A New Paradigm for Cross-Domain Rumor Verification: Leveraging Fine-Tuned Language Models for Efficient and Robust Retrieval-Augmented Generation

Lisu Wang

Beijing National Day School

Our society encounters various challenges brought by the advent of abundant misinformation on online platforms, especially during worldwide events like COVID-19. Therefore, real-time rumor verification plays an indispensable role in diminishing the spread of misleading narratives. This paper presents a novel dual-pathway paradigm where, for Path 1, multiple language models with fine-tuning strategies are combined with RAG, and for Path 2, we focus on interpretable feature-based models. Our approach enables local deployment with a single 4090 GPU, processing 450 entries within one hour, which is an eight to twelve times speedup compared to RAG with large language models (>100B parameters), improving F1 by 1 to 5 percentage points. Compared to our baseline model (a fine-tuned Llama3-8B), we improve F1 by 6 to 8 percentage point. The fine-tuned GLM-4-9b reaches an F1 score of 0.9262, while the statistical validation provides concrete indications of the effectiveness of our features, providing insights for fine-tuning strategies and improving feature interpretability. Overall, our framework achieves a balance between performance, efficiency, and interpretability, enabling high-quality real-time rumor detection under resource-constrained environment.

Table of Contents

1. Introduction

In today's world, online platforms have become essential to our daily lives. Users are able to publicly comment and express their ideas on various issues and news [1]. However, the information posted by the users can sometimes be unreliable and misleading. These rumors, especially those on social media, can cause severe real-world consequences, ranging from public health crises to social unrest. As a result, it is imperative to evaluate whether a piece of information is a rumor in real-time, as it emerges and evolves constantly and is a critical research problem.

Existing methods of rumor detection and verification often find it challenging to generalize to new and unseen events, and they frequently do not employ the external knowledge that is available at the time the rumor starts to spread. In order to address this gap, we propose an innovative approach that integrates time-sensitive external evidence for improved rumor verification. This paper presents a new paradigm that combines fine-tuned language models with retrieval-augmented generation (RAG), achieving high performance and computational efficiency. Our research aims to show a robust and adaptable framework for rumor verification, allowing accurate detection of online misinformation, which is dynamic in its nature, and has therefore also been a challenging task, while enabling local deployment on single GPU cards.

2. Literature review

In academia, extensive studies have been conducted on rumor detection, including a variety of methodologies. Early research often relied on hand-crafted features and traditional machine learning models [2]. For instance, LSTM-based models were among the first to be applied to this task, illustrating the capability of recurrent neural networks to detect rumors on the Weibo dataset [3]. Later, subsequent research explored Convolutional Neural Networks (CNNs), which further improved detection performance [4].

Graph-based neural networks have also achieved notable results by effectively capturing the complex relationships inherent in data within social media contexts. Bi-directional Graph Convolutional Networks (BiGCN) can model both the propagation and dispersion of rumors [5]. Other advanced models include PLAN [6] and PPA-WAE [7], which leverage attention mechanisms and propagation path aggregation, respectively. Additional models such as UCD-RD [8], ARG [9], and LeRuD [10] have also been proposed to enhance rumor detection performance. Recent advances in fine-tuning strategies for RAG system have provided new possibilities for efficient rumor detection. Research has shown that fine-tuning can significantly enhance RAG robustness against retrieval defects [11] and improve the model's ability to evaluate and utilize retrieved evidence [12]. These prior developments help to provide our method with theoretical foundation, which integrates fine-tuning with RAG to achieve superior performance in cross-domain rumor verification.

3. Methodology

There are several key stages of our methodology: (1) data collection and preprocessing, (2)dual-path framework implementation featuring both fine-tuning RAG and feature-based approaches, (3) statistical feature analysis and selection, and (4) model training and evaluation using both individual machine learning models and an ensemble approach.

3.1. Data

In this study, we utilized a cross-domain experimental setup. Cross-domain refers to the setting in which the training and test sets derive from different data distributions [13]. Specifically, the Weibo dataset served as the training data, and the Weibo-COVID dataset was the test data. While the Weibo dataset represents the social media domain, the latter is classified as the public health domain. This cross-domain approach allowed us to evaluate our models' generalizability and robustness when faced with new and unseen events[10].

The training data contains 4606 posts, and the test data consists of 411 posts. Each entry consists of a source post (the original Weibo blog) and its comments.

3.2. Dual-Path Framework

We propose a dual-path research framework that addresses both efficiency and interpretability.

For Path 1, it is mainly a local optimization method, where we utilize fine-tuning with RAG. We use models such as Llama3-8B, DeepSeek-8B, and GLM4-9B, and it can be deployed on a single 4090/5090 card, enabling us to achieve optimal model performance with minimal computational resources.

Path 2 is an interpretability-focused method, where we combine LLM feature extraction with traditional machine learning. It is suitable for API environments, and the ensemble model of Path 2 achieves an F1 score of 0.9009, meaning it can have outstanding performance and high interpretability at the same time.

