

LINGUISTAGENT: A REFLECTIVE MULTI-MODEL PLATFORM FOR AUTOMATED LINGUISTIC ANNOTATION

Bingru Li

University of Birmingham

bx1426@student.bham.ac.uk

ABSTRACT

Data annotation remains a significant bottleneck in the Humanities and Social Sciences, particularly for complex semantic tasks such as metaphor identification. While Large Language Models (LLMs) show promise, a significant gap remains between the theoretical capability of LLMs and their practical utility for researchers. This paper introduces **LinguistAgent**, an integrated, user-friendly platform that leverages a reflective multi-model architecture to automate linguistic annotation. The system implements a dual-agent workflow, comprising an *Annotator* and a *Reviewer*, to simulate a professional peer-review process. LinguistAgent supports comparative experiments across three paradigms: Prompt Engineering (Zero/Few-shot), Retrieval-Augmented Generation, and Fine-tuning. We demonstrate LinguistAgent’s efficacy using the task of metaphor identification as an example, providing real-time token-level evaluation (Precision, Recall, and F_1 score) against human gold standards. The application and codes are released on <https://github.com/Bingru-Li/LinguistAgent>.

1 INTRODUCTION

Linguistic annotation often requires deep contextual and logical reasoning. A long-standing case is the identification of metaphors in discourse Johnson & Lakoff (1980); Cameron (2003; 2010b); Steen (2009); Cameron (2010a); Littlemore & Turner (2020); Fuoli et al. (2022); Turner & Littlemore (2023), which suffers from intensive labor. Both the Metaphor Identification Procedure (MIP) Group (2007); Steen et al. (2010) and the Procedure for Identifying Metaphorical Scenes (PIMS) Johansson Falck & Okonski (2023) involve comparing the contextual meaning of a word with its basic, more concrete meaning. Manually applying these protocols to large-scale corpora is time-consuming.

Recent benchmarks Ge et al. (2023); Tian et al. (2024) suggest that Large Language Models (LLMs) can achieve near-human reliability. However, a significant gap remains between the theoretical capability of LLMs and their practical utility. Current computational approaches typically rely on prompt-based approaches (Puraivan et al. (2024); Liang et al. (2025); Hicke & Kristensen-McLachlan (2024)), which often oversimplifies the multi-stage reasoning inherent in protocols like MIP. Conversely, more sophisticated pipelines (Fuoli et al. (2025)) often require extensive programming knowledge, making them inaccessible to non-expert researchers. **LinguistAgent** bridges this gap by providing a no-code, multi-agent environment designed specifically for large-scale data annotation and benchmarking.

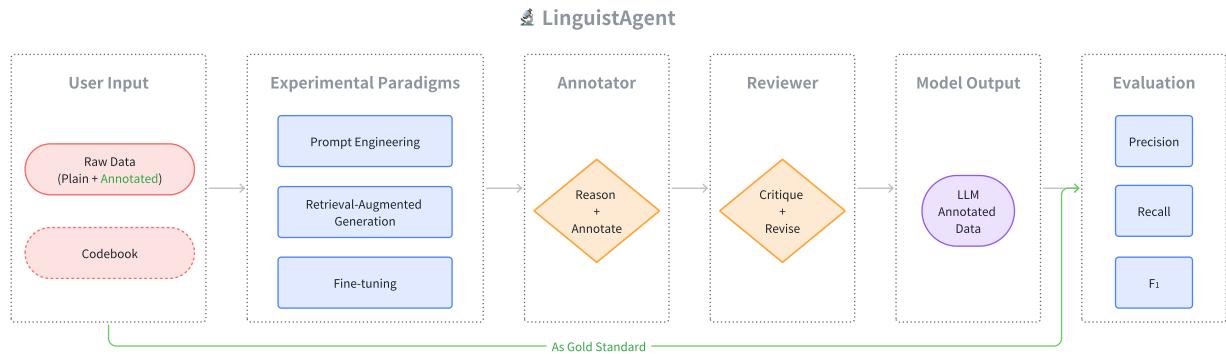


Figure 1: The architecture of LinguistAgent.

LinguistAgent employs a **Reflective Multi-Agent Workflow** to mitigate hallucinations and improve annotation accuracy. Fig. 1 illustrates the architecture of the system, where the user is expected to input the raw data (including plain text and human annotated text as gold standards) and the codebook (optional). Under a specific experimental paradigm, i.e., Prompt Engineering, Retrieval-Augmented Generation (RAG), and Fine-tuning, the Annotator LLM will code the data for the first round, and then the Reviewer LLM will review and revise when necessary. The final LLM annotated data will be compared against the input gold standards on such evaluation metrics as precision, recall, and F_1 . The default user interface (UI) is shown in Fig. 2.

