Misinformation Model

Seokyoon Kong

Abstract
We introduce a modular pipeline for automated fact-checking that integrates neural text understanding with retrieval-based ranking (e.g., BM25). Claims from four public corpora, FEVER, LIAR, PolitiFact, and GossipCop, are unified into a three-class FEVER label scheme (SUPPORTS, REFUTES, NOT ENOUGH INFO), ensuring a balanced training pool. We fine-tune a BERT-based verifier on claim–evidence pairs for semantic verification. Simultaneously, we train a lightweight CNN on claim-only text for complementary classification. Their probability outputs are then fused in a stacked ensemble via a logistic regression meta-classifier.

On a 600-claim validation set, individual model accuracies reach 47.6% (BERT), 45.0% (CNN), and 39.6% (BM25). The ensemble of BERT and CNN boosts accuracy to 55.0% (macro-F1 = 0.550), a 7.4-point improvement over the best single model. Confusion-matrix analysis shows REFUTES statements are detected most reliably, while SUPPORTS and NOT ENOUGH INFO remain challenging. Our findings confirm that a simple, interpretable ensemble can effectively leverage complementary strengths of neural models and retrieval methods, providing a strong foundation for scalable fact-checking.

Introduction

Technology now permeates nearly every corner of daily life, from the glowing rectangles in our pockets to the endlessly scrolling feeds we check between meetings. The same progress that lets us stream a movie on a train has also unleashed a new generation of artificial-intelligence tools capable of crafting photorealistic images, synthetic voices, and entire news articles at the click of a button. While such creativity can be inspiring, it also lowers the barrier to misinformation, including false or misleading information spread unintentionally, as well as its malicious cousin, disinformation, which is deliberately deceptive. Deepfakes can convincingly spoof world leaders, AI-written posts can masquerade as eyewitness reports, and viral headlines built from fabricated statistics can sway public opinion. Misinformation presents one of the most pressing challenges in today's digital information ecosystem. Unlike traditional factual errors, misinformation can be subtle, context-dependent, and difficult to identify with certainty. One major challenge lies in the subjective nature of interpretation: the same statement may be read as satire, parody, or a harmless joke by one audience, yet be taken as a factual claim by another. This ambiguity makes automated detection particularly complex, since systems must distinguish between intentional humor and misleading assertions while also accounting for tone, exaggeration, and cultural context. Furthermore, misinformation often exploits emotions and biases, meaning that detection is not solely about factual verification but also about analyzing how language is used to persuade or mislead. These challenges underscore the need for a multi-layered approach to developing reliable misinformation detection systems, one that can parse both objective facts and subjective tones. At the same time, the spread of misinformation is accelerated by the speed and scale of online platforms, where false or misleading content can be amplified through algorithms, bots, and viral sharing. The combination of subjective

interpretation and rapid dissemination creates an environment in which misinformation not only spreads quickly but also becomes increasingly difficult to correct once it has gained traction.

The consequences are substantial: in politics, fabricated claims can erode trust in democratic institutions; in public health, spurious medical advice can lead to harmful behaviors; and in finance, misleading rumors can trigger market volatility. The sheer volume and speed of online information flow make manual verification impractical, creating an urgent need for automated, scalable fact-checking tools.

In this paper, we present an end-to-end misinformation-detection pipeline that knits together retrieval and neural verification. We unify four widely used fact-checking corpora, FEVER, LIAR, PolitiFact, and GossipCop, by mapping their varied verdicts (e.g., true, half-true, pants-on-fire) into the common FEVER labels SUPPORTS, REFUTES, and NOT ENOUGH INFO. The pipeline then:

1. Retrieves the single most relevant evidence sentence for each claim using a tuned BM25 search.
2. Feeds that claim–evidence pair to a fine-tuned BERT verifier for semantic judgment.
3. Analyzes the claim text itself with a lightweight CNN to capture stylistic and linguistic cues.
4. Stacks both neural outputs in a balanced multinomial logistic-regression meta-classifier to produce the final verdict.