3.3. Web Search and Retrieval-Augmented Generation (RAG)

In order to provide our models with external context, we employed a Retrieval-Augmented Generation (RAG) approach. RAG combines two main components: information retrieval and generative modeling [14]. The model can query an external knowledge source to collect relevant documents. After integrating the extracted information into the input, the large language model is able to generate responses that are more accurate, grounded, and context-aware [15].

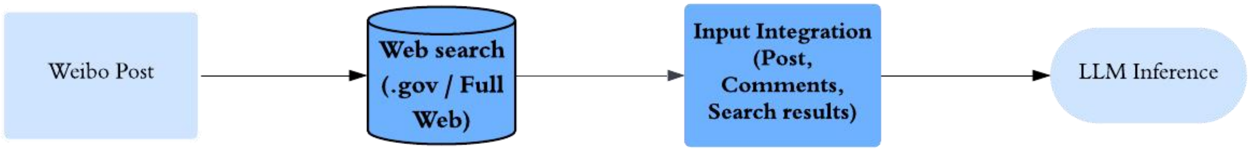
Weibo Post
Web search (.gov / Full Web)
Input Integration (Post, Comments, Search results)
LLM Inference

Fig. 1. Flow chart illustrating the Web Search RAG pipeline

We first extracted the blog content from each JSON entry, and then applied this content as the query to the search engine using the Search-Pro API. The search was initially limited to official government sources and expanded to the entire web in later experiments, and, for each entry, we retrieved the top five sources in the search results. For each, the information includes the title, link, content, reference, and publish data.

The content of these extracted search results was then used to augment the input to our language model in order to provide it with additional information to measure the veracity of the original post. By formatting the retrieved results into a prompt, we instruct the model to integrate external knowledge with the evidence provided, analyze both the content and discussion, and then determine whether the blog post is classified as a rumor. The prompt structure we provided to the language models is shown below:

```
  classification_prompt = """You are a COVID-19 rumor analyst. Based on the following
search page content, your knowledge, and the blog comments, determine step by step
whether the blog content is a rumor. If it is a rumor, finally output yes; if it
is not a rumor, finally output no.
  ## Blog content: <<content>>
  ## Blog comments: <<comments>>
  ## Search page content: <<search_result>>"""
```

Prompt 1 . web search RAG classification prompt

Then, the processed instances were saved into new JSONL files. The predictions can then be compared with the original labels to calculate standard evaluation metrics.

For Path 1, we implemented fine-tuning strategies specifically designed to enhance RAG robustness. The fine-tuning process focused on improving the model's ability to handle retrieval defects and better utilize retrieved evidence, following methodologies established in recent research [11, 12].

3.4. *Feature Engineering*

For Path 2, we maintained our comprehensive feature engineering approach. A part of our methodology is using a large language model for feature engineering. We employed the GLM-4.5 model to extract a set of 20 distinct features from each blog post. The features were carefully designed to capture a wide range of semantic and linguistic characteristics relevant to misinformation.

We constructed a feature-specific prompt. For each data instance, the blog content, comments, and retrieved search results were substituted into the template, allowing the model to analyze the text and output feature-specific scores.

Fig. 2 shows the model training process, while Fig. 3 illustrates the testing workflow:

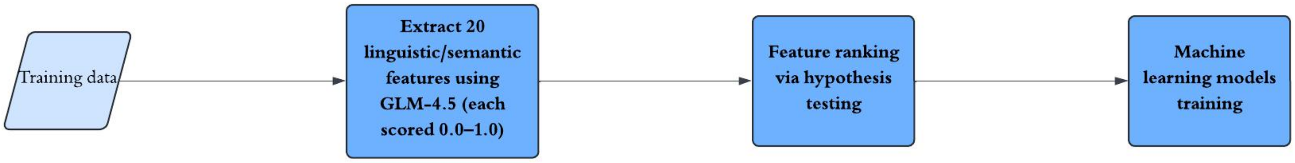
Training data
Extract 20 linguistic/semantic features using GLM-4.5 (each scored 0.0–1.0)
Feature ranking via hypothesis testing
Machine learning models training

Fig. 2. The model training process

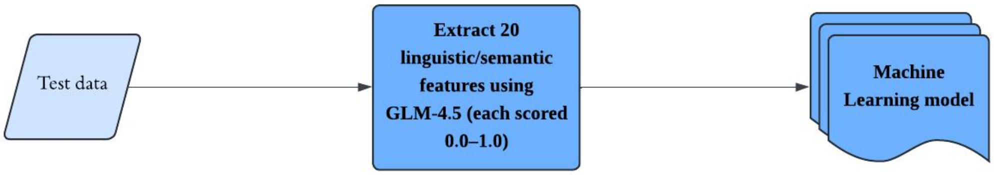
Test data
Extract 20 linguistic/semantic features using GLM-4.5 (each scored 0.0–1.0)
Machine Learning model

