A sentiment analysis application for improving Brazilian inflation forecasting

Anonymous ACL submission

Abstract

TO DO

1 Introduction

TO DO

2 Related Works

TO DO

3 Methodology

3.1 Creating the Phrase Dataset

3.1.1 Scraping

We collected COPOM (Central Bank of Brazil's Monetary Policy Committee) minutes using Python and Selenium from the official listing (Banco Central do Brasil, 2025a), downloading both HTML and PDF versions when available.

The dataset C contains 251 COPOM minutes from January 1996 to July 2025. Each minute c has an associated date d_i and may have HTML and/or PDF versions.

3.1.2 Parsing

For each COPOM minute c in C:

1. Type-Specific Pre-Processing

   HTML: extract body content, remove formatting tags (`strong`, `i`, `br`) while preserving inner content, remove other tags with content.

   PDF: extract phrases using SpaCy-Layout (Neumann et al., 2019) with `pt_core_news_lg` model.

   We create phrase lists P_c^{html} and P_c^{pdf}, each containing phrases from respective versions.

2. General Pre-Processing

   For each phrase: (1) Remove newlines and tabs; (2) Remove tag entities (e.g., ` `); (3) Reduce consecutive spaces, commas, periods to single characters; (4) Add period at end if missing.

3. Length Filtering

   Discard single-word phrases and phrases with character count below μ, the mean character count from the respective source P_c^{x}.

4. Blacklist Filtering

   Remove phrases containing: (1) *javascript*; (2) *cookies*; (3) *expand_less*; (4) *content_copy*; or (5) *Garantir a estabilidade do poder de compra da moeda*.

   While terms (1) to (4) are related to web page elements and scripts, term (5) is the Brazilian Central Bank's motto, which often appears in the minutes and is not relevant for sentiment analysis.

Finally, we compare phrase counts between sets and select the one with more phrases (PDF if equal to ensure superior quality), creating the selected set F_{d_i} for each date d_i. The set F contains all sets F_{d_i}.

3.1.3 Phrase Selection

We flatten F into list L of tuples (phrase, date).

We perform dense passage retrieval using semantic similarity filtering. We compute embeddings with **Qwen3-Embedding-0.6B** (Zhang et al., 2025) and retain phrases with cosine similarity > 0.6 to "inflation". We use PyTorch for GPU acceleration, pandas for manipulation, and scikit-learn for similarity.

The final dataset F^{infl} contains 9,378 inflation-related phrases across 251 dates (37.4 phrases per date).

3.2 Creating the Sentiment Datasets

3.2.1 LLM Evaluation Dataset

We evaluated phrase sentiment using nine LLMs from different companies:

1. *openai/gpt-5*
2. *anthropic/claude-sonnet-4*
3. *google/gemini-2.5-pro*
4. *x-ai/grok-4-fast*
5. *openai/gpt-oss-120b*
6. *meta-llama/llama-4-maverick*
7. *google/gemma-3-27b-it*
8. *microsoft/phi-4*
9. *deepseek/deepseek-chat-v3.1*

For each model, we made one independent request per phrase in F^{infl}, without prior context.

The prompt, formulated in Brazilian Portuguese by economist Cézio Luiz Ferreira Junior, explained the task and appended the phrase:

> **DEFINIÇÃO DE OTIMISMO:** Ocorre quando as projeções indicam que a inflação ficará abaixo da meta ou dentro do intervalo de tolerância com folga. Isso pode sinalizar que o Banco Central vê espaço para reduzir juros ou manter uma política monetária mais acomodatícia.
>
> **DEFINIÇÃO DE PESSIMISMO:** Ocorre quando as projeções apontam para inflação acima da meta ou próxima do teto do intervalo de tolerância. Isso sugere preocupação com pressões inflacionárias e pode justificar uma política monetária mais restritiva.
>
> **AVALIE A FRASE COMO:** O para OTIMISTA, N para NEUTRA, P para PESSIMISTA. SUA RESPOSTA DEVE SER APENAS UMA LETRA, SEM QUALQUER OUTRO TEXTO.
>
> **FRASE A SER AVALIADA:** ««PHRASE»»

Models classify phrases as optimistic, neutral, or pessimistic. Responses (O, N, P) are converted to 1, 0, -1; unparseable responses labeled -2 (rare occasions).

We use OpenRouter API for unified access. We determine token limits by testing on the first date's phrases; if any receives -2, we double the limit and repeat testing until all responses are successful. Table 1 shows the final token limits used.

We discarded evaluations not equal to 1 or -1. We concatenated results into sets E_m for each model, containing tuples (phrase, date, sentiment). The set E_{Models} contains all E_m.

