assessment of the model performance than, say, simple accuracy. The F1 score is the harmonised mean over precision and recall. Precision also known as positive predictive value is defined as

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}. \tag{1}$$

Recall is also known as sensitivity and is defined as

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{2}$$

Here, tp represents the true positives, fp the false positives, and fn the false negatives. The F1 score is then

$$\text{F1} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}. \tag{3}$$

For multilabel classification (more than two classes), the macro F1 score calculates the F1 metric for each class label (one-vs-rest) and then takes the arithmetic (unweighted) mean over the per-class F1 scores. The macro F1 score therefore treats all classes equally independent of their support. To generate robust estimates for the F1 score for each model on each task we run each experiment 50 times and average out the F1 score across each run.

Our experimental setup is the following : for each dataset, we create one held out testing set of 20% of the entire dataset (except for the FFB where we use 25% to keep with the original paper's testing set size) and create 50 random training, validation sets by shuffling the remaining non-testing data. Each model is then trained and validated once on on each training-validation pair before being tested.

4 Models

In this section, we briefly outline the NLP models in our benchmark evaluation.

4.1 Dictionary Models

We use the Loughran and Mcdonald (2011) financial dictionary (LMdict) as the representative dictionary model for our benchmark. It is a dictionary models widely used by both researchers and practitioners in the fields of economics and finance. LMdict provides hand-crafted word lists along seven categories: negative, positive, uncertain, litigious, strong modal, weak modal, and constraining. Typically, these seven LMdict features represent the percentage share of words in the input sequence found in the respective word list categories. In practice, researchers often focus only on the first four of these categories (Das et al., 2022). We deploy the software implementation from the Software Repository for Accounting and Finance[5] by the University of Notre Dame. We define three different LMdict approaches.

LMdict-naive: We label the most simplistic implementation LMdict-naive. This approach only uses the positive ($\%positive$) and negative ($\%negative$) feature category. Moreover, LMdict-naive does not learn any parameter weights in a data-driven way. It classifies a sequences as negative, neutral, or positive, according to the following naive thresholds:

$$y = \begin{cases} negative, & \eta \geq 0.75 \\ neutral, & 0.75 < \eta > 0.25 \\ positive, & \eta \leq 0.25 \end{cases} \tag{4}$$

where η is the *negative-positive ratio* defined as

$$\eta = \frac{\%negative}{\%negative + \%positive}. \tag{5}$$

LMdict-lin: We use the seven word list features from the LMdict software implementation: *% negative*, *% positive*, *% uncertainty*, *% litigious*, *% strong modal*, *% weak modal*, *% constraining*. We add the additional features *# of words*, *# of digits*, *# of numbers*, which are also provided by the LMdict software implementation. We then feed these features into a logistic regression to learn the parameter weights for the features given the training dataset.

LMdict-nonlin: We use the same feature set as in LMdict-lin, but instead of a logistic regression, we fit a model zoo of non-linear machine learning methods on the respective training dataset. We use the AutoGluon (Erickson et al., 2020) machine learning model zoo (see Appendix A).

4.2 Word Count Models

We extract the word counts from the input text sequences and represent them in a document-term-matrix (DTM). The word counts are subsequently weighted by their tf-idf score. The tf-idf scaled word counts then serve as features in a linear and a non-linear classification model.

WordCount-lin: We use the tf-idf scaled word counts as features in a logistic regression model that uses elastic net regularisation. The ratio between lasso and ridge regularisation is a hyperparameter in the following range [.1, .5, .7, .9, .95, .99, 1] that is tuned via cross-validation in the model training phase. Meanwhile, the strength of the regularisation penalty is kept as its default value 100.

WordCount-nonlin: We use the same feature set as in WordCount-lin, but instead of a logistic regression, we fit a model zoo of non-linear machine learning methods on the respective training dataset. We use the AutoGluon machine learning model zoo (see Appendix A).

4.3 Topic Models

We fit unsupervised latent Dirichlet allocation (LDA) models (Blei et al., 2003) on the respective training datasets. We estimate LDA models with $K = [5, 10, 50, 100, 250, 500, 1000]$ topics. We then use the estimated topic parameters as features in a logistic regression.

4.4 Transformer Models

We use BERT (Devlin et al., 2018) as our reference architecture for transformer models. We use the following pretrained Hugging Face[6] implementations and then fine-tune them on our respective training datasets. The finetuning task is predicting the class label from positive,negative or neutral as described in Table 1 for each dataset. The batch size, and learning rate is shown in Table 2. The latter is chosen using hyperparamter tuning.

[5] sraf.nd.edu/loughranmcdonald-master-dictionary/

[6] https://huggingface.co/

BERT-base: The standard BERT-base uncased model (Devlin et al., 2018) (hereafter BERT-base to distinguish it from other BERT model extensions and variants) which has not been further pretrained for financial domain applications.

FinBERT: FinBERT (Araci, 2019) is a BERT model that has been further pretrained on a financial corpus containing a subset of the Thomson Reuters Text Research Collection (TRC2), and then been fine-tuned on a subset of the Financial Phrase Bank dataset (Malo et al., 2014).

FLANG-BERT: FLANG-BERT Shah et al. (2022) is also based on a BERT model. It has been pretrained on general English corpora from Wikipedia and BookCorpus as well as financial corpora from SEC EDGAR, Reuters, Bloomberg, Seeking Alpha, and Investopedia. Furthermore, the model makes use of preferential token masking for financial terms.

4.5 Multimodal Model Extensions

We add model extensions to deal with multimodal tasks.

Tab+: We use the TabularPredictor model architecture from AutoGluon to fuse tabular numeric data with text data. This yields the same models as described above, but with a multimodal fusion capability. Such models are labeled with a "Tab+" prefix.

Multimodal Transformer: We add one additional model, which we name Multimodal Transformer. This model uses the *MultimodalPredictor* model architecture instead of *TabularPredictor* (Erickson et al., 2020). The difference is that this approach directly fuses multiple neural network models for the different modalities. For more details, see Erickson et al. (2020). For the text modality, the model uses a transformer architecture, we use BERT-base.

Model	Batch Size	Num. Epochs	Learning Rates
BERT-base	8	10	$\{10^{-4}, 5 \times 10^{-5}, 10^{-6}\}$
FinBERT	8	10	$\{10^{-4}, 5 \times 10^{-5}, 10^{-6}\}$
FLANG-BERT	8	10	$\{10^{-4}, 5 \times 10^{-5}, 10^{-6}\}$
AutoGluon	8	10	$\{10^{-4}, 5 \times 10^{-5}, 10^{-6}\}$
Multimodal Transformer	8	10	$\{10^{-4}, 5 \times 10^{-5}, 10^{-6}\}$

Table 2: Training parameters for Transformer models.