Fig. 3. The model testing process

3.5. *Statistical Analysis and Feature Selection*

To validate the efficacy of the extracted 20 features, we conducted a rigorous and thorough statistical analysis. We used hypothesis testing for each feature, to assess whether there were a statistically significant difference between the feature scores of rumor and non-rumor posts. Specifically, the null hypothesis assumed a equal mean between the two groups [17]; in practice, this means that the observed difference is small enough to be perceived as statistically insignificant. The alternative hypothesis stated that the means are not equal [18]. From each test, we obtained a corresponding P-value, the probability of observing a difference at least as extreme as the one measured, if the null hypothesis is true [19]. A small p-value (typically smaller than 0.05) suggests that the observed difference is near impossible to occur by chance, demonstrating a statistically significant difference between rumor and non-rumor posts. Conversely, a large p-value indicates that the observed difference could plausibly be due to random variation [20]. This setup directly tests whether each feature we proposed is useful for rumor detection.

Before applying a t-test that can directly prove the effectiveness of our features, we first need to use Levene's test for the homogeneity of variances. Since the Student's t-test requires the assumption of equal variances, and Welch's t-test is designed for cases with unequal variances, it is essential to select the two-sample t-test according to the variance characteristics of the groups[21]. In Levene's test, we set up another null hypothesis that the data have equal variances with an alternative hypothesis that the variances are unequal. The calculated results of all features suggest that the assumption of equal variances failed. The results is shown below (for reproting purposes, we denote extremely small values, when $p < 1 \times 10^{-300}$, as 0):

Feature	Levene_statistic	Levene_p_value
Coercive Language Analysis	1055.578979	1.167773×10^{-208}
Divisive Content Identification	1098.316056	3.464952×10^{-216}
Manipulative Rhetoric Analysis	1654.515357	2.683953×10^{-309}
Absolutist Language Detection	297.841184	1.018695×10^{-64}
Factual Consistency Verification	59.096268	1.823622×10^{-14}
Logical Fallacy Identification	698.313687	2.138527×10^{-143}
Attribution And Source Evaluation	25.942845	3.657308×10^{-07}
Conspiracy Theory	3259.779694	0.000000
Emotional Appeal Analysis	168.384189	7.634375×10^{-38}
Pseudoscientific Language Identification	135.144044	8.242163×10^{-31}
Call Action Assessment	715.307933	1.338356×10^{-146}
Authority Impersonation Detection	564.783760	7.399861×10^{-118}
Bot Activity Sign Detection	245.835356	5.133163×10^{-54}
User Reaction Assessment	403.176851	4.916363×10^{-86}
Dissemination Modification Tracking	907.508687	3.960184×10^{-182}
Source Credibility Assessment	55.774005	9.670968×10^{-14}

Factual Accuracy Verification	10.185302	1.425164×10^{-03}
Information Completeness Check	140.812660	5.157067×10^{-32}
External Consistency Analysis	68.978981	1.294898×10^{-16}
Expert Consensus Alignment	580.076637	8.139063×10^{-121}

Table 1 The calculated results of Levene’s test

Therefore, we leveraged Welch's t-test, which can be applied when the variances are unequal, to compare the means of the two groups. The t-statistic is calculated as:

$$t=\frac{\overline{X_1}-\overline{X_2}}{\sqrt{\frac{{s_1}^2}{n_1}+\frac{{s_2}^2}{n_2}}} \qquad \text{Eq. 1}$$

Where $\overline{X}$ is the sample mean, s^2 is the sample variance, and n is the sample size.

The P-value was used to assess statistical significance. The resulted P-values of all the 20 features is shown below:

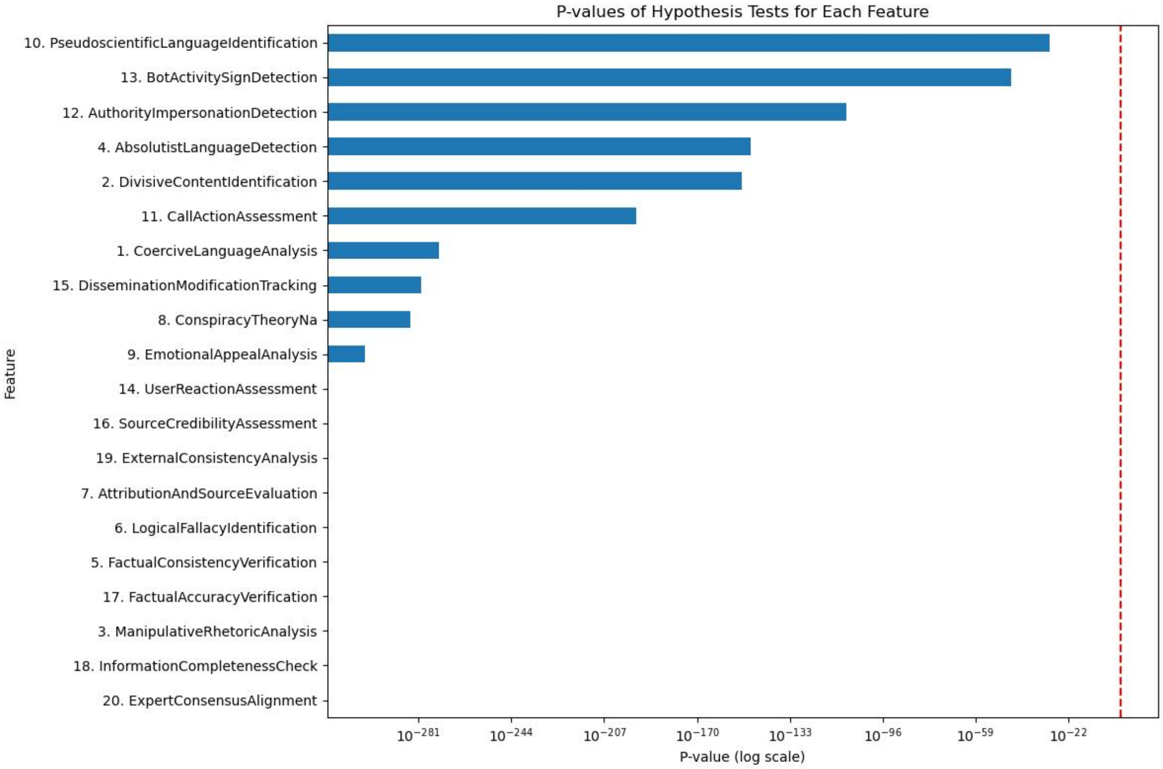

Fig. 4. P-values of Hypothesis Tests for Each Feature

As shown in Fig. 4, our analysis revealed that all 20 features demonstrate a statistically significant difference between the two classes ($p<0.05$), confirming their relevance (in the visualization, some features appear to have values very close to zero, which may give the impression of missing data, but, in fact, their bars are simply too small to be visible at the chosen scale).

We also calculated the effect size for each feature using Cohen's d, the resulted values are shown below:

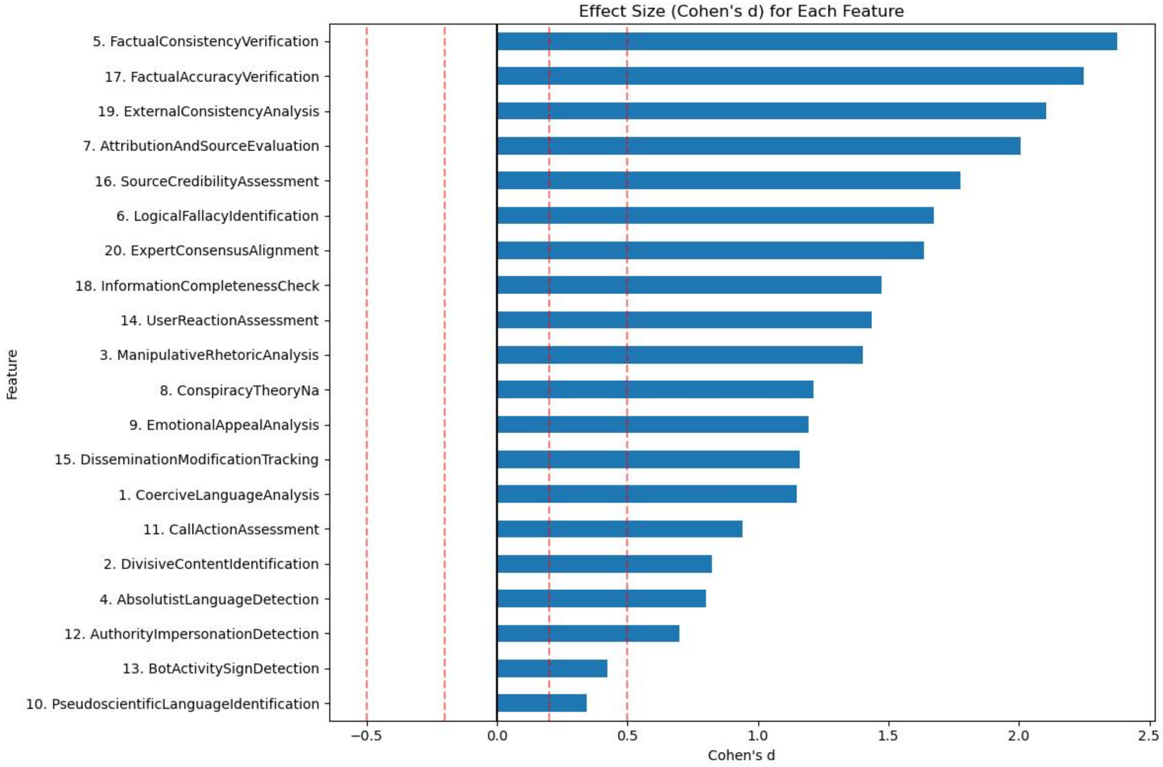

Fig. 5. Effect Size (Cohen d) for Each Feature

The results verified that the majority of the features, such as “5. FactualConsistencyVerification” (d=2.3775) and “17. FactualAccuracyVerification” (d=2.2473), have a large effect size. However, two features, “13. Bot Activity Sign Detection” and “10. Pseudoscientific Language Detection”, have effect sizes below 0.5. According to conventional benchmarks (0.2 is considered small, 0.5 medium, and 0.8 large[22]), these values are classified into the medium range. This finding does not necessarily indicate that these features are ineffective. On the contrary, both features were still statistically significant, according to the previous hypothesis test. The medium effect size suggests that the magnitude of the separation is smaller than that of other features, but still moderately meaningful.