Model	Token Limit
openai/gpt-5	1024
openai/gpt-oss-120b	512
google/gemini-2.5-pro	128
google/gemma-3-27b-it	8
deepseek/deepseek-chat-v3.1	4
others	1

Table 1: Token limits per LLM model.

3.3 Human Evaluation Dataset

We created three human evaluation datasets:

1. *Open*

   A website with O/N/P selection for randomly selected phrases from F^{infl}, limited to 10 phrases per browser per 24h. Distributed to economics graduate students at USP and Unicamp. Result: E_{Open} with 278 tuples.

2. *Specialist*

   A subset $F^{infl-350}$ of 350 random phrases from F^{infl}, with date labels Base64-encoded to prevent bias. Labeled by economist Cézio Luiz Ferreira Junior as: 1 (optimistic), 0 (neutral), -1 (pessimistic), -2 (non-related), -3 (not understood). Result: $E_{Specialist}$ with 350 tuples.

3. *Consolidated*

   $F^{infl-350}$ re-analyzed by the specialist and two additional professors together, discussing each phrase to reach consensus. Result: $E_{Consolidated}$ with 220 tuples.

Again, we discarded evaluations not equal to 1 or -1. for all methods. Set E_{Humans} contains all E_h.

3.4 Testing Inflation Prediction Performance

We test two common inflation prediction models: (1) **ARIMA** and (2) **LSTM**.

The goal is to check whether adding sentiment variables from LLM evaluations reduces RMSE compared to historical inflation data alone, and whether bias correction from human evaluations further improves performance.

3.4.1 Creating the Input Datasets

For each set of the power set of E_{Models}, except for the empty one, we will concatenate the tuples of the selected E_m sets into a single set named U_i.

For each U_i created, we will create j more tuples in the form (U_i, V_j), where V_j is one of the three human evaluation datasets in E_{Humans}.

For each tuple (U_i, V_j) created, we will create k more tuples in the form (U_i, V_j, eq_k), where eq_k is one of the equations to be used for bias correction later.

The tuple (U_i, V_j, eq_k) represents the sentiment evaluations from the selected LLM models combined with the human evaluation dataset V_j for bias correction using equation eq_k.

The possible equation forms for eq_k are: linear ($x + a$), affine ($bx + a$), quadratic ($cx^2 + bx + a$), and cubic ($dx^3 + cx^2 + bx + a$).

For each tuple (U_i, V_j, eq_k), we create three different input datasets for inflation prediction models, each one of them will provide a list of tuples in the form of ($Inflation, Sentiment$):

1. *Only Inflation (Baseline)*

   IPCA monthly (Series 433) (Banco Central do Brasil, 2025b).

   The sentiment variable is set to 0 for associated inflation values.

2. *Inflation + Sentiment (Without Correction)*

   IPCA monthly (Series 433) + Sentiment variable created as an average grade per date of the evaluations in U_i (interpolated by cubic spline and fitted to the available IPCA dates)

3. *Inflation + Sentiment (With Correction)*

   IPCA monthly (Series 433) + Sentiment variable created as an average grade per date of the evaluations in U_i (interpolated by cubic spline and fitted to the available IPCA dates) corrected based on the bias measured from V_j.

   The correction process works as follows:

   First, both LLM sentiment scores from U_i and human evaluations from V_j are averaged by date and interpolated using cubic spline to create continuous daily time series.

   Then, we find a single set of parameters of the transformation equation eq_k that when applied to all dates individually minimizes the mean squared error (MSE).

   The equation is applied per date with the variable x representing the average LLM sentiment score in that date, and the resulting value representing the bias-corrected sentiment score.

   The optimization uses gradient descent with the Adam optimizer (Kingma and Ba, 2014) (1000 epochs, learning rate 0.01) implemented in PyTorch.

   These optimized parameters are then applied to the equation to transform the LLM sentiment score for each individual date in U_i, producing bias-corrected values aligned with human judgment from V_j.

Finally, for each tuple (U_i, V_j, eq_k) created, we have 3 new associated lists of tuples in the form of ($Inflation, Sentiment$), each called IN_{ijkm} where i is the LLM model combination used; j is the human evaluation dataset used for bias correction; k is the equation type used for bias correction; and $m \in$ {*Baseline*, *Without Correction*, or *With Correction*.}

The set IN contains all sets IN_{ijkm}.

3.4.2 Running the Tests

Looking at the IN, we see that this approach involves repetition of IN_{ijkm} datasets since, for example, *Baseline* is the same for all tuples (U_i, V_j, eq_k).

While this is bad from a computational efficiency perspective, it provides a control for every experiment: *Baseline* should be a control *Without Correction* and *With Correction*, while *Without Correction* should be a control for *With Correction*.