5 Results

Table 3 shows the results for the text-only datasets. Table 4 summarises the results for the multimodal Greenbook dataset. In these tables, we report the best and worst performing models as well as the ensembled model of the AutoGluon model zoo. A detailed report of all individual models can be found in Appendices B and C. In Figure 1, we show the macro F1 score benchmark across all datasets and across the NLP model classes.

Financial transformer models perform best on text-only datasets Table 3: Overall, (financial) transformer models perform best on text-only datasets. The difference is by and large statistically significant apart from the FOMC Bluebook dataset. While a financial transformer achieved a higher average F1 scores than a non-financial transformer, the difference is not statistically significant at a 5% level, as both means lie within two standard deviations of one another. Interestingly, relatively simple WordCount-based models perform quite competitively. They perform as well as (financial) transformers on the FBA dataset and closely behind the transformers on the TFN and FPB dataset.

Financial transformer models struggle on multimodal economics and finance datasets Table 4: For the multimodal Greenbook CPI dataset, the transformer models are far from being among the top performing models. The best performing model is an AutoGluon ensembling model that uses the Loughran-McDonald text features as well as the numeric metadata. These differences are statistically significant. The Greenbook dataset contains very few observations (< 200). Furthermore, its respective text sequences are relatively long with an average length of $3,000$ words per document. Such data characteristics can be quite common for economic or financial text datasets and we observe that even finance-tuned large language models such as FinBERT and FLANG-BERT struggle with this dataset and yield no edge over non-financial transformers (BERT-base) on this dataset. The Multimodal Transformer performs in the same ballpark as the text-only transformers, unable to improve performance by leveraging the information in the numeric metadata. A possible reason for the subpar performance of transformer models on the might be the long sequence length of the dataset. For example, any model based on the BERT-architecture will have to truncate input sequences at 512 tokens. Since Greenbook sequences are on average even longer than Bluebook sequences, this means that even more

information is lost. A maximum sequence length of 512 tokens is quite a strong limitation especially for a financial transformer, given that many real world datasets in the economic and financial domain contain substantially longer sequences (e.g. SEC filings and central bank documents). We aim to add transformer models that can handle longer text sequences, such as Longformer (Beltagy et al., 2020) Bigbird (Zaheer et al., 2020), which can handle up to 4096 tokens per input sequence. This should yield us additional insights whether sequencing length limitations might be the driving force behind the relatively weak performances of transformers on the multimodal dataset. However, the Bluebook dataset's average sequence length is with 2,700 words also much beyond the BERT cut-off. On that dataset, BERT models still rank among the best overall models. Sequence length limitations might therefore not be the only problem transformers are facing on the multimodal Greenbook dataset.

Relatively simple word count models do very well across all datasets Table 3: An interesting observation is that a relatively simple word count model performs quite well. The linear word count model (WordCount-lin) performs on par with BERT-base on the FBA dataset. It also outperforms all alternative models on the TFN and FPB dataset apart from its non-linear word count counterparts and the considerably larger and more complex transformer models. On the multimodal dataset, models using fusing numeric data with word count features are also performing competitively, 8% (or 4 F1 score units) below the best model.

The Loughran-McDonald dictionary model underperforms substantially across all datasets Figure 2: Figure 2 compares the performance of approaches using LMdict features versus alternative NLP models. In particular, we compare LMdict-naive, LMdict-lin, and LMdict-nonlin against both FinBERT, WordCount-lin, and WordCount-nonlin.

In the text-only datasets, the LMdict-naive underperforms FinBERT between 34-76% (F1 score is 33-72 units lower), see Figure 2. LMdict-lin and LMdict-nonlin underperform FinBERT by 33-48% and 22-39%, respectively. And even in comparison to a linear word count feature model (WordCount-lin), all LMdict methods substantially underperform, even LMdict-nonlin. LMdict-naive underperforms WordCount-lin by 32-72%. LMdict-lin underperforms WordCount-lin by 30-40%. And LMdict-nonlin, the best non-linear machine learning model using LMdict features, underperforms WordCount-lin by 18-30%. All LMdict methods markedly underperform the non-linear word count model (WordCount-nonlin).

Similarly, in the multimodal dataset the LMdict methods lack behind in performance. LMdict-naive, LMdict-lin, and LMdict-nonlin underperform FinBERT by 47%, 20%, and 6% respectively. LMdict-naive and LMdict-lin also underperform WordCount-lin. Only LMdict-nonlin does marginally better than WordCount-lin. However, compared to the non-linear word count model (WordCount-nonlin), even the best performing LMdict model, LMdict-nonlin, underperforms by 26% (or 9.5 F1 score units).

Unsupervised topic models show lack in performance across all datasets Figure 1: Models using topic features from an unsupervised topic model (LDA) do not show competitive performance on sentiment analysis and text classification tasks. Current findings suggest that topic modelling strength relative to other models seems to correlate with sequence lengths in the respective dataset. For example, the topic modelling approaches do not outperform the majority class baseline on the Twitter dataset, which also has the shortest average sequence lengths with 86 words per document. Moreover, on that Twitter dataset, the LDA models across all trials show very low variance, almost always predicting the same output, suggesting that the logistic regression is unable to leverage the features extracted to beat the majority class prediction. On the Bluebook dataset that has considerably longer average sequence lengths (2,700 words), the topic modeling approaches perform acceptably, yet over 16 F1 score units (28%) behind the best models. On the Greenbook dataset, the performance distance to the best models is 16 F1 score units (45 %), yet they are on par with BERT or FinBERT. Unsupervised topic models have been quite frequently deployed for similar empirical research settings in economics and finance. The current results suggest that alternative models might yield better performances. We are currently working on adding non-linear machine learning models using topic features from unsupervised topic models. Equally, we are in the process of adding supervised topic models to obtain more nuanced empirical results on the competitive performance of topic modeling approaches (for sentiment analysis and text classification) in economics and finance.

