Reproducibility Report: Benchmarking LLMs for Political Science — A United Nations Perspective

FTEC5660 Agentic AI for Business and FinTech

Name: Zhuocheng XU **Student ID:** 1155244383

GitHub Repository:
`https://github.com/wsnddddddddddddddd/FTEC5660`

Presentation Video:
`https://drive.google.com/file/d/1CY1Y2JzLyChl3_xgNqEXykv_bmKJ6jTx/view?usp=sharing`

March 2026

1 Project Summary

This report presents a reproducibility study of UNBench [1], a multi-stage benchmark built on United Nations Security Council (UNSC) records to evaluate large language models (LLMs) across political decision-making tasks. The paper was accepted as an oral presentation at AAAI 2026.

UNBench covers four interconnected tasks: (1) coauthor selection, (2) voting simulation, (3) draft adoption prediction, and (4) diplomatic statement generation. I chose to reproduce **Task 3: Draft Adoption Prediction**, a binary classification task where the model predicts whether a UNSC draft resolution will be adopted or not, based solely on the draft text.

The reproduction target is the performance of **DeepSeek-V3** on Task 3, as reported in Table 9 of the original paper. As a modification, I replaced DeepSeek-V3 with **Gemini-2.5-Pro** to investigate how model choice affects performance on political reasoning tasks.

The agentic nature of this system lies in its multi-step pipeline: the LLM must read a lengthy diplomatic document, reason about the political dynamics of 15 Council members including potential veto threats and support coalitions, and produce a structured binary prediction — a process requiring document-level comprehension and geopolitical reasoning beyond simple prompt-response.

2 Setup Notes

2.1 Environment

- **OS:** macOS (Apple Silicon)
- **Python:** 3.13.9 (Anaconda base environment)
- **Jupyter Notebook:** via Anaconda Navigator
- **Key packages:** `openai`, `scikit-learn`, `imbalanced-learn`, `tqdm`, `pandas`

2.2 Data

The public GitHub repository (`https://github.com/yueqingliang1/UNBench`) provides a 30-sample subset of Task 3 in `data/task3.json`. This subset contains 29 adopted (label = 1) and

1 not-adopted (label = 0) draft resolution. The full dataset used in the paper contains 1,978 drafts (1,880 adopted, 98 not adopted).

2.3 API Access

- **Reproduction:** DeepSeek-V3 via DeepSeek API (`deepseek-chat` model, endpoint: `https://api.deeps`
  accessed through the OpenAI-compatible Python client.
- **Modification:** Gemini-2.5-Pro via Google AI Studio (`gemini-2.5-pro` model, endpoint: `https://generativelanguage.googleapis.com/v1beta/openai/`).

2.4 Compute

All experiments were run locally. Each run of 30 API calls took approximately 45–60 seconds. Total API cost was negligible (under $1 USD combined for all experiments).

3 Reproduction Target and Metric Definitions

The reproduction target is **Table 9** of the original paper, specifically the DeepSeek-V3 row. The paper evaluates Task 3 using the following metrics (note: the evaluation code swaps labels 0 and 1, treating "not adopted" as the positive class):

- **Accuracy:** Fraction of correctly classified drafts.
- **AUC:** Area under the ROC curve.
- **Balanced Accuracy (Bal. ACC):** Average of recall for each class, accounting for class imbalance.
- **Precision / Recall / F1:** Binary metrics for the minority class (not adopted).
- **PR AUC:** Area under the Precision-Recall curve.
- **MCC:** Matthews Correlation Coefficient, a balanced metric for imbalanced classification.
- **G-Mean:** Geometric mean of sensitivity and specificity.
- **Specificity:** True negative rate for the majority class (adopted).

These metrics are computed by the authors' evaluation function `calculate_metrics()`, which was used without modification.

4 Reproduction Results

4.1 Experimental Setup

We used the authors' official notebook (`run_task3.ipynb`) with one controlled substitution: the original code accesses DeepSeek-V3 via the Together API platform, whereas we accessed DeepSeek-V3 directly through DeepSeek's own OpenAI-compatible endpoint. All other parameters were identical: the same prompt template, system message (`"You are a helpful assistant."`), `temperature=0.0`, and `max_tokens=5`.