For each IN_{ijkm} in IN, we run both ARIMA and LSTM (Hochreiter and Schmidhuber, 1997) inflation prediction models on the respective dataset with a 70/30 train/test split.

We employ ARIMA with sentiment as exogenous variable (Moslemi et al., 2024) using walk-forward validation, and LSTM with 5000 neurons trained with NAdam optimizer (Dozat, 2016) (learning rate 0.001, max 10,000 epochs, early stopping patience 10). The highly parameterized LSTM aligns with recent insights on double descent (Schaeffer et al., 2023), where increased complexity improves generalization in the overparameterized regime. Both models are evaluated using Root Mean Squared Error (RMSE).

In total, we conducted 36,792 tests: $(2^9 - 1)$ LLM combinations $\times$ 3 human datasets $\times$ 4 equation types $\times$ 3 dataset types $\times$ 2 models.

4 Results

In Figure 1 it is possible to observe that, despite some variability, all the LLMs' sentiment follow

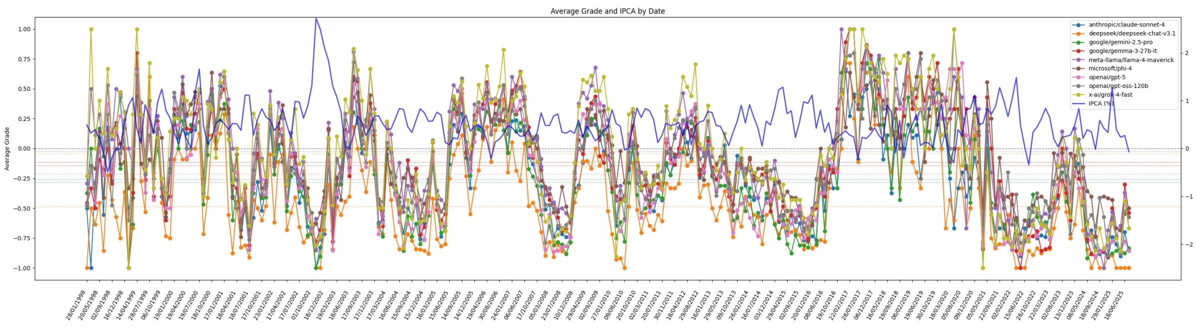

Figure 1: Average LLM sentiment grade by date and model (with IPCA inflation).

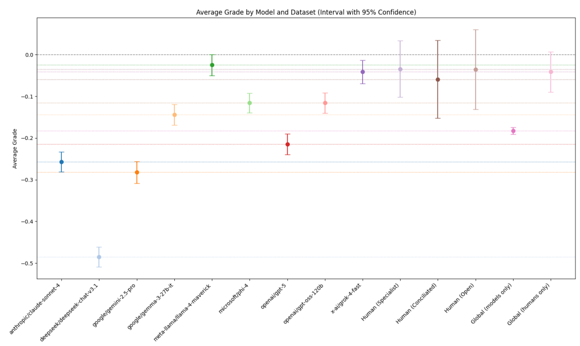

Figure 2: Average grade and confidence intervals by dataset at 95% confidence level.

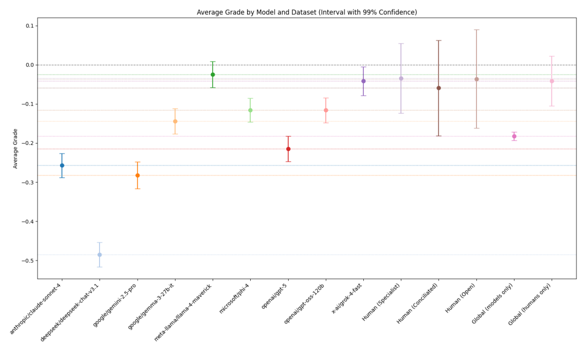

Figure 3: Average grade and confidence intervals by dataset at 99% confidence level.

a similar trend over time, with peaks and valleys occurring around the same dates. Even the exceptional cases such as Deepseek-chat-v3.1, which shows a significant lower average than the other models, still follows the same general trend.

This suggests that models can indeed capture market sentiment dynamics, but the bias should be taken into account when using their outputs as sentiment indicators. This also indicate that correcting for bias might improve the models' performance in downstream tasks such as inflation prediction.

The inflation in the graph appears to be stable, but it is noticeable that some drops in sentiment happen at the same time we have peaks in inflation, such as in 2002. Interestingly, that same correlation occurs in reverse in 2022 and also don't happen at all in some other periods we would expect it to happen such as 2008.

In figures 2, 3 and table 2 we can see the average grades and confidence intervals for each dataset used in the evaluation.