Table 3: Benchmarks: text-only datasets

(a) **FOMC Bluebook Alternatives**
#train: 327 | #test: 82

model	macro F1
WordCount-nonlin-best	96.4 (2.8)
FinBERT	96.0 (5.0)
WordCount-nonlin-ens	95.7 (3.7)
WordCount-lin	92.4 (13.8)
FLANG-BERT	91.0 (9.2)
BERT-base	86.6 (12.6)
WordCount-nonlin-worst	79.7 (5.4)
LDA-K1000	77.7 (8.2)
LDA-K500	76.8 (7.8)
LMdict-nonlin-best	77.1 (7.7)
LDA-K250	75.9 (7.4)
LMdict-nonlin-ens	75.4 (6.3)
LDA-K100	70.5 (8.1)
LDA-K50	67.8 (8.9)
LMdict-lin	64.6 (6.6)
LMdict-naive	63.3 (6.5)
LMdict-nonlin-worst	54.4 (5.8)
LDA-K10	50.2 (10.6)
LDA-K5	40.1 (7.8)
Majority class	28.8 (0.8)

(b) **Twitter Financial News**
#train: 9,545, #test: 2,386

model	macro F1
FLANG-BERT	82.2 (2.9)
FinBERT	81.5 (1.3)
BERT-base	80.8 (1.7)
WordCount-nonlin-best	77.7 (0.0)
WordCount-nonlin-ens	76.6 (0.0)
WordCount-lin	67.8 (1.1)
LMdict-nonlin-best	51.5 (0.0)
LMdict-nonlin-ens	50.0 (0.0)
LMdict-nonlin-worst	41.7 (0.0)
LMdict-lin	37.3 (-)
LMdict-naive	29.6 (-)
LDA-K100	26.4 (0.2)
LDA-K50	26.4 (0.0)
LDA-K10	26.4 (0.0)
LDA-K5	26.4 (0.0)
LDA-K1000	26.4 (0.0)
LDA-K500	26.4 (0.0)
Majority class	26.4 (0.0)
WordCount-nonlin-worst	26.4 (0.0)
LDA-K250	26.4 (0.0)

(c) **Financial Phrase Bank benchmark**
#train: 1,698, #test: 566

model	macro F1
FinBERT	94.7 (1.2)
BERT-base	94.4 (1.5)
FLANG-BERT	94.1 (1.5)
WordCount-nonlin-best	87.1 (1.6)
WordCount-nonlin-ens	87.0 (1.6)
WordCount-lin	82.9 (1.8)
LDA-K1000	63.7 (4.0)
LMdict-nonlin-ens	58.3 (2.2)
LMdict-nonlin-best	58.3 (2.3)
LMdict-nonlin-worst	52.9 (2.1)
WordCount-nonlin-worst	51.8 (2.9)
LMdict-lin	50.0 (-)
LDA-K500	47.6 (9.3)
LDA-K100	44.1 (4.0)
LDA-K250	44.0 (4.4)
LDA-K50	42.7 (3.2)
LDA-K10	29.6 (6.1)
LDA-K5	25.6 (2.2)
Majority class	25.1 (-)
LMdict-naive	23.0 (-)

The tables display average F1-score over 50 model runs with standard deviation in brackets, both in percent. The original F1 test score for FinBERT is 95%. The FLANG-BERT paper does not report F1 scores.

Table 4: Benchmarks: multimodal dataset

FOMC Greenbook CPI Multimodal
#train: 112, #test: 28

model	macro F1
Tab+LMdict-nonlin-best	46.7 (0.0)
Tab-best	44.2 (0.0)
Tab+encod-best	43.1 (0.0)
Tab+WordCount-nonlin-ens	42.8 (0.0)
Tab+WordCount-nonlin-best	42.7 (0.0)
WordCount-nonlin-best	42.1 (0.0)
Tab+LMdict-nonlin-ens	39.8 (0.0)
Tab-ens	37.3 (0.0)
WordCount-nonlin-ens	36.8 (0.0)
Tab+encod-ens	34.8 (0.0)
BERT-base	31.0 (7.1)
LDA-K100	30.6 (8.3)
LDA-K50	30.3 (8.0)
LDA-K10	29.9 (7.9)
LDA-K250	29.9 (7.6)
LDA-K1000	29.5 (5.6)
LDA-K500	29.5 (6.3)
FinBERT	29.0 (7.2)
FLANG-BERT	28.7 (6.6)
Multimodal Transformer	28.1 (0.0)
LDA-K5	28.1 (7.6)
LMdict-nonlin-ens	27.3 (0.0)
WordCount-lin	25.6 (1.9)
LMdict-lin	23.3 (0.0)
Tab+WordCount-nonlin-worst	21.7 (0.0)
Tab+encod-worst	21.7 (0.0)
Tab+LMdict-nonlin-worst	21.7 (0.0)
Majority class	21.7 (-)
Tab-worst	21.7 (0.0)
Tab+LMdict-nonlin-worst	21.5 (0.0)
Tab+WordCount-nonlin-worst	21.1 (0.0)
LMdict-naive	15.3 (0.0)

The table displays the average F1-score over 50 model runs with standard deviation in brackets, both in percent.