This statistical analysis supports the selection of the 20 features for subsequent model evaluation and ensure the reliability of our methodology.

3.6. Computational Efficiency Optimization

A key innovation of our research is the optimization for local deployment, with our model achieving high computational efficiency.

With a 4090 GPU processing 450 entries within one hour, the method using fine-tuning with RAG demonstrates exceptional efficiency. Specifically, achieving eight times the speed of GLM-4-Plus + RAG (running more than 8 hours), and 12 times that of the DeepSeek-R1 RAG (running more than 12 hours).

This efficiency enables real-time rumor detection in resource-constrained environments.

3.7. Machine Learning Models and Ensemble

After rigorous statistical analysis, we trained and evaluated several traditional machine learning models for Path 2, implemented with the scikit-learn (sklearn) library, using the extracted feature scores mentioned before, which serve as numerical representations of linguistic and social signals identified by the LLM (GLM-4.5). The models are Logistic Regression, which captures linear decision boundaries [23]; a Decision Tree, which classifies by hierarchical, non-linear decision rules [24]; and a Support Vector Machine (SVM), which finds an optimal hyperplane that maximizes the margin between rumor and non-rumor samples [25].

For Path 1, we focused on improving fine-tuned RAG models, while for Path 2, we developed an ensemble model that combines the predictions of our best-performing machine learning models with the direct output of a independent separate LLM; Specifically, we applied the voting-based ensemble approach, where the ultimate label for each instance is decided by the predictions from three separate models. During the inference, each model, trained independently, produces a binary label for each test sample. Then, we saved these predictions into separate columns. After that, we compute a majority-vote ensemble. That is, we counted how many base classifiers predicted the sample as “rumor” (1), and if two or more classifiers (out of three) identified the sample as 1, the ensemble prediction was set to 1. Otherwise, the ensemble model will output “non-rumor” (0). Because the number of base models employed here was odd, there was no ambiguity from ties.

For Path 2, we also used a simple weighted evaluation approach with a threshold of 0.2 (the score given by the LLM during feature extraction). This method combines the 20 LLM-extracted features through a weighted sum, where features with scores above 0.2 are considered useful when identifying the rumor. While this approach achieved a high recall of 0.9053, its precision (0.7785) and negative F1 (0.6295) were lower than the more sophisticated machine learning models, leading to an overall F1 score of 0.8371.

The voting based ensemble model used in Path 2 is illustrated in Fig. 6.