4.2 Results

We ran the experiment twice. Both runs produced identical results, as expected under deterministic decoding.

4.3 Analysis

Our reproduction achieved perfect scores across all metrics. While this appears to diverge from the paper, the result is explainable and consistent with the paper's findings.

Table 1: Reproduction results: DeepSeek-V3 on Task 3.

Metric	Paper (n=1,978)	Ours (n=30)
Accuracy	0.966	1.000
AUC	0.724	1.000
Bal. ACC	0.724	1.000
Precision	0.828	1.000
Recall	0.453	1.000
F1	0.585	1.000
PR AUC	0.655	1.000
MCC	0.597	1.000
G-Mean	0.671	1.000

Sample size and composition. The 30-sample subset contains only 1 negative example versus 98 in the full dataset. Achieving a perfect score requires correctly classifying just one negative instance — a far easier task than identifying 98 negative cases among 1,978 drafts.

Consistency with the paper's high baseline. The paper reports 96.6% accuracy for DeepSeek-V3, indicating strong capability on this task. A perfect score on a small, easy subset is therefore unsurprising.

Deterministic outputs. With `temperature=0.0`, both runs produced identical predictions, confirming fully deterministic behavior and eliminating stochastic variance.

Key takeaway. The reproduction validates that the evaluation pipeline functions correctly and that DeepSeek-V3 performs strongly on Task 3. The perfect scores reflect the simplicity of the subset rather than an improvement over the paper. A definitive reproduction would require access to the full dataset.

5 Modification: Model Change

5.1 Description

We replaced **DeepSeek-V3** with **Gemini-2.5-Pro** as the underlying LLM. This is a model change — the first type of modification listed in the assignment guidelines. The modification was isolated: only the model identifier and API endpoint were changed. The prompt, system message, temperature, max tokens, and evaluation script remained identical.

One minor code adjustment was necessary: a `None` check was added before parsing the model's response (see Section 6), as Gemini's safety filters occasionally returned empty content.

5.2 Results

Due to non-deterministic behavior introduced by Gemini's safety filter (triggering the random fallback), we ran the experiment **5 times**.

5.3 Analysis

Severe and unstable performance. Gemini-2.5-Pro averaged only 55.3% accuracy with high variance (std = 0.12). The MCC fell to approximately zero (−0.014), indicating near-random predictive power.

Bimodal behavior. The 5 runs exhibit a bimodal pattern. In Runs 1 and 4, Gemini achieved Recall = 1.0 and Bal. ACC above 0.81, successfully identifying the single not-adopted draft. In Runs 2, 3, and 5, Recall dropped to 0.0, missing the not-adopted case entirely. This all-

Table 2: Per-run results for Gemini-2.5-Pro on Task 3.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Accuracy	0.6333	0.3667	0.6000	0.6667	0.5000
AUC	0.8103	0.1897	0.3103	0.8276	0.2586
Bal. ACC	0.8103	0.1897	0.3103	0.8276	0.2586
Precision	0.0833	0.0000	0.0000	0.0909	0.0000
Recall	1.0000	0.0000	0.0000	1.0000	0.0000
F1	0.1538	0.0000	0.0000	0.1667	0.0000
MCC	0.2274	−0.2274	−0.1413	0.2441	−0.1737

Table 3: Summary comparison: DeepSeek-V3 vs. Gemini-2.5-Pro.

Metric	DeepSeek-V3 (Ours, n=30)	Gemini-2.5-Pro (Mean ± Std)	Paper (n=1,978)
Accuracy	1.0000	0.5533 ± 0.1216	0.966
AUC	1.0000	0.4793 ± 0.3131	0.724
Bal. ACC	1.0000	0.4793 ± 0.3131	0.724
F1	1.0000	0.0641 ± 0.0879	0.585
MCC	1.0000	−0.0142 ± 0.2303	0.597

or-nothing pattern is driven by the interaction between Gemini's safety filters and the random fallback mechanism.

Safety filter interference. Unlike DeepSeek-V3, which returned valid responses for all 30 drafts, Gemini-2.5-Pro frequently returned `None` content — likely triggered by geopolitical language in UNSC resolutions (sanctions, armed conflicts, weapons embargoes). These null responses were handled by the random fallback, explaining both the degraded accuracy and the high run-to-run variance.