Figure 1: F1 scores across all datasets, in percent

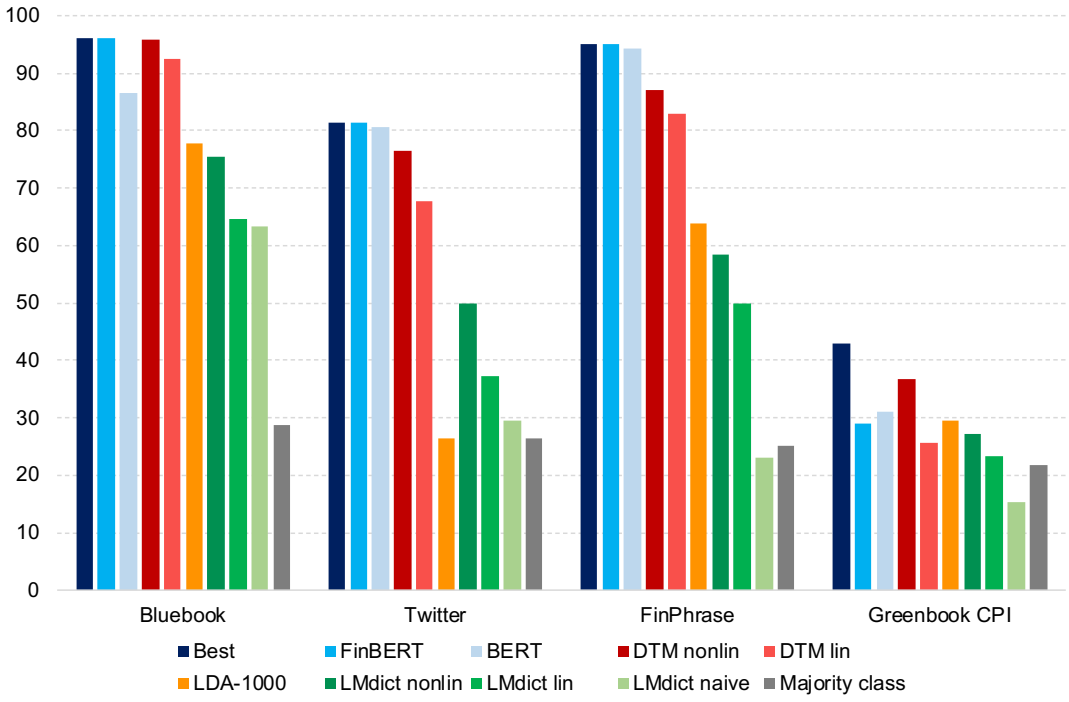
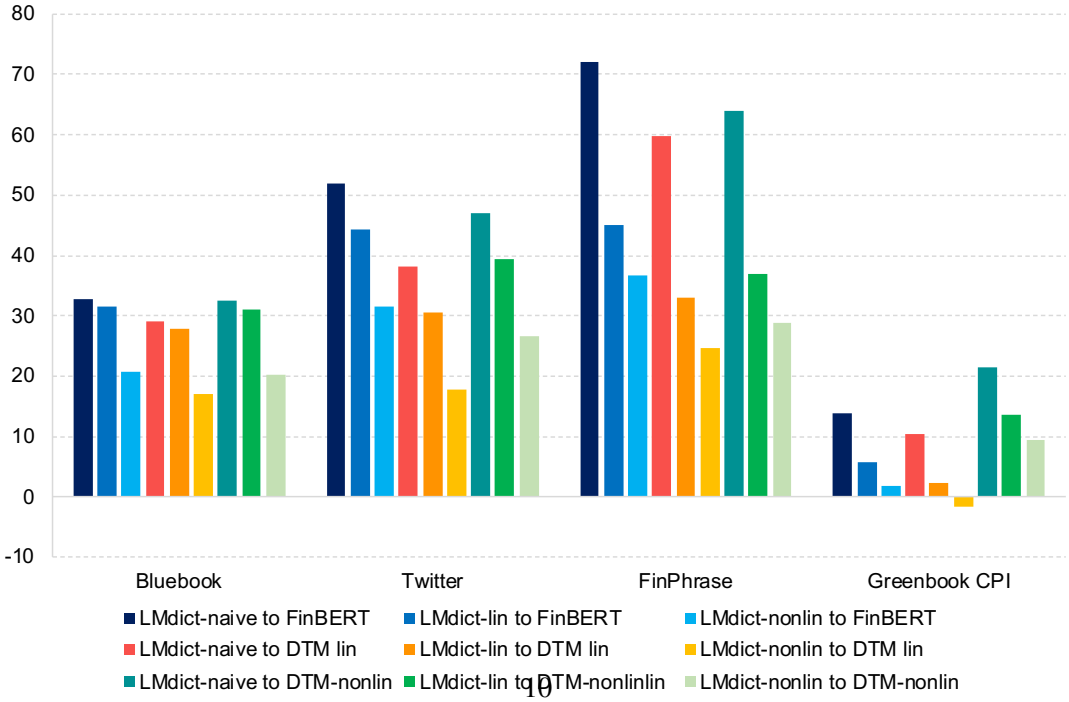


Figure 2: F1 score distances between dictionary model and other models, in percent



6 Conclusion

This paper introduces a more comprehensive NLP benchmark evaluation for sentiment detection in the domains of economics and finance. Over the past years, researchers introduced various NLP methods in these domains to be able to answer important research questions and extract signals from text. Now might be a good time to take stock and start systematically evaluating which NLP methods work best for the most common domain specific tasks, given the most common domain specific dataset characteristics. In particular, we would like to underline the importance of evaluating multimodal (for now, text and numeric data) tasks, since many NLP datasets in economics and finance come accompanied by potentially relevant numeric metadata. We have identified four main points from our analysis. Firstly, we find that financial transformer models perform best on text-only classification datasets. Secondly however, we find these models but struggle on multimodal economics and finance datasets, highlighting more need for concerted research efforts on fusing multimodal data and pre-training/fine-tuning models for tasks in these domains. Thirdly, we highlight that the Loughran and Mcdonald (2011) dictionary model – still widely used as a default text analysis tool in economics and finance – underperforms substantially across all evaluated datasets. Lastly, we find that relatively simple word count models do very well across all datasets. Our evaluation suite is by no means making a claim on being exhaustive. With this paper, we aim to lay the foundational upon which the research community can easily build and extend. We highly encourage suggestions for additions of new models, datasets, tasks, and evaluation metrics. Going forward, we aim to make our benchmark suite more interactive, providing a online interface that allows the research community to easily evaluate new datasets and models. We also plan to extend our benchmark suite beyond English, establishing a more comprehensive language coverage of both high and low resource languages. Furthermore, we have not included generative models in this benchmarking exercise. These models are difficult to benchmark exhaustively due a combination of computational costs for locally run models and the difficulty to maintain reproducibility for models on cloud like the GPT-family of models. We anticipate to introduce them as a useful extension in future work.

References

Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models.

Elliott Ash, Germain Gauthier, and Philine Widmer. 2021. Text Semantics Capture Political and Economic Narratives. *SSRN Electronic Journal*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv*.

David M Blei, Blei@cs Berkeley Edu, Andrew Y Ng, Ang@cs Stanford Edu, Michael I Jordan, and Jordan@cs Berkeley Edu. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Mark Boukes, Bob van de Velde, Theo Araujo, and Rens Vliegenthart. 2020. What's the Tone? Easy Doesn't Do It: Analyzing Performance and Agreement Between Off-the-Shelf Sentiment Analysis Tools. *Communication Methods and Measures*, 14(2):83–104.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

Sanjiv R. Das, Michele Donini, Muhammad Bilal Zafar, John He, and Krishnaram Kenthapadi. 2022. FinLex: An effective use of word embeddings for financial lexicon generation. *Journal of Finance and Data Science*, 8:1–11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Sahibsingh A Dudani. 1976. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63(1):3–42.