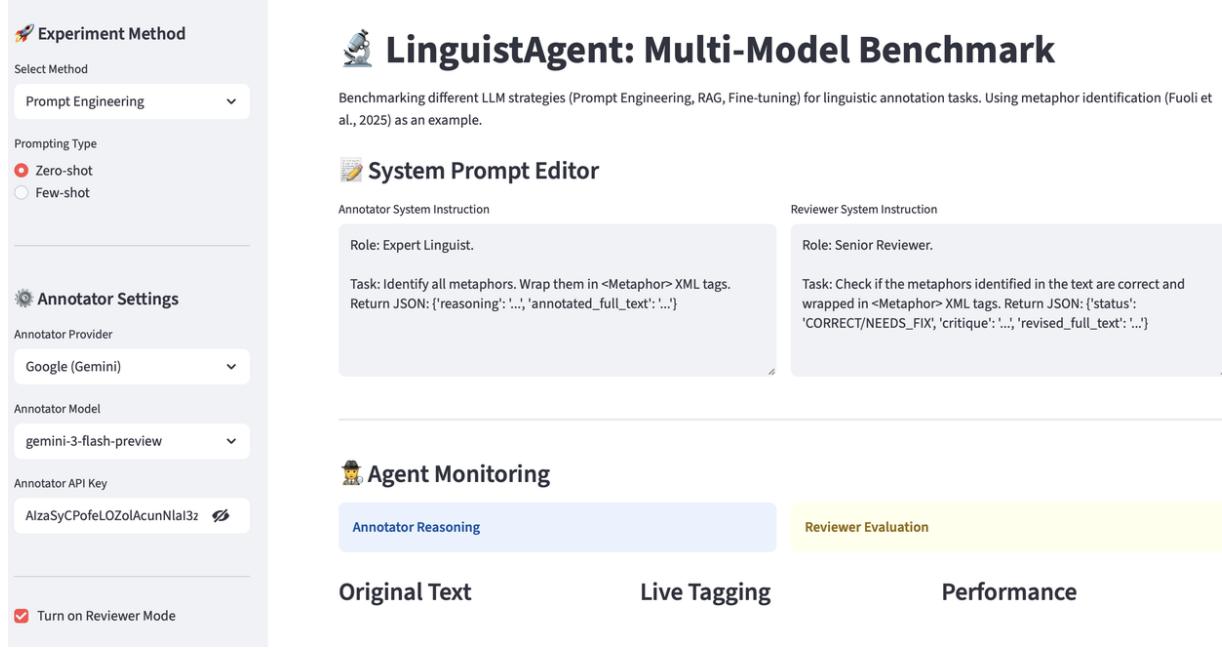


Figure 2: The user interface of LinguistAgent.

2 THE ARCHITECTURE OF LINGUISTAGENT

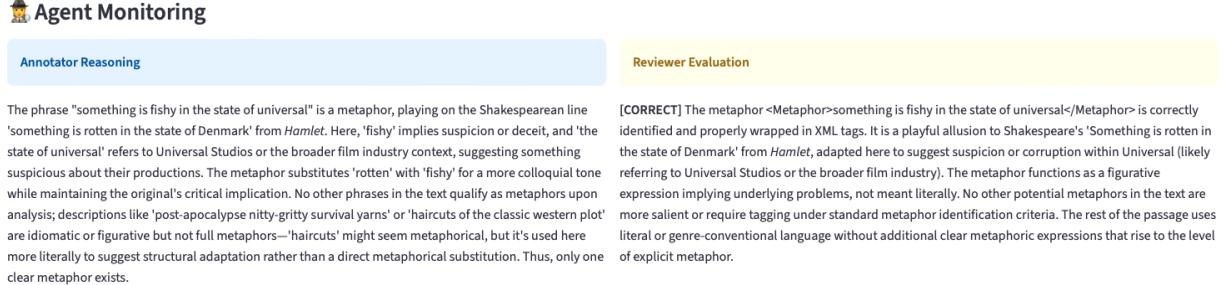


Figure 3: The reasoning of the Annotator and the critique of the Reviewer.

2.1 THE ANNOTATOR AGENT

The Annotator acts as the primary executor. It receives the raw text and a specialized *Codebook* (under the RAG experiment). Its task is to perform sequence labeling by wrapping target expressions in XML tags (e.g., `<Metaphor>...</Metaphor>`). The agent is instructed to provide a “Reasoning” field to justify its choices based on the provided protocol, as shown in Fig. 3.