Implications for model selection in political NLP. This result demonstrates that model choice involves more than general capability. DeepSeek-V3's more permissive content policy makes it more suitable for political science benchmarks, while Gemini's aggressive safety filtering creates practical barriers to legitimate academic and policy-analysis tasks — not because Gemini lacks reasoning ability, but because it often refuses to engage with the input.

6 Debug Diary

6.1 Blocker 1: API Client Incompatibility

Problem: The original code uses the `Together` Python library to access DeepSeek-V3 via the TogetherAI platform. This library is not compatible with DeepSeek's own API or Gemini's API.

Resolution: Replaced the `Together` client with the `OpenAI` Python client, which provides a unified interface compatible with both DeepSeek and Gemini endpoints. Only the `base_url` and `api_key` parameters needed to be changed:

```
from openai import OpenAI
client = OpenAI(
    api_key="...",
    base_url="https://api.deepseek.com"  # or Gemini URL
)
```

6.2 Blocker 2: Gemini NoneType Error

Problem: When switching to Gemini-2.5-Pro, the function crashed with `AttributeError: 'NoneType' object has no attribute 'strip'`. The model returned `None` for `message.content` on certain inputs, likely due to safety filters blocking UN resolution content.

Resolution: Added a `None` check before parsing the response:

```
content = response.choices[0].message.content
if content is None:
    results.append(random.choice([0, 1]))
    continue
result = content.strip().lower()
```

This fallback is consistent with the original code's handling of unparseable outputs and revealed an important finding about Gemini's safety filter behavior.

6.3 Blocker 3: Indentation Error

Problem: After adding the `None` check, an `IndentationError` occurred due to misaligned code when pasting the fix into Jupyter Notebook.

Resolution: Replaced the entire cell content with properly indented code. A minor issue but a common pitfall when editing Python in notebook environments.

7 Conclusions

7.1 What Is Reproducible

- The evaluation pipeline is fully functional and easy to run. The authors' code is clean, well-structured, and requires minimal setup.
- DeepSeek-V3 performs strongly on Task 3, achieving perfect classification on the 30-sample subset. This is consistent with the paper's reported accuracy of 96.6%.
- The deterministic behavior under `temperature=0.0` is confirmed: two independent runs produced identical results.
- The API-switching approach (Together → DeepSeek direct → Gemini) works seamlessly thanks to OpenAI-compatible interfaces, demonstrating good portability of the evaluation pipeline.

7.2 What Is Not Fully Reproducible (and Why)

- **Exact metric values cannot be reproduced** with the public 30-sample subset. The paper's metrics are computed on 1,978 drafts (with 98 negative cases), while the subset contains only 1 negative case. This makes the subset too easy and too imbalanced for meaningful metric comparison.
- **The full dataset is not included in the repository.** While it is available via Google Drive, the repository README could more clearly indicate that the 30-sample files are representative samples, not the full evaluation set.

7.3 Key Lessons and Recommendations

For future users of UNBench:

- Download the full dataset from Google Drive for meaningful metric comparison. The 30-sample subset is useful for pipeline testing but insufficient for evaluation.
- When switching models, be aware that safety filters vary significantly across providers. Gemini's content restrictions make it unreliable for political NLP tasks, while DeepSeek has fewer restrictions in this domain.

- The OpenAI-compatible API format enables easy model swapping, making this benchmark portable across different LLM providers.

For the broader reproducibility community:

- Safety alignment can inadvertently harm performance on legitimate academic tasks. Benchmark designers should consider testing across models with different safety policies.
- When releasing evaluation subsets, authors should document the subset's composition relative to the full dataset and note any limitations for metric comparison.
- Even on small subsets, reproducibility exercises are valuable: they verify pipeline correctness, reveal model-specific behaviors (such as safety filter interference), and generate new insights beyond the original paper's scope.

References

[1] Liang, Y., Yang, L., Wang, C., Xia, C., Meng, R., Xu, X., Wang, H., Payani, A., and Shu, K. (2026). Benchmarking LLMs for Political Science: A United Nations Perspective. *Proceedings of the AAAI Conference on Artificial Intelligence.* `https://arxiv.org/abs/2502.14122`