Stephen Hansen and Michael McMahon. 2016. Shocking language: Understanding the macroeconomic effects of central bank communication. *Journal of International Economics*, 99:114–133.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.

Kemal Kirtac and Guido Germano. 2024. Sentiment trading with large language models. *Finance Research Letters*, 62:105227.

Vegard H. Larsen and Leif A. Thorsrud. 2019. The value of news for economic developments. *Journal of Econometrics*.

Markus Leippold. 2023. Sentiment Spin : Attacking Financial Sentiment with. pages 1–24.

Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? a study on several typical tasks. *arXiv preprint arXiv:2305.05862*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, and Paul G Allen. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Technical report.

Tim Loughran and Bill Mcdonald. 2011. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance*, 66(1):35–65.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Dimitrios K Nasiopoulos, Konstantinos I Roumeliotis, Damianos P Sakas, Kanellos Toudas, and Panagiotis Reklitis. 2025. Financial sentiment analysis and classification: A comparative study of fine-tuned deep learning models. *International Journal of Financial Studies*, 13(2):75.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31.

Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. WHEN FLUE MEETS FLANG: Benchmarks and Large Pre-trained Language Model for Financial Domain.

Wouter van Atteveldt, Mariken A.C.G. van der Velden, and Mark Boukes. 2021. The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms. *Communication Methods and Measures*, 15(2):121–140.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A MULTI-TASK BENCHMARK AND ANALYSIS PLATFORM FOR NATURAL LANGUAGE UNDERSTANDING. In *Proceedings of ICLR*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 2020-Decem.

A AutoGluon: Machine Learning Model Zoo

AutoGluon is an automated machine learning (AutoML) framework that has been developed to fuse multimodal features such as text, images, and tabular data. We chose this AutoML framework because it outperformed competing frameworks in multimodal benchmark tasks (see Erickson et al. (2020)).

Base models: AutoGluon fits machine learning *base models* and then combines them through ensembling and stacking to boost performance. AutoGluon allows us to apply hyperparameter optimization over all models. The *base models* in AutoGluon span the following broad machine learning algorithm classes:

1. **K-nearest neighbours** (Dudani, 1976): AutoGluon uses two variations of k-nearest neighbours (KNN) that differ in their weighting approaches. One allocates uniform weights to all points while the other weights points according to the inverse of their respective distances.
2. **Random forests** (Breiman, 2001): AutoGluon again deploys two variations of this algorithm class. One option uses the information gain of nodes for the assessment of the split quality. The other option uses Gini impurity instead.
3. **Extremely randomized trees** (Geurts et al., 2006): For the random tree class, AutoGluon deploys both an implementation resorting to information gain and another option that uses Gini impurity for the assessment of split quality.
4. **Boosted decision trees**: AutoGluon runs (where applicable to the task) Extreme Gradient Boosting (Chen and Guestrin, 2016), Light Gradient Boosting (Ke et al., 2017), Categorical Boosting (Prokhorenkova et al., 2018).
5. **Neural networks**: A detailed description of the the neural network architecture can be found in **?**. The architecture has been specifically designed for the multimodal use of categorical (text, images) and numerical data. It uses variable-specific embeddings for each of the categorical features. These are then concatenated with the numerical features into one overall input vector. This vector is in turn fed through a 3-layer feed-forward network as well as through a linear skip-connection. Model ensembling and stacking can be applied and are optimally chosen in the validation process.

B Text-only datasets: detailed results

B.0.1 Bluebooks

Table 5: FOMC Bluebooks Alternatives benchmark
train size: 327 | test size: 82 (shuffled)

	macro F1		macro F1
Tabpred:dtm_XGBoost_BAG_L1	96.4 (2.8)	LDA-K500	76.8 (7.8)
FinBERT	96.0 (5.0)	Tabpred:lm_XGBoost_BAG_L1	76.4 (7.1)
Tabpred:dtm_LightGBMLarge_BAG_L1	95.9 (3.1)	LDA-K250	75.9 (7.4)
Tabpred:dtm_CatBoost_BAG_L1	95.7 (2.9)	Tabpred:lm_RandomForestEntr_BAG_L1	75.9 (8.3)
Tabpred:dtm_WeightedEnsemble_L2	95.7 (3.7)	Tabpred:lm_WeightedEnsemble_L2	75.4 (6.3)
Tabpred:dtm_LightGBM_BAG_L1	95.1 (3.7)	Tabpred:lm_ExtraTreesEntr_BAG_L1	74.5 (7.7)
Tabpred:dtm_LightGBMXT_BAG_L1	95.0 (3.4)	Tabpred:lm_ExtraTreesGini_BAG_L1	74.2 (8.4)
Tabpred:dtm_NeuralNetTorch_BAG_L1	94.2 (4.3)	Tabpred:lm_LightGBMLarge_BAG_L1	73.9 (6.8)
WordCount-lin	92.4 (13.8)	Tabpred:lm_LightGBM_BAG_L1	73.7 (6.3)
Tabpred:dtm_RandomForestEntr_BAG_L1	92.1 (4.3)	Tabpred:lm_LightGBMXT_BAG_L1	71.1 (8.1)
Tabpred:dtm_ExtraTreesEntr_BAG_L1	92.1 (4.3)	LDA-K100	70.5 (8.1)
Tabpred:dtm_ExtraTreesGini_BAG_L1	92.0 (4.2)	Tabpred:lm_NeuralNetTorch_BAG_L1	70.4 (8.4)
Tabpred:dtm_RandomForestGini_BAG_L1	91.2 (5.0)	Tabpred:lm_NeuralNetFastAI_BAG_L1	69.5 (8.5)
FLANG-BERT	91.0 (9.2)	LDA-K50	67.8 (8.9)
BERT-base	86.6 (12.6)	LMdict-lin	64.6 (6.6)
Tabpred:dtm_NeuralNetFastAI_BAG_L1	84.3 (5.5)	LMdict-naive	63.3 (6.5)
Tabpred:dtm_KNeighborsDist_BAG_L1	81.1 (5.1)	Tabpred:lm_KNeighborsDist_BAG_L1	56.5 (7.0)
Tabpred:dtm_KNeighborsUnif_BAG_L1	79.7 (5.4)	Tabpred:lm_KNeighborsUnif_BAG_L1	54.4 (5.8)
LDA-K1000	77.7 (8.2)	LDA-K10	50.2 (10.6)
Tabpred:lm_RandomForestGini_BAG_L1	77.1 (7.7)	LDA-K5	40.1 (7.8)
Tabpred:lm_CatBoost_BAG_L1	77.0 (7.1)	majority class	28.8 (0.8)