It is noticeable that we have a significant variation in the average grades assigned by different models, with all averages being slightly negative, including the human evaluated ones. This might indicate a general pessimistic bias in the COPOM minutes.

Grok-4-fask and **Llama-4-maverick** were the models with a bias closer to the human averages, while **Deepseek-chat-v3.1** was the furthest and also the most pessimistic by a large margin.

It is also noticeable that the confidence intervals are quite wide at human evaluated averages, even with lower standard deviation values, because they have a smaller number of samples. The global human evaluations are also more optimistic than the LLM ones.

In figure 4 we can see the comparison of the best model configurations for each of the six setups presented.

We can see that in most cases we have a small improvement when using sentiment grades compared to the baseline model without sentiment. While this is always true in the ARIMA setups, in the LSTM setups the results are mixed and much more unstable.

A fascinating insight is that the most frequent best models are **Grok-4-fask** and **Llama-4-maverick**, which were also the models with sentiment averages closer to the human evaluated ones.

Dataset	Average	Std. Dev.
Global		
Models only	−0.1826	0.9832
Humans only	−0.0413	0.7187
By LLM		
Claude Sonnet 4	−0.2572	0.9664
Deepseek Chat v3.1	−0.4851	0.8745
Gemini 2.5 Pro	−0.2823	0.9594
Gemma 3 27B IT	−0.1442	0.9896
Llama 4 Maverick	−0.0248	0.9998
Phi 4	−0.1158	0.9933
GPT-5	−0.2146	0.9768
GPT-OSS-120B	−0.1160	0.9933
Grok 4 Fast	−0.0415	0.9992
By Human		
Specialist	−0.0343	0.6459
Conciliated	−0.0591	0.7030
Open	−0.0360	0.8142

Table 2: Average sentiment grades and standard deviations.

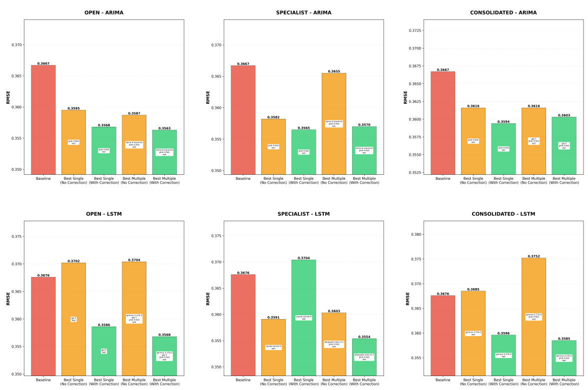

Figure 4: RMSE comparison across six different model configurations.

This suggests that altering the bias towards a more human-like sentiment might improve the models' performance in inflation prediction.

In table 3, we can observe the RMSE reduction percentages when including sentiment with and without correction compared to baseline (only inflation), an average of all our 36,792 tests.

As we can see we had a small improvement in all configurations, with ARIMA models benefiting the most from the inclusion of sentiment overall.

While ARIMA models observed a reduction in predition performance when using corrected sentiment grades, LSTM models saw an improvement.

Model	Uncorrected	Corrected
LSTM	0.16%	0.23%
ARIMA	1.20%	0.73%

Table 3: RMSE reduction across different models.

5 Conclusion

It is safe to safe that the inclusion of sentiment analysis in time series forecasting models has demonstrated a measurable improvement in predictive accuracy, even if modest.

The ARIMA models, in particular, benefited significantly from the integration of sentiment data, suggesting that these models are more adept at leveraging qualitative information to enhance their forecasts. The LSTM models also showed improvement to a lesser extent.

We also observed that models whose sentiment evaluations were closer to human assessments tended to perform better in forecasting tasks. This finding highlights the importance of aligning model biases with human perspectives.

References

Banco Central do Brasil. 2025a. Copom minutes – chronological listing. https://www.bcb.gov.br/publicacoes/atascopom/cronologicos.

Banco Central do Brasil. 2025b. Ipca monthly series 433 – api. https://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados?formato=json.

Timothy Dozat. 2016. Incorporating nesterov momentum into adam. Technical report, Stanford University. ICLR 2016 Workshop track.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Published at ICLR 2015.

Zahra Moslemi, Logan Clark, Sarah Kernal, Samantha Rehome, Scott Sprengel, Ahoora Tamizifar, Shawna Tuli, Vish Chokshi, Mo Nomeli, Ella Liang, Moury Bidgoli, Jeff Lu, Manish Dasaur, and Marty Hodgett. 2024. Comprehensive forecasting of california's energy consumption: A multi-source and sectoral analysis using arima and arimax models. *arXiv preprint arXiv:2402.04432*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*. BioNLP@ACL2019.

Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. 2023. Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle. *arXiv preprint arXiv:2303.14151*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.