2.2 THE REVIEWER AGENT

The Reviewer acts as a senior supervisor, and can be activated by turning on Reviewer Mode. It receives the original text and the Annotator’s output. It evaluates the labels against the Codebook, identifying false positives or missed instances. If discrepancies are found, the Reviewer provides a “Critique” and generates a “Revised Text,” completing the self-correction loop (see Fig. 3).

2.3 EXPERIMENTAL PARADIGMS

The platform integrates three distinct LLM application strategies to allow researchers to find the optimal balance between cost and accuracy:

- **Prompt Engineering:** Supports *Zero-shot* (instruction only) and *Few-shot* (instruction + examples) configurations to test the model’s in-context learning capabilities.
- **RAG (Full-Context):** Integrates the entire external codebook into the system instruction. Leveraging the ultra-long context of Gemini 3 Team (2025) and Qwen3 Yang et al. (2025) models, LinguistAgent ensures that the models have access to the complete linguistic theory without fragmentation.
- **Fine-tuning:** Allows users to input a *Tuned Model ID*. This enables the evaluation of specialized models trained on domain-specific data, often resulting in higher performance with lower token latency.

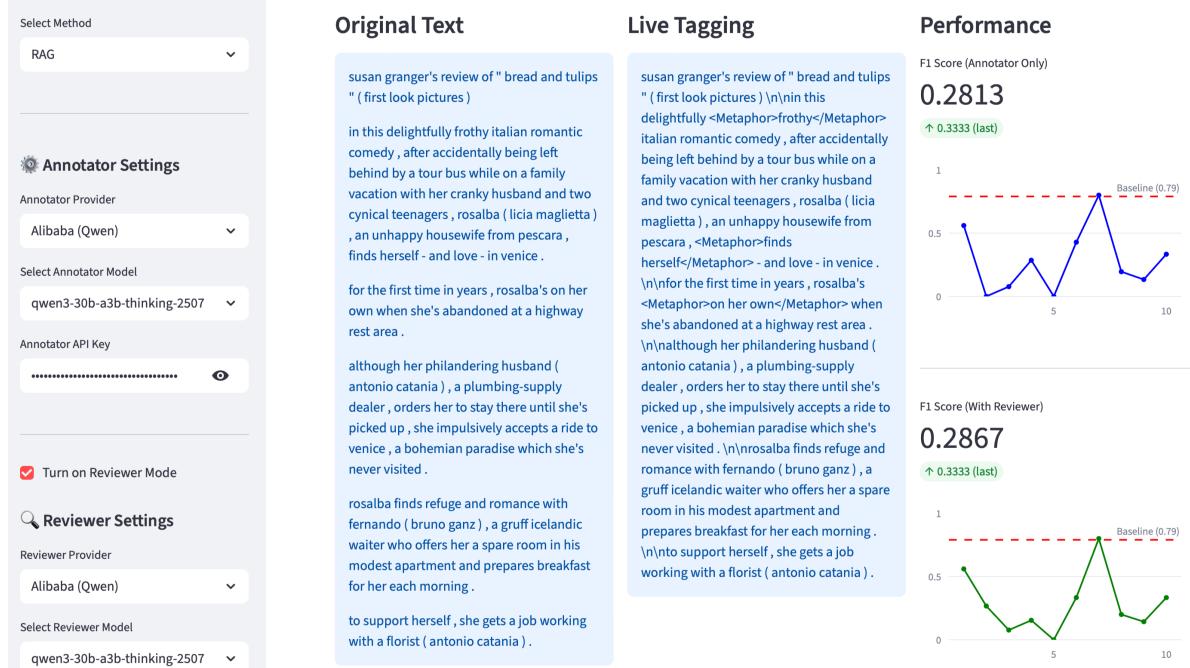


Figure 4: The live tagging display, the performance (Average F1 Score), and the progress bar.

2.4 EVALUATION

Live performance and a progress bar are displayed during the experiment, as shown in Fig. 4. LinguistAgent performs automated **Token-level Evaluation**. It tokenizes both the human “Gold Standard” and the “LLM Prediction” into binary sequences where 1 represents a tagged unit and 0 represents a non-tagged unit.