Avg. over 50 runs, standard deviation in brackets

B.1 Twitter Financial News

Table 6: Twitter Financial News benchmark
train size: 9,545 | test size: 2,386 (not shuffled)

	macro F1		macro F1
FLANG-BERT	82.2 (2.9)	Tabpred:lm_ExtraTreesGini_BAG_L1	45.9 (0.0)
FinBERT	81.5 (1.3)	Tabpred:lm_LightGBM_BAG_L2	45.9 (0.0)
BERT-base	80.8 (1.7)	Tabpred:lm_RandomForestGini_BAG_L1	45.9 (0.0)
Tabpred:dtm_LightGBMLarge_BAG_L2	77.7 (0.0)	Tabpred:lm_ExtraTreesEntr_BAG_L1	45.7 (0.0)
Tabpred:dtm_XGBoost_BAG_L2	77.2 (0.0)	Tabpred:lm_RandomForestEntr_BAG_L1	45.6 (0.0)
Tabpred:dtm_NeuralNetFastAI_BAG_L2	76.9 (0.0)	Tabpred:lm_LightGBMXT_BAG_L2	45.6 (0.0)
Tabpred:dtm_RandomForestGini_BAG_L2	76.8 (0.0)	Tabpred:lm_WeightedEnsemble_L3	45.6 (0.0)
Tabpred:dtm_CatBoost_BAG_L2	76.7 (0.0)	Tabpred:lm_LightGBMLarge_BAG_L1	45.5 (0.0)
Tabpred:dtm_ExtraTreesEntr_BAG_L2	76.7 (0.0)	Tabpred:lm_LightGBMLarge_BAG_L2	45.0 (0.0)
Tabpred:dtm_LightGBM_BAG_L2	76.6 (0.0)	Tabpred:lm_ExtraTreesGini_BAG_L2	45.0 (0.0)
Tabpred:dtm_WeightedEnsemble_L3	76.6 (0.0)	Tabpred:lm_RandomForestGini_BAG_L2	44.9 (0.0)
Tabpred:dtm_LightGBMXT_BAG_L2	76.6 (0.0)	Tabpred:lm_LightGBM_BAG_L1	44.9 (0.0)
Tabpred:dtm_RandomForestEntr_BAG_L2	76.5 (0.0)	Tabpred:lm_XGBoost_BAG_L1	44.8 (0.0)
Tabpred:dtm_ExtraTreesGini_BAG_L2	76.0 (0.0)	Tabpred:lm_RandomForestEntr_BAG_L2	44.7 (0.0)
Tabpred:dtm_WeightedEnsemble_L2	75.8 (0.0)	Tabpred:lm_ExtraTreesEntr_BAG_L2	44.4 (0.0)
Tabpred:dtm_LightGBMLarge_BAG_L1	74.1 (0.0)	Tabpred:lm_LightGBMXT_BAG_L1	44.4 (0.0)
Tabpred:dtm_XGBoost_BAG_L1	72.3 (0.0)	Tabpred:dtm_KNeighborsDist_BAG_L1	44.1 (0.0)
Tabpred:dtm_CatBoost_BAG_L1	69.5 (0.0)	Tabpred:lm_CatBoost_BAG_L1	43.3 (0.0)
Tabpred:dtm_LightGBM_BAG_L1	69.1 (0.0)	Tabpred:lm_NeuralNetTorch_BAG_L1	42.5 (0.0)
Tabpred:dtm_RandomForestGini_BAG_L1	68.3 (0.0)	Tabpred:lm_KNeighborsUnif_BAG_L1	41.8 (0.0)
Tabpred:dtm_ExtraTreesGini_BAG_L1	68.3 (0.0)	Tabpred:lm_KNeighborsDist_BAG_L1	41.7 (0.0)
Tabpred:dtm_LightGBMXT_BAG_L1	68.3 (0.0)	Tabpred:dtm_KNeighborsUnif_BAG_L1	40.2 (0.0)
Tabpred:dtm_NeuralNetFastAI_BAG_L1	68.0 (0.0)	LMdict-lin	37.3 (nan)
WordCount-lin	67.8 (1.1)	LMdict-naive	29.6 (nan)
Tabpred:dtm_ExtraTreesEntr_BAG_L1	67.0 (0.0)	LDA-K100	26.4 (0.2)
Tabpred:dtm_RandomForestEntr_BAG_L1	66.8 (0.0)	LDA-K5	26.4 (0.0)
Tabpred:dtm_NeuralNetTorch_BAG_L2	57.9 (0.0)	LDA-K10	26.4 (0.0)
Tabpred:lm_NeuralNetFastAI_BAG_L2	51.5 (0.0)	LDA-K50	26.4 (0.0)
Tabpred:lm_NeuralNetFastAI_BAG_L1	50.1 (0.0)	LDA-K500	26.4 (0.0)
Tabpred:lm_WeightedEnsemble_L2	50.0 (0.0)	LDA-K1000	26.4 (0.0)
Tabpred:lm_NeuralNetTorch_BAG_L2	48.5 (0.0)	majority class	26.4 (0.0)
Tabpred:lm_CatBoost_BAG_L2	46.7 (0.0)	Tabpred:dtm_NeuralNetTorch_BAG_L1	26.4 (0.0)
Tabpred:lm_XGBoost_BAG_L2	46.4 (0.0)	LDA-K250	26.4 (0.0)

