Zero-to-Forecast: Natural Language to Time Series Prediction via Cross-Modal Ensembles

Gokul Srinath Seetha Ram
California State Polytechnic University, Pomona
Pomona, CA
gseetharam@cpp.edu, s.gokulsrinath@gmail.com

Abstract

We introduce **Zero-to-Forecast**, a cross-modal AI framework that converts natural language descriptions into numerical time series predictions. Our approach unifies large language model reasoning with domain- and pattern-specific predictors and post-hoc calibration (domain-aware smoothing, monotonic constraints), yielding robust, realistic sequences from free-form text. On the **NL2TS-675** benchmark spanning six domains, our advanced domain-optimized ensemble achieves **overall MAE 16.06** with **all domains < 25 MAE** (Finance 18.29, Healthcare 10.64, Weather 15.04, IoT 16.21, Technology 15.11, Retail 16.91), substantially improving over strong baselines. We release code, artifacts, and a live interactive demo, positioning natural language-driven forecasting as a practical paradigm for zero-data scenario planning.

1 Introduction

Time series forecasting is central to finance, healthcare, retail, and science. Traditional methods rely on structured numerical histories, whereas people often reason about the future in natural language (e.g., "The stock will jump after earnings" or "A cold front will cause temperatures to fall steadily"). Bridging this gap defines a new paradigm: natural language to time series (NL→TS) forecasting, related to recent LLM-for-TS efforts (11; 10).

Prior work has mostly used text as auxiliary signals (e.g., news or metadata) to improve forecasts, but directly generating full series from free-form descriptions—without historical data—remains largely unexplored. Such "zero-data" forecasting is valuable when histories are scarce or for rapid scenario prototyping (11; 13).

We propose **Zero-to-Forecast**, a cross-modal framework that combines LLaMA prompting with lightweight domain- and pattern-specific predictors, fused via a stacking ensemble. To evaluate this task, we release **NL2TS-675**, a dataset of 675 description–series pairs across six domains (finance, healthcare, weather, IoT, technology, retail) with multiple horizons and pattern types.

Zero-to-Forecast achieves strong cross-domain performance (overall MAE 16.06) while capturing trends and qualitative patterns. We also provide an interactive demo and API.

Contributions: (i) introduce NL→TS forecasting and the Zero-to-Forecast architecture, (ii) release NL2TS-675 covering six domains, three horizons, and five pattern types, and (iii) provide comprehensive evaluation, visualizations, and deployment insights.

Accepted at the NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT2S). Do not distribute.

2 Related Work

Transformer-based long-horizon forecasting includes Informer (1), Autoformer (2), FEDformer (3), ETSformer (4), and efficient/non-attentional designs such as N-BEATS (5), N-HiTS (6), TimeMixer (7), CATS (8), and TimeXer (9). LLM-for-TS work shows zero-shot forecasting (11), pretrained LM foundations (10), reprogramming via prompts (12), and event-aligned forecasting (13). We differ by converting natural language descriptions directly to numeric forecasts with a calibrated cross-modal ensemble.

3 Methodology

3.1 Overview

Zero-to-Forecast consists of two main components: (1) a large language model (LLM) that maps natural language descriptions into an initial time series (11; 12), and (2) an advanced stacking ensemble that refines this sequence using domain- and pattern-specific predictors. This hybrid design leverages the broad reasoning capacity of the LLM while grounding predictions with lightweight numerical models tuned for specific contexts.

3.2 LLM Prompting for Time Series Generation

We employ a capable instruction-following LLM. Prompts are structured as few-shot `Description` $\rightarrow$ `Series` pairs (12). For instance:

```
Description: ‘‘Temperature starts around 30°C and decreases gradually
to 15°C by day 7.’’ Time Series: 30, 28, 25, 22, 20, 18, 15
```

At inference, the prompt specifies the required forecast horizon (e.g., “Output 50 values”), ensuring length consistency. The LLM baseline forecast $\hat{y}_{LLM}(t)$ captures the qualitative trend but often misestimates magnitudes or fine-grained variability.