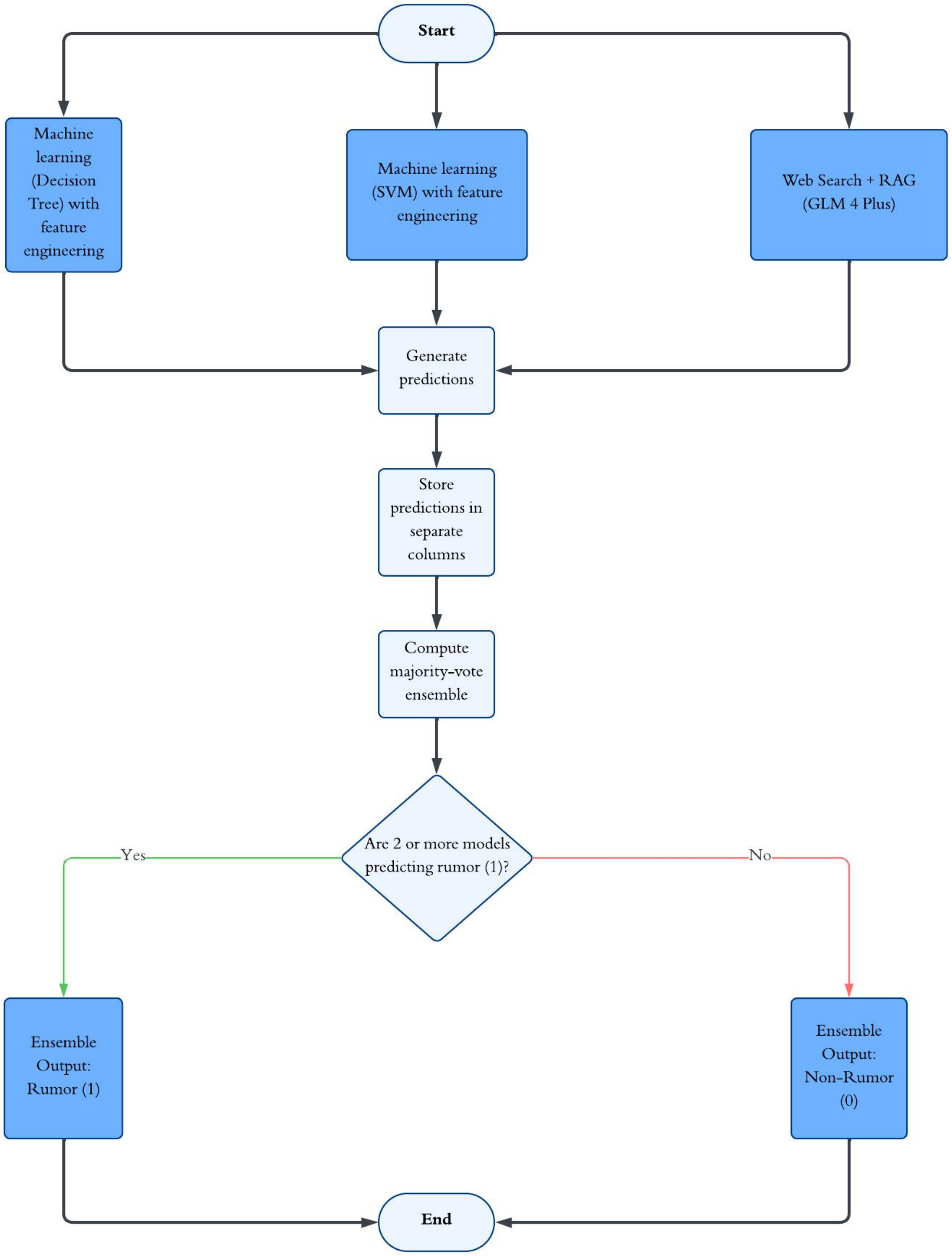
Start
Machine learning (Decision Tree) with feature engineering
Machine learning (SVM) with feature engineering
Web Search + RAG (GLM 4 Plus)
Generate predictions
Store predictions in separate columns
Compute majority-vote ensemble
Are 2 or more models predicting rumor (1)?
Yes
No
Ensemble Output: Rumor (1)
Ensemble Output: Non-Rumor (0)
End

Fig. 6 Overview of the proposed ensemble model

4. Experiments and Results

In our study, we conducted a series of experiments to validate our proposed methodology. The baseline model is a LoRA fine-tuned Llama3-8B, trained using the Weibo dataset and evaluated with the separate Weibo-COVID dataset.

Model/Method	Accuracy	Precision	Recall	Negative F1	F1-score
Precious Studies					
LSTM [2]	0.4185	–	–	0.4156	0.4213
CNN [3]	0.4161	–	–	0.4059	0.4258
BiGCN [4]	0.6058	–	–	0.4938	0.6773
UCD-RD [12]	0.6788	–	–	0.4677	0.7700
Our study					
Fine-Tuning Llama3-8B (Baseline)	0.7737	0.8834	0.7462	0.7224	0.8090
Path 1: Fine-Tuned RAG Models					
Web Search RAG (Llama3-8B)	0.3771	0.7222	0.0492	0.5259	0.0922
Web Search RAG (Fine-Tuning Llama3-8B)	0.8297	0.8849	0.8447	0.7712	0.8643
Fine-Tuning DeepSeek-8B	0.8029	0.9256	0.7538	0.7638	0.8309
Web Search RAG (DeepSeek-R1)	0.8188	0.8038	0.9513	0.6939	0.8714
Web Search RAG (DeepSeek-8B)	0.3917	0.6250	0.1326	0.5020	0.2188
Web Search RAG (Fine-Tuning DeepSeek-8B)	0.8589	0.8732	0.9129	0.7943	0.8926
Fine-Tuning GLM-4-9B	0.8345	0.9900	0.7500	0.8101	0.8534
Web Search RAG (GLM-4-Plus)	0.8418	0.8419	0.9280	0.7566	0.8829
Web Search RAG (GLM-4-9B)	0.3698	0.8571	0.0227	0.5299	0.0443
Web Search RAG (Fine-Tuning GLM-4-9B)	0.9027	0.9029	0.9508	0.8571	0.9262
Path 2: Feature-Based Models					
Web Search RAG (LLM feature extraction + Simple Weighted Evaluation (threshold=0.2))	0.7737	0.7785	0.9053	0.6295	0.8371
Web Search RAG (LLM Feature Extraction + Logistic Regression)	0.8200	0.8545	0.8674	0.7448	0.8609
Web Search RAG (LLM Feature extraction + Decision Tree)	0.8248	0.8721	0.8523	0.7600	0.8621
Web Search RAG (LLM Feature extraction + SVM)	0.8345	0.8769	0.8636	0.7718	0.8702
Ensemble Model (Decision Tree + SVM + Web Search RAG (GLM-4-Plus + all))	0.8710	0.8893	0.9129	0.8153	0.9009

Table 2 Performance Comparison of Our Models and Previous Work under the Cross-Domain Setting (Weibo → Weibo-COVID)

As shown in Table 2, the fine-tuned GLM-4-9B model with full web-ranged RAG achieves the best performance with an F1 score of 0.9262, which significantly outperforms all other previous methods. The Fine-tuned DeepSeek-R1 with full web-ranged RAG shows strong performance (F1 = 0.8926), while maintaining high computational efficiency. The ensemble model has robust performance (F1 = 0.9009), while maintaining its interpretability. Additionally, all the fine-tuned RAG small models (under 14B parameters) substantially outperform larger models (over 100B parameters) that only use RAG without fine-tuning.