Avg. over 50 runs, standard deviation in brackets

B.2 Financial Phrase Bank

Table 7: Financial Phrase Bank benchmark
train size: 1,698 | test size: 566

	macro F1		macro F1
FinBERT*	95.0 (-)	Tabpred:lm_CatBoost_BAG_L2	56.9 (2.2)
BERT-base	94.4 (1.5)	Tabpred:lm_LightGBMXT_BAG_L1	56.6 (2.2)
FLANG-BERT	94.1 (1.5)	Tabpred:lm_CatBoost_BAG_L1	56.6 (2.1)
Tabpred:dtm_LightGBMXT_BAG_L2	87.1 (1.6)	Tabpred:lm_RandomForestGini_BAG_L2	56.4 (2.1)
Tabpred:dtm_WeightedEnsemble_L3	87.0 (1.6)	Tabpred:lm_RandomForestEntr_BAG_L2	56.1 (2.4)
Tabpred:dtm_LightGBMLarge_BAG_L2	86.9 (1.6)	Tabpred:lm_ExtraTreesEntr_BAG_L2	56.1 (2.3)
Tabpred:dtm_LightGBM_BAG_L2	86.8 (1.5)	Tabpred:lm_WeightedEnsemble_L3	56.1 (2.3)
Tabpred:dtm_XGBoost_BAG_L2	86.8 (1.6)	Tabpred:lm_ExtraTreesGini_BAG_L2	56.0 (2.5)
Tabpred:dtm_CatBoost_BAG_L2	86.8 (1.6)	Tabpred:lm_XGBoost_BAG_L2	55.7 (2.0)
Tabpred:dtm_RandomForestEntr_BAG_L2	85.9 (1.8)	Tabpred:lm_LightGBMXT_BAG_L2	55.6 (2.1)
Tabpred:dtm_RandomForestGini_BAG_L2	85.8 (1.9)	Tabpred:lm_LightGBMLarge_BAG_L2	55.5 (2.0)
Tabpred:dtm_ExtraTreesGini_BAG_L2	85.6 (1.8)	Tabpred:lm_LightGBM_BAG_L2	55.4 (2.2)
Tabpred:dtm_ExtraTreesEntr_BAG_L2	85.5 (1.8)	Tabpred:lm_XGBoost_BAG_L1	55.3 (2.1)
Tabpred:dtm_NeuralNetFastAI_BAG_L2	85.3 (1.6)	Tabpred:lm_LightGBM_BAG_L1	54.4 (2.3)
Tabpred:dtm_CatBoost_BAG_L1	85.2 (1.9)	Tabpred:lm_ExtraTreesGini_BAG_L1	54.3 (2.1)
Tabpred:dtm_NeuralNetTorch_BAG_L2	85.1 (1.8)	Tabpred:lm_ExtraTreesEntr_BAG_L1	54.2 (2.2)
Tabpred:dtm_WeightedEnsemble_L2	84.9 (1.9)	Tabpred:lm_RandomForestEntr_BAG_L1	53.8 (2.2)
Tabpred:dtm_XGBoost_BAG_L1	83.5 (2.0)	Tabpred:lm_RandomForestGini_BAG_L1	53.7 (2.2)
Tabpred:dtm_LightGBMLarge_BAG_L1	83.3 (1.8)	Tabpred:lm_LightGBMLarge_BAG_L1	52.9 (2.1)
WordCount-lin	82.9 (1.8)	Tabpred:dtm_KNeighborsDist_BAG_L1	52.8 (3.1)
Tabpred:dtm_LightGBM_BAG_L1	79.4 (2.6)	Tabpred:lm_KNeighborsDist_BAG_L1	52.2 (2.3)
Tabpred:dtm_NeuralNetFastAI_BAG_L1	78.8 (2.3)	Tabpred:dtm_NeuralNetTorch_BAG_L1	52.1 (1.2)
Tabpred:dtm_LightGBMXT_BAG_L1	78.8 (2.7)	Tabpred:lm_KNeighborsUnif_BAG_L1	52.0 (1.9)
Tabpred:dtm_ExtraTreesGini_BAG_L1	75.6 (2.5)	Tabpred:dtm_KNeighborsUnif_BAG_L1	51.8 (2.9)
Tabpred:dtm_ExtraTreesEntr_BAG_L1	74.8 (2.7)	LMdict-lin	50.0 (0.0)
Tabpred:dtm_RandomForestGini_BAG_L1	74.7 (2.8)	LDA-K500	47.6 (9.3)
Tabpred:dtm_RandomForestEntr_BAG_L1	74.2 (2.7)	LDA-K100	44.1 (4.0)
LDA-K1000	63.7 (4.0)	LDA-K250	44.0 (4.4)
Tabpred:lm_NeuralNetFastAI_BAG_L1	58.3 (2.3)	LDA-K50	42.7 (3.2)
Tabpred:lm_WeightedEnsemble_L2	58.3 (2.2)	LDA-K10	29.6 (6.1)
Tabpred:lm_NeuralNetTorch_BAG_L2	57.8 (2.1)	LDA-K5	25.6 (2.2)
Tabpred:lm_NeuralNetFastAI_BAG_L2	57.7 (2.3)	majority class	25.1 (nan)
Tabpred:lm_NeuralNetTorch_BAG_L1	57.2 (2.2)	LMdict-naive	23.0 (0.0)

Avg. over 50 runs, standard deviation in brackets. FinBERT was trained on part of the Financial Phrase Bank. We report the F1-score of the test set performance reported in the original paper Araci (2019). When we fine-tuned FinBERT on the Financial Phrase Bank nonetheless, we measured an F1 score of 94.7 (1.2) on our test set. Note that there might data leakage between our test set and the original FinBERT test set.