The following metrics are calculated for each sample:

- **Precision (P):** The ratio of correctly predicted metaphorical tokens to the total predicted metaphorical tokens.

$$P = \frac{TP}{TP + FP} \quad (1)$$

- **Recall (R):** The ratio of correctly predicted metaphorical tokens to all metaphorical tokens in the gold standard.

$$R = \frac{TP}{TP + FN} \quad (2)$$

- **F1 Score (F_1):** The harmonic mean of Precision and Recall.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (3)$$

2.5 TRACEABILITY AND ERROR ANALYSIS

A key challenge in multi-agent workflows is the potential for silent failures during inter-agent communication. LinguistAgent solves this through an integrated Monitoring and Debugging framework:

- **Real-time Reasoning Logs:** The Annotator's chain-of-thought is extracted from the JSON response and rendered in the UI, providing a window into the agent's linguistic logic before the final labels are applied.
- **Persistent Debug Console:** Unlike standard stateless interfaces, our platform implements a session-based log that records every raw interaction with the Gemini 3 and OpenAI-compatible APIs, as shown in Fig. 5.
- **Failure Diagnosis:** This console enables the identification of specific structural issues, such as the model reaching the `max_output_tokens` limit (resulting in a truncated JSON) or the API quota limit.

This level of transparency ensures verifiable outcomes of a transparent reasoning process, allowing for in-depth analysis of the LLM annotation.

The screenshot shows a web-based interface titled "Global Debug Console (Raw API Responses)". It displays two sections of raw JSON logs:

Sample 1 | Agent: Reviewer

```
{
  "status": "CORRECT",
  "critique": "The metaphor '<Metaphor>something is fishy in the state of universal</Metaphor>' is correctly identified and appropriately wrapped in <Metaphor> tags. T
  "revised_full_text": " <Metaphor>something is fishy in the state of universal</Metaphor> . \"\\n\\about ten years back , with the unexpected success of mad max and th
}
```

Sample 1 | Agent: Annotator

```
{
  "reasoning": "The phrase \"something is fishy in the state of universal\" is a metaphor. It plays on the famous line from Shakespeare's *Hamlet*: 'Something is rotte
  "initial_full_text": " \\n<Metaphor>something is fishy in the state of universal</Metaphor> . \"\\n\\about ten years back , with the unexpected success of mad max an
}
```

Clear Results

Figure 5: The debug section, keeping logs of all raw model responses and error types.

3 IMPLEMENTATION AND SYSTEM OBSERVABILITY

The current version of LinguistAgent is developed as a web-based application designed to bridge the gap between complex AI workflows and intuitive research interfaces. By providing a user-friendly, code-free environment, LinguistAgent empowers non-technical scholars to harness the potential of LLMs for their specific research needs without profound programming knowledge. This accessibility facilitates a more inclusive research ecosystem.

3.1 FRONTEND ARCHITECTURE: STREAMLIT INTEGRATION

The application frontend is built using the **Streamlit** framework, chosen for its ability to create highly reactive user interfaces.

- **Dynamic Configuration Interface:** The UI provides a configuration suite where researchers can dynamically switch between experimental methods (Prompt Engineering, RAG, and Fine-tuning). It supports independent model and API key selection for each agent role, allowing for heterogeneous workflows.

- **Reactive State Management:** We leveraged the *Session State* of Streamlit to ensure persistence of experimental data. This allows the dashboard, including the live tagging view and the performance charts, to remain interactive even after a batch process triggers a page rerun to enable data export.
- **Real-time Visualization:** LinguistAgent integrates **Plotly** for dynamic performance tracking. As each sample is processed, token-level F_1 scores are plotted in real-time against an adjustable academic baseline, providing immediate visual feedback on the model’s reliability.

3.2 BACKEND LOGIC AND MODEL HETEROGENEITY

The backend is engineered to handle various model architectures through a unified communication wrapper.

- **Multi-Provider Support:** LinguistAgent standardizes interactions with the Google GenAI SDK and OpenAI-compatible endpoints. This allows for the simultaneous use of disparate models, such as using Alibaba’s Qwen for initial annotation and Google’s Gemini for the high-reasoning Reviewer role.
- **Structured Output Control:** We enforce Native JSON Mode across all providers to ensure that reasoning chains and annotations are programmatically separated, which is vital for the automated evaluation engine.

4 CASE STUDY: METAPHOR IDENTIFICATION

Adapting from the IMDb metaphor dataset [Fuoli et al. \(2025\)](#), we sampled a small set of data and benchmarked LinguistAgent by comparing `qwen3-30b-a3b-thinking-2507` and `gemini-3-flash-preview` both under Annotator-only Mode and Reviewer Mode (with each as their own reviewer). Preliminary results (Table 1) indicate that, under Zero-Shot Prompting, Few-Shot Prompting and RAG, the Reviewer Mode consistently outperforms Annotator-only scenario by identifying metaphors that are often overlooked or mislabeled in a single pass.