5. Comparative Analysis and Discussion

The noticeable and superior performance of our method can be attributed to several key innovations and advantages listed below.

5.1. Fine-Tuning Enhanced RAG Robustness

The outstanding performance of our fine-tuned RAG models can be attributed to the models' robustness brought by the specialized fine-tuning strategies. The fine-tuning can effectively address two challenges in the RAG system: first, the instability of the retrieval quality can be addressed by robust fine-tuning strategies [11]; second, the context window contamination by "hard negatives" is mitigated by utilizing the evidence more thoroughly [26]. This enhancement can be seen from the difference in performance between the fine-tuned RAG and the non-fine-tuned models.

5.2. Computational Efficiency and Local Deployment

Our approach achieves strong performance while allowing local deployment. Specifically, a single 4090 GPU only needs one hour to process 450 entries. The fine-tuning with RAG strategies improves efficiency substantially, with an eight-times speedup compared to the GLM-4-Plus model with RAG, and a twelve-times speedup compared to DeepSeek-R1 RAG. This is an important improvement, since it balances model performance and computational efficiency, while previous methods often require extensive computational resources or cloud-based deployment. Thus, we make real-time rumor detection feasible under resource-constrained environments.

5.3. Path 2 Feature Validation for Path 1 Optimization

Our statistical validation provides valuable insights to improve Path 1. With all 20 features demonstrating statistical significance and 18 features showing large effect sizes, this validation can guide the fine-tuning of the models in Path 1, ensuring that the models focus on the most effective and discriminative aspects of the evidence and rumor identification. However, the simple weighted model, while achieving an outstanding recall score of 0.9053, shows the limitations of basic feature combination methods. Its precision of 0.7785 and negative F1 of 0.6295 is lower than the SVM model's 0.8769 precision and 0.7718 negative F1, potentially pointing out the advantages of using machine learning to obtain better utilization of features.

5.4. Computational Trade-offs

Our dual-path approach addresses the challenge of balancing model performance with interpretability. For Path 1, where we use fine-tuned RAG models, the model can achieve outstanding performance (F1 = 0.9262) through local deployment. For Path 2, it also achieves high performance (F1 = 0.9009) while maintaining full interpretability. This flexibility allows selecting the best-performing approach under different requirements and resource constraints.

6. Conclusions

We presented an innovative paradigm for cross-domain rumor detection by using a fine-tuned large language model with retrieval-augmented generation, which eventually achieves outstanding performance with high computational efficiency. Our fine-tuned GLM-4-9B with RAG model achieves an F1-score of 0.9262, showing significant advances compared to previous methods while simultaneously maintaining the advantage of enabling local deployment on a single GPU card.

This work presents a new performance benchmark for cross-domain rumor detection and provides an outstanding and practical deployable solution for real-world rumor detection tasks.

7. Future direction

We identify that there are several key aspects that future studies can delve into:

1. RAG Robustness Optimization: Further research is needed on solving the problems brought by “hard

negatives" [26] to further improve the reliability of the method.

2. Advanced Retrieval Architecture Integration: Exploration of Multi-head RAG [27] that can deal with complex and multi-aspect queries.

3. Cross-Domain Generalization: Further extension of our approach by generalizing it to additional domains and languages to validate its universal applicability.

4. Performance and Interpretability Fusion: Development of methods that combine the efficiency of fine-tuned RAG with the interpretability of feature-based approaches more thoroughly.

REFERENCES

[1] T. Bian, “Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks”, AAAI, vol. 34, no. 01, pp. 549-556, Apr. 2020.

[2] M. Al-Sarem, W. Boulila, M. Al-Harby, J. Qadir, and A. Alsaeedi, "Deep learning-based rumor detection on microblogging platforms: A systematic review," *IEEE Access*, vol. 7, pp. 152788–152812, 2019, doi: 10.1109/ACCESS.2019.2947855.

[3] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, “Detecting rumors from microblogs with recurrent neural networks,” in Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI), New York, NY, USA, July 2016, pp. 3818–3824.

[4] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, “CAMI: A convolutional approach for misinformation identification,” in Proc. 26th Int. Joint Conf. Artificial Intelligence (IJCAI), Melbourne, VIC, Australia, 2017, pp. 3905–3911.