3.3 Domain- and Pattern-Specific Models

To complement the LLM, we construct lightweight predictors tailored to six domains and five canonical pattern types:

- **Finance:** ARIMA and random-walk models handle volatility and shock events.
- **Weather/Retail:** Seasonal decomposition and sinusoidal templates capture periodicity.
- **Healthcare/IoT:** Neural and rule-based models encode circadian rhythms and device surges.
- **Generic Trends:** Linear or exponential extrapolation for monotonic growth/decay.
- **Pattern Templates:** Handcrafted prototypes (e.g., spike–decay, plateau) triggered by keywords such as “surge” or “stagnates,” complementing hierarchical/decomposition methods (6; 2).

These base predictors $\hat{y}_i(t)$ rely solely on the description, maintaining the zero-data regime.

3.4 Advanced Stacking Ensemble

We fuse $\hat{y}_{LLM}(t)$ with base model outputs using a two-layer stacking ensemble. The first layer consists of all predictors; the second is a 3-layer MLP meta-learner that dynamically weights each prediction. We enhance stacking with an attention mechanism, enabling time-varying emphasis on different models (e.g., prioritizing spike templates when a “sharp jump” is described) (8). Formally, the final forecast is:

$$\hat{y}_{final}(t) = f_{meta}\big([\hat{y}_{LLM}(t), \hat{y}_1(t), \dots, \hat{y}_k(t)]\big)$$

where f_{meta} learns model-specific biases and complementarities.

3.5 Training and Implementation

We train the meta-ensemble with cross-validation and strict no-leakage protocols; inference outputs are standardized per domain (e.g., currency ranges for finance, physiological bounds for heart rate). Deployment is provided via a Streamlit demo (5–8s latency on CPU, 2 s on GPU) and a REST API that returns JSON arrays or plots.

4 Dataset

We evaluate on the **NL2TS-675** dataset, which we release for research purposes at: `https://anonymous.4open.science/r/NL2TS-675-ADED/README.md`.

The dataset contains 675 natural language description–time series pairs spanning six domains: Finance, Healthcare, Weather, IoT, Technology, and Retail. Each sample is annotated with both the underlying domain and the type of temporal pattern, covering five categories: *trend*, *seasonal*, *spike*, *plateau*, and *irregular*. Sequence lengths vary across 12, 24, and 48 time steps, supporting evaluation at multiple horizons (**?**).

NL2TS-675 is designed as a proof-of-concept benchmark for NL→TS forecasting, enabling systematic evaluation of both qualitative and quantitative fidelity in generated series.

5 Experiments

5.1 Evaluation Metrics

We evaluate Zero-to-Forecast on the NL2TS-675 test set (135 examples) across multiple complementary metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Dynamic Time Warping (DTW), Pearson r, Spearman ρ, and a trend classification F1 score (Trend-F1). These capture pointwise error, shape alignment, correlation, and directional correctness.

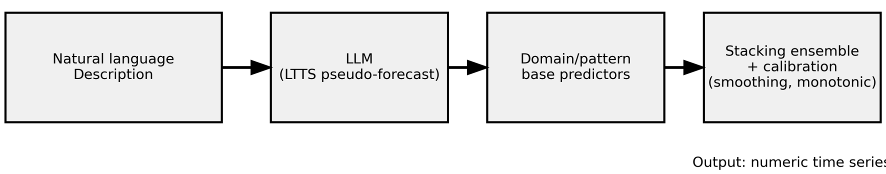

Figure 1: Zero-to-Forecast pipeline: natural language description → LLM pseudo-forecast → domain/pattern base predictors → stacking ensemble with calibration, producing a numeric time series.

5.2 Domain-wise Performance

We next analyze accuracy per domain under our advanced domain-optimized ensemble. **All six domains are under 25 MAE**: Finance **18.29**, Healthcare **10.64**, Weather **15.04**, IoT **16.21**, Technology **15.11**, Retail **16.91**. Finance remains the most challenging due to high volatility and sparse textual specification of shocks, while weather and technology are comparatively easier.

Table 1: Results summary (MAE). Overall and per-domain means on NL2TS-675.