LinguistAgent also generates a downloadable CSV containing the original data, LLM annotations, and per-sample evaluation metrics (before and after review), to assist in further in-depth manual investigation.

Table 1: The performance of two LLMs with different experimental paradigms on the metaphor identification task.

Model × Paradigm	Zero-shot Prompting	Few-shot Prompting	RAG
<i>Reviewer Mode: Off</i>			
<i>Reviewer Mode: On</i>			
qwen3-30b-a3b-thinking-2507	0.2803	0.3041	0.2813
gemini-3-flash-preview	0.5070	0.4707	0.4746
qwen3-30b-a3b-thinking-2507	0.2837	0.3294	0.2867
gemini-3-flash-preview	0.5753	0.5373	0.5809

5 CONCLUSION

LinguistAgent demonstrates that agentic workflows can transform LLMs into rigorous scientific instruments. By automating the “Annotate-Review-Evaluate” cycle, the platform empowers researchers in the Humanities and Social Sciences to scale their analyses while maintaining high levels of transparency and replicability. Future work will focus on integrating standard RAG and Fine-tuning procedures as well as “Human-in-the-loop” verification to further refine the agent.

REFERENCES

- Lynne Cameron. *Metaphor in educational discourse*. A&C Black, 2003.
- Lynne Cameron. What is metaphor and why does it matter. *Metaphor analysis: Research practice in applied linguistics, social sciences and the humanities*, pp. 3–25, 2010a.
- Lynne Cameron. Responding to the risk of terrorism: the contribution of metaphor. *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, 26:587–614, 2010b.

-
- Matteo Fuoli, Jeannette Littlemore, and Sarah Turner. Sunken ships and screaming banshees: Metaphor and evaluation in film reviews. *English Language & Linguistics*, 26(1):75–103, 2022.
- Matteo Fuoli, Weihang Huang, Jeannette Littlemore, Sarah Turner, and Ellen Wilding. Metaphor identification using large language models: A comparison of rag, prompt engineering, and fine-tuning. *arXiv preprint arXiv:2509.24866*, 2025.
- Mengshi Ge, Rui Mao, and Erik Cambria. A survey on computational metaphor processing techniques: From identification, interpretation, generation to application. *Artificial Intelligence Review*, 56(Suppl 2):1829–1895, 2023.
- Pragglejaz Group. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and symbol*, 22(1):1–39, 2007.
- Rebecca MM Hicke and Ross Deans Kristensen-McLachlan. Science is exploration: computational frontiers for conceptual metaphor theory. *arXiv preprint arXiv:2410.08991*, 2024.
- Marlene Johansson Falck and Lacey Okonski. Procedure for identifying metaphorical scenes (pims): The case of spatial and abstract relations. *Metaphor and Symbol*, 38(1):1–22, 2023.
- Mark Johnson and George Lakoff. *Metaphors we live by*, volume 1. University of Chicago press Chicago, 1980.
- Jiahui Liang, Aletta G Dorst, Jelena Prokic, and Stephan Raaijmakers. Using gpt-4 for conventional metaphor detection in english news texts. *Computational Linguistics in the Netherlands Journal*, 14:307–341, 2025.
- Jeannette Littlemore and Sarah Turner. Metaphors in communication about pregnancy loss. *Metaphor and the Social World*, 10(1):45–75, 2020.
- Eduardo Puraivan, Irene Renau, and Nicolás Riquelme. Metaphor identification and interpretation in corpora with chatgpt. *SN Computer Science*, 5(8):976, 2024.
- Gerard Steen. Three kinds of metaphor in discourse: A linguistic taxonomy. *Metaphor and discourse*, pp. 25–39, 2009.
- Gerard J Steen, Aletta G Dorst, Tina Krennmayr, Anna A Kaal, and J Berenike Herrmann. A method for linguistic metaphor identification. 2010.
- Google DeepMind Team. Gemini-3, 2025. URL <https://deepmind.google/models/gemini/>.
- Yuan Tian, Nan Xu, and Wenji Mao. A theory guided scaffolding instruction framework for llm-enabled metaphor reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7731–7748, 2024.
- Sarah Turner and Jeannette Littlemore. Literal or metaphorical? conventional or creative? contested metaphoricity in intense emotional experiences. *Metaphor and the Social World*, 13(1):37–58, 2023.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.