[5] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, “Rumor detection on social media with bi-directional graph convolutional networks,” in Proc. 34th AAAI Conf. Artificial Intelligence (AAAI), New York, NY, USA, 2020, pp. 549–556.

[6] L. M. S. Khoo, H. L. Chieu, Z. Qian, and J. Jiang, “Interpretable rumor detection in microblogs by attending to user interactions (PLAN),” in Proc. 34th AAAI Conf. Artificial Intelligence (AAAI), New York, NY, USA, 2020, pp. 6367–6374.

[7] P. Zhang, H. Ran, C. Jia, X. Li, and X. Han, “A lightweight propagation path aggregating network with neural topic model for rumor detection (PPA-WAE),” Neurocomputing, vol. 458, pp. 468–477, 2021.

[8] H. Zhang, J. Dong, and C. Yu, “UCD-RD: Uncertainty-aware contrastive learning for debunking rumors,” in Proc. 59th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 2021, pp. 234–245.

[9] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: A data mining perspective,” SIGKDD Explor. Newsl., vol. 19, no. 1, pp. 22–36, 2017.

[10] Y. Yuan, J. Ma, and H. Cao, “LeRuD: Leveraging representation learning for rumor detection on social media,” in Proc. 29th Int. Joint Conf. Artificial Intelligence (IJCAI), Yokohama, Japan, 2020, pp. 4751–4757.

[11] Y. Tu, W. Su, Y. Zhou, Y. Liu, and Q. Ai, “RbFT: Robust Fine-tuning for Retrieval-Augmented Generation against Retrieval Defects,” *arXiv preprint* arXiv:2501.18365, 2025. doi: 10.48550/arXiv.2501.18365.

[12] D. Russo, S. Menini, J. Staiano, and M. Guerini, “Face the Facts! Evaluating RAG-based fact-checking pipelines in realistic settings,” *arXiv preprint* arXiv:2412.15189, 2024. doi: 10.48550/arXiv.2412.15189.

[13] H. Ran and C. Jia, “Unsupervised Cross-Domain Rumor Detection with Contrastive Learning and Cross-Attention”, AAAI, vol. 37, no. 11, pp. 13510-13518, Jun. 2023.////

[14] J. Liu, J. Lin, and Y. Liu, “How much can RAG help the reasoning of LLM?,” arXiv:2410.02338 [cs.CL], 2024. doi: 10.48550/arXiv.2410.02338.

[15] C. Hsieh, C. Moreira, I. B. Nobre, S. C. Sousa, C. Ouyang, M. Brereton, J. Jorge, and J. C. Nascimento, "DALL-M: Context-aware clinical data augmentation with large language models," *Computers in Biology and Medicine*, vol. 190, p. 110022, 2025, doi: 10.1016/j.compbiomed.2025.110022.

[16] G. Kennedy, “Asia Pacific,” Computer Law & Security Rev., vol. 35, no. 3, pp. 361–368, 2019.

[17] D. S. Collingridge, “A primer on quantitized data analysis and permutation testing,” J. Mixed Methods Res., vol. 7, no. 1, pp. 81–97, 2012.

[18] L. B. Christensen, R. Burke Johnson, and L. A. Turner, "Hypothesis Testing," in *Research Methods, Design, and Analysis*, Pearson, 2013.

[19] J. C. Travers, B. G. Cook, and L. Cook, “Null hypothesis significance testing and p values,” Learn. Disabil. Res. Pract., vol. 32, no. 4, pp. 208–215, 2017, doi: 10.1111/ldrp.12147.

[20] D. Curran-Everett, "Explorations in statistics: hypothesis tests and P values," Advances in Physiology Education, vol. 33, no. 2, pp. 81-86, 2009.

[21] S. Schlotzhauer, "Chapter 9 Comparing More Than Two Groups," in *Elementary Statistics Using SAS*, SAS Institute, 2009.

[22] Kallogjeri D, Piccirillo JF. A Simple Guide to Effect Size Measures. JAMA Otolaryngol Head Neck Surg. 2023;149(5):447–451. doi:10.1001/jamaoto.2023.0159

[23] M. P. LaValley, “Logistic regression,” Circulation, vol. 117, no. 18, pp. 2395–2399, 2008.

[24] Y.-Y. Song and Y. Lu, “Decision tree methods: applications for classification and prediction,” Shanghai Arch. Psychiatry, vol. 27, no. 2, pp. 130–135, 2015.

[25] V. Jakkula, “Tutorial on support vector machine (SVM),” School of EECS, Washington State Univ., Pullman, Tech. Rep. 37.2.5, 2006.

[26] B. Jin, J. Yoon, J. Han, and S. O. Arik, “Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG,” *arXiv preprint* arXiv:2410.05983, 2024. doi: 10.48550/arXiv.2410.05983.

[27] Y. Zhu, W. Zhao, A. Li, Y. Tang, J. Zhou, and J. Lu, “FlowIE: Efficient Image Enhancement via Rectified Flow,” *arXiv preprint* arXiv:2406.00508, 2024. doi: 10.48550/arXiv.2406.00508.