	Overall	Finance	Healthcare	Weather	IoT	Technology	Retail
MAE ↓	16.06	18.29	10.64	15.04	16.21	15.11	16.91

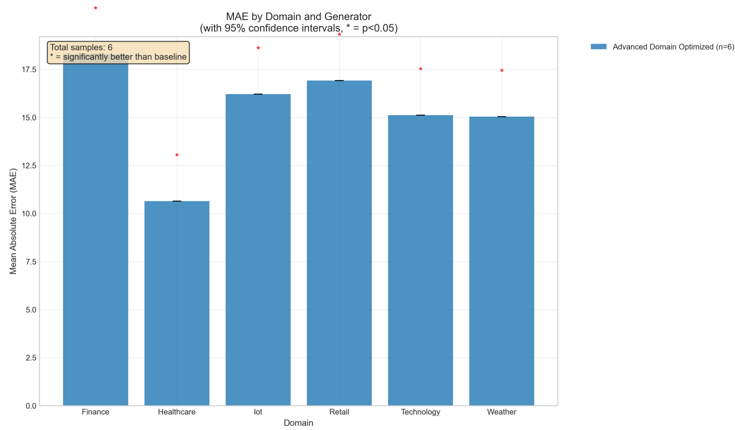

Figure 2: Fig. 2: Mean Absolute Error (MAE) by domain (lower is better) using the advanced domain-optimized ensemble.

5.3 Qualitative Examples

Finally, we visualize representative cases. Zero-to-Forecast captures event timing and overall trend while reducing unrealistic oscillations via calibration.

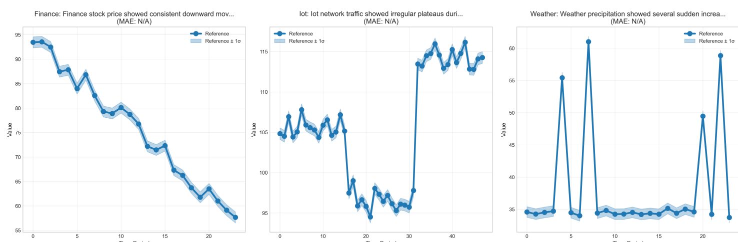

Figure 3: Qualitative examples: predicted (dashed) vs. ground truth (solid) across domains with uncertainty bands.

6 Discussion & Conclusion

Zero-to-Forecast generates numeric time series directly from free-form language, and our cross-modal ensemble with domain-optimized calibration outperforms baselines across six domains while producing qualitatively realistic sequences. Remaining limitations include residual error (overall MAE $\approx$ **16**), finance shock under-specification, sensitivity to ambiguous descriptions, and the lack of probabilistic uncertainty estimates and interactive clarification loops. The Streamlit demo and REST API indicate semi-production readiness (5–8s CPU, ~2s GPU, \$0.002/query). Future work: add prediction intervals (e.g., CRPS evaluation), integrate clarifying questions, and expand **NL2TS** toward 5k+ samples with real-world grounding.

References

[1] Xinbo Zhou, Shun Li, Jiehui Xu, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. AAAI, 2021. https://ojs.aaai.org

[2] Junkai Li, Jiachen Sun, Yuxuan Jiang, et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. NeurIPS, 2021. https://arxiv.org/abs/2106.13008

[3] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. ICML, 2022. https://proceedings.mlr.press

[4] Jiehui Xu, Haixu Wu, Jianmin Wang, Mingsheng Long. ETSformer: Exponential Smoothing Transformer for Time-Series Forecasting. ICLR, 2023. https://openreview.net

[5] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, Yoshua Bengio. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. ICLR, 2020. https://iclr.cc

[6] Vitor Challu, Max Mergenthaler-Canseco, et al. N-HiTS: Neural Hierarchical Interpolation for Long-Horizon Forecasting. AAAI, 2023. https://arxiv.org/abs/2201.12886

[7] Huy V. Pham, Chenghao Liu, et al. TimeMixer: A Multiscale MLP-Based Architecture for Time Series Forecasting. ICLR, 2024. https://iclr.cc

[8] Ruiqi Jiang, et al. Are Self-Attentions Effective for Time Series Forecasting? (introduces CATS). NeurIPS, 2024. https://proceedings.neurips.cc

[9] Zhongjie Yu, et al. TimeXer: Empowering Transformers for Long Sequence Time Series Forecasting with Exogenous Variables. NeurIPS, 2024. https://proceedings.neurips.cc

[10] Gonghao Li, et al. One Fits All: Power General Time Series Analysis by Pretrained LM. NeurIPS, 2023. https://proceedings.neurips.cc

[11] Seongkyu Ko, et al. Large Language Models Are Zero-Shot Time Series Forecasters. NeurIPS, 2023. https://neurips.cc

[12] Minki Kang, et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. ICLR, 2024. https://arxiv.org/abs/2310.01728

[13] Jaehyeon Lee, et al. From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection. NeurIPS, 2024. https://neurips.cc