C Multimodal datasets – detailed results

C.1 FOMC Greenbook CPI forecast

Table 8: FOMC Greenbook CPI Multimodal
#train: 112 | #test: 28

	macro F1		macro F1
Tabpred:Tab+lm_ExtraTreesGini_BAG_L1	46.7 (0.0)	textpred:Tab+text_WeightedEnsemble_L2	34.8 (0.0)
Tabpred:Tab+lm_XGBoost_BAG_L1	45.2 (0.0)	textpred:Tab+text_LightGBMXT_BAG_L1	34.8 (0.0)
Tabpred:tab_LightGBMXT_BAG_L1	44.2 (0.0)	Tabpred:dtm_KNeighborsDist_BAG_L1	34.8 (0.0)
Tabpred:Tab+lm_NeuralNetFastAI_BAG_L1	43.6 (0.0)	Tabpred:Tab+dtm_LightGBMLarge_BAG_L1	34.8 (0.0)
textpred:Tab+text_NeuralNetTorch_BAG_L1	43.1 (0.0)	textpred:Tab+text_XGBoost_BAG_L1	34.8 (0.0)
Tabpred:Tab+dtm_WeightedEnsemble_L2	42.8 (0.0)	Tabpred:lm_XGBoost_BAG_L1	34.8 (0.0)
Tabpred:Tab+dtm_XGBoost_BAG_L1	42.7 (0.0)	Tabpred:dtm_KNeighborsUnif_BAG_L1	34.8 (0.0)
Tabpred:tab_CatBoost_BAG_L1	42.1 (0.0)	Tabpred:lm_ExtraTreesEntr_BAG_L1	34.7 (0.0)
Tabpred:Tab+lm_LightGBMLarge_BAG_L1	42.1 (0.0)	Tabpred:lm_RandomForestGini_BAG_L1	34.1 (0.0)
Tabpred:dtm_LightGBMXT_BAG_L1	42.1 (0.0)	Tabpred:Tab+dtm_NeuralNetFastAI_BAG_L1	33.9 (0.0)
Tabpred:tab_NeuralNetFastAI_BAG_L1	41.7 (0.0)	Tabpred:dtm_XGBoost_BAG_L1	32.8 (0.0)
Tabpred:tab_LightGBMLarge_BAG_L1	41.6 (0.0)	textpred:Tab+text_LightGBMLarge_BAG_L1	32.8 (0.0)
Tabpred:Tab+dtm_RandomForestGini_BAG_L1	40.9 (0.0)	Tabpred:lm_NeuralNetFastAI_BAG_L1	32.1 (0.0)
Tabpred:tab_ExtraTreesEntr_BAG_L1	39.8 (0.0)	Tabpred:lm_LightGBMLarge_BAG_L1	32.1 (0.0)
Tabpred:Tab+lm_WeightedEnsemble_L2	39.8 (0.0)	Tabpred:dtm_CatBoost_BAG_L1	32.1 (0.0)
Tabpred:Tab+lm_LightGBMXT_BAG_L1	39.8 (0.0)	textpred:Tab+text_LightGBM_BAG_L1	32.1 (0.0)
Tabpred:Tab+lm_ExtraTreesEntr_BAG_L1	39.8 (0.0)	BERT-base	31.0 (7.1)
Tabpred:tab_ExtraTreesGini_BAG_L1	39.7 (0.0)	LDA-K100	30.6 (8.3)
Tabpred:tab_XGBoost_BAG_L1	39.7 (0.0)	LDA-K50	30.3 (8.0)
Tabpred:Tab+dtm_RandomForestEntr_BAG_L1	39.5 (0.0)	Tabpred:lm_CatBoost_BAG_L1	30.0 (0.0)
Tabpred:dtm_LightGBMLarge_BAG_L1	39.5 (0.0)	Tabpred:dtm_LightGBM_BAG_L1	30.0 (0.0)
Tabpred:Tab+dtm_CatBoost_BAG_L1	38.9 (0.0)	LDA-K10	29.9 (7.9)
textpred:Tab+text_CatBoost_BAG_L1	38.9 (0.0)	LDA-K250	29.9 (7.6)
Tabpred:Tab+dtm_ExtraTreesGini_BAG_L1	38.9 (0.0)	LDA-K1000	29.5 (5.6)
Tabpred:dtm_ExtraTreesGini_BAG_L1	38.0 (0.0)	LDA-K500	29.5 (6.3)
Tabpred:Tab+lm_NeuralNetTorch_BAG_L1	37.4 (0.0)	FinBERT	29.0 (7.2)
Tabpred:tab_WeightedEnsemble_L2	37.3 (0.0)	FLANG-BERT	28.7 (6.6)
Tabpred:tab_LightGBM_BAG_L1	37.3 (0.0)	MMpred:Tab+text_MMpredictor	28.1 (0.0)
Tabpred:Tab+dtm_NeuralNetTorch_BAG_L1	37.3 (0.0)	LDA-K5	28.1 (7.6)
Tabpred:Tab+lm_LightGBM_BAG_L1	37.3 (0.0)	Tabpred:lm_NeuralNetTorch_BAG_L1	27.3 (0.0)
Tabpred:Tab+lm_CatBoost_BAG_L1	37.3 (0.0)	Tabpred:lm_WeightedEnsemble_L2	27.3 (0.0)
Tabpred:tab_RandomForestEntr_BAG_L1	37.3 (0.0)	WordCount-lin	25.6 (1.9)
Tabpred:Tab+dtm_LightGBM_BAG_L1	37.3 (0.0)	Tabpred:lm_KNeighborsDist_BAG_L1	23.6 (0.0)
Tabpred:Tab+lm_RandomForestEntr_BAG_L1	37.3 (0.0)	Tabpred:lm_LightGBM_BAG_L1	23.4 (0.0)
Tabpred:lm_ExtraTreesGini_BAG_L1	37.3 (0.0)	LMdict-lin	23.3 (0.0)
Tabpred:Tab+dtm_LightGBMXT_BAG_L1	37.3 (0.0)	Tabpred:Tab+lm_KNeighborsUnif_BAG_L1	21.7 (0.0)
Tabpred:dtm_NeuralNetFastAI_BAG_L1	36.9 (0.0)	Tabpred:tab_KNeighborsUnif_BAG_L1	21.7 (0.0)
Tabpred:tab_NeuralNetTorch_BAG_L1	36.9 (0.0)	majority class	21.7 (nan)
Tabpred:lm_RandomForestEntr_BAG_L1	36.9 (0.0)	Tabpred:Tab+dtm_KNeighborsUnif_BAG_L1	21.7 (0.0)
Tabpred:lm_KNeighborsUnif_BAG_L1	36.9 (0.0)	Tabpred:Tab+dtm_KNeighborsDist_BAG_L1	21.7 (0.0)
Tabpred:dtm_WeightedEnsemble_L2	36.8 (0.0)	Tabpred:Tab+lm_KNeighborsDist_BAG_L1	21.7 (0.0)
Tabpred:tab_RandomForestGini_BAG_L1	36.1 (0.0)	Tabpred:tab_KNeighborsDist_BAG_L1	21.7 (0.0)
Tabpred:dtm_ExtraTreesEntr_BAG_L1	36.1 (0.0)	textpred:Tab+text_TextPredictor_BAG_L1	21.7 (0.0)
Tabpred:dtm_RandomForestEntr_BAG_L1	36.1 (0.0)	Tabpred:lm_LightGBMXT_BAG_L1	21.5 (0.0)
Tabpred:Tab+dtm_ExtraTreesEntr_BAG_L1	36.1 (0.0)	Tabpred:dtm_NeuralNetTorch_BAG_L1	21.1 (0.0)
Tabpred:dtm_RandomForestGini_BAG_L1	36.1 (0.0)	LMdict-naive	15.3 (0.0)
Tabpred:Tab+lm_RandomForestGini_BAG_L1	35.1 (0.0)		

Avg. over 50 runs, standard deviation in brackets