

Partisan Discourse on X: An Analysis of Political Polarization in Social Media

Ziv Barretto
Ashoka University

Agriya Yadav
Ashoka University

Kyra Chhetri
Ashoka University

Anirban Sen
Ashoka University

Abstract—
Index Terms—

I. INTRODUCTION

The rise of social media platforms has fundamentally transformed political communication and discourse. X (formerly known as Twitter) has emerged as a critical platform for political discussion, serving as a space where politicians, journalists, activists, and citizens engage in real-time conversations about political issues. However, this democratization of political discourse has coincided with growing concerns about political polarization and the formation of ideological echo chambers.

This paper investigates partisan discourse on X, examining how political identities and affiliations shape online communication patterns.

II. LITERATURE REVIEW

III. METHODOLOGY

A. Data Cleaning and Formatting

To analyze partisan discourse, we first established a ground-truth dataset of partisan affiliations. We utilized a dataset of tweets where the `retweet_author` was a known political figure. We employed a mapping file to assign a political party to each `retweet_author`.

We then classified these parties into two main blocs: “Ruling” (e.g., BJP, NDA, JDU) and “Opposition” (e.g., INC, TMC, SP). To determine the partisan leaning of the original authors (the “influencers” whose content was retweeted), we calculated a yearly polarity score (P_{yearly}) for each author based on the volume of retweets they received from politicians of each bloc:

$$P_{yearly} = \frac{R - O}{R + O} \quad (1)$$

where R is the count of retweets from Ruling party members and O is the count from Opposition members.

We then computed an average polarity score (P_{avg}) across all available years for each author. Based on this score, authors were classified as:

- **Pro-Ruling:** $P_{avg} \geq 0.5$
- **Pro-Opposition:** $P_{avg} \leq -0.5$
- **Neutral:** $-0.5 < P_{avg} < 0.5$

This classification allowed us to label the original tweets in our dataset with a “Pro-Ruling”, “Pro-Opposition”, or “Neutral” tag based on the affiliation of their author.

B. Fine-tuning and Stance Detection Model

To classify the stance of tweets towards specific entities (keywords), we employed a Few-Shot Stance Classification approach enhanced with Low-Rank Adaptation (LoRA). We utilized a pre-trained Large Language Model (LLM) as the base model.

A critical component of our approach is the dynamic retrieval of few-shot examples. For each target keyword (entity), we maintain a specific set of labeled examples (stored as JSON files). During inference, the system reads the keyword associated with a tweet and retrieves the corresponding few-shot examples from the database. If specific examples for a keyword are unavailable, the system falls back to a global set of examples. This ensures that the model receives contextually relevant guidance for each classification task.

The model was adapted using LoRA, a parameter-efficient fine-tuning technique that freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture. This allowed us to adapt the model to the specific task of political stance detection without the computational cost of full fine-tuning.

C. Stance Detection Process

The stance detection pipeline was designed to process tweets in batches. For each tweet, we constructed a prompt that included:

- 1) A task description defining stance classification (Support, Deny, Neutral, Unrelated).
- 2) A set of few-shot examples provided in JSON format to guide the model’s output structure.
- 3) The target entity (keyword) and the tweet text.

The model was constrained to generate a valid JSON object containing two keys: “stance” (the predicted label) and “reason” (a brief explanation). We employed a custom inference pipeline that handled tokenization, batched generation, and robust JSON parsing. To ensure consistency, the model’s output labels were normalized to a standard set: “supports” (mapped to “For”), “denies” (mapped to “Against”), “neutral”, and “unrelated”. This structured approach ensured high-quality, interpretable stance labels for our large-scale dataset.

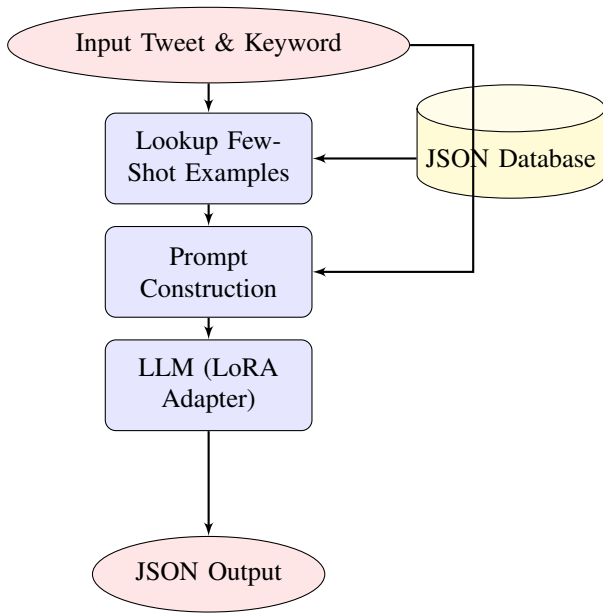


Fig. 1. Architecture of the Stance Detection System. The system retrieves keyword-specific few-shot examples to construct a prompt, which is then processed by the LoRA-adapted LLM to generate a structured JSON output.

IV. RESULTS

- A. Partisan Network Structure
- B. Content Characteristics
- C. Interaction Patterns
- D. Temporal Dynamics

V. DISCUSSION

VI. CONCLUSION

ACKNOWLEDGMENT

We would like to thank Ashoka University for supporting this research.

REFERENCES

- [1] Author, A. A., Author, B. B., and Author, C. C. "Title of the paper," *Journal Name*, vol. X, no. Y, pp. ZZ-ZZ, Month Year.
- [2] Author, D. D. "Title of another paper," in *Proceedings of Conference Name*, City, Country, Year, pp. XX-XX.
- [3] Author, E. E. and Author, F. F. "Yet another paper title," *Journal Name*, vol. X, no. Y, pp. ZZ-ZZ, Month Year.