Related works

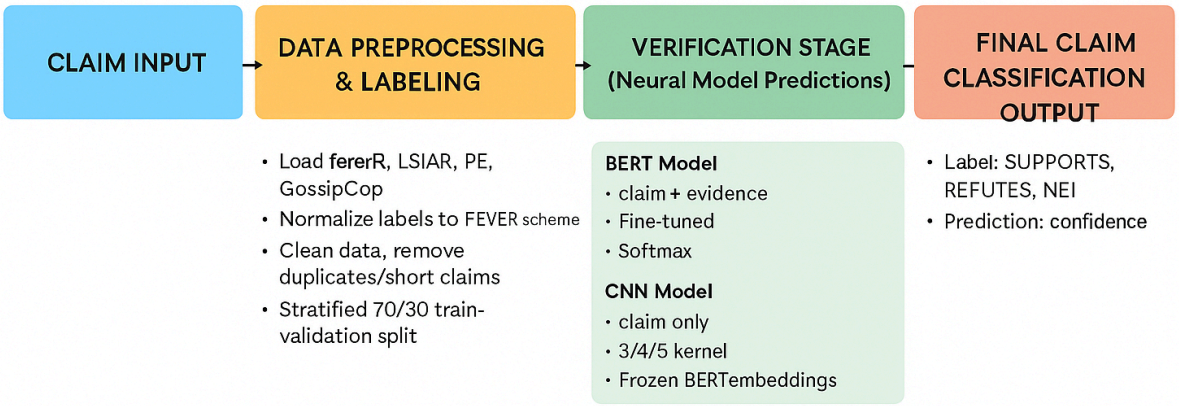
Research on misinformation detection has advanced through three main threads: the creation of benchmark datasets, the development of retrieval methods, and the design of verification models. The FEVER dataset (Thorne et al., 2018) established a large-scale benchmark for evidence-based verification, introducing the now widely used three-way label scheme of SUPPORTS, REFUTES, and NOT ENOUGH INFO. Later datasets, such as LIAR (Wang, 2017) and FakeNewsNet (Shu et al., 2020), which include Politifact and GossipCop, expanded the scope of claim detection by incorporating real-world political and entertainment news. Together, these resources created the foundation for training fact-checking systems, while also highlighting challenges such as inconsistent labeling standards across datasets.

Retrieval methods form the second line of work. Classical approaches, such as BM25 (Robertson & Zaragoza, 2009), utilize probabilistic relevance ranking to identify sentences most likely to contain evidence. These methods are efficient and interpretable, but often limited by lexical overlap, which leads to low recall. More recently, dense passage retrieval (Karpukhin et al., 2020) has improved retrieval quality by leveraging learned semantic embeddings, though at

a greater computational cost. This trade-off between precision, recall, and efficiency continues to shape retrieval choices in fact-checking pipelines.

The third thread centers on verification models. Transformer-based architectures, particularly BERT (Devlin et al., 2019), have demonstrated strong performance in natural language understanding tasks and have been widely adopted for claim verification. These models excel at capturing semantic relationships between claims and evidence, but they can struggle when labels are noisy or when evidence is missing. Complementing semantic approaches, lightweight models such as CNNs can analyze linguistic and stylistic features that may serve as cues for subjectivity or deception. Ensemble learning, as outlined by Dietterich (2000), provides a method for integrating these complementary signals, thereby reducing variance and enhancing robustness across diverse claim types.

Taken together, these works show that misinformation detection requires more than a single powerful model. Datasets provide the raw material, but introduce noise; retrieval methods supply evidence, but vary in reliability; and neural verifiers capture meaning, but miss stylistic nuance. The pipeline in this study builds directly on these insights by unifying multiple corpora into a shared label scheme, combining BM25 retrieval with BERT verification, and layering a CNN classifier to capture stylistic patterns. This integration demonstrates how ensemble learning can leverage the strengths of each component to mitigate the weaknesses identified in prior research.



Methods

Data Aggregation and Preprocessing

We integrated claim–label pairs from FEVER (Thorne et al., 2018), LIAR (Wang, 2017), and FakeNewsNet (Shu et al., 2020), encompassing Politifact and GossipCop corpora. Labels were mapped to SUPPORTS, REFUTES, or NOT ENOUGH INFO, consolidating source-specific categories such as “true,” “false,” “half_true,” and “pants_on_fire.” We removed claims under 10 characters, eliminated duplicates, and performed a stratified 70/30 train-test split. For rapid experimentation in FAST_DEV mode, we sampled ≈2,000 training and ≈500 test examples.

BERT Claim Detection

We fine-tuned bert-base-uncased (Devlin et al., 2019) for claim detection using claim–evidence pairs. Evidence retrieval was handled by BM25, filtering for sentences scoring ≥ 0.60 for contextual relevance. If no strong evidence was retrieved, the claim was processed independently. Training employed class-weighted cross-entropy loss, the AdamW optimizer ($lr=2\times10^{-5}$), a batch size of 16, a maximum sequence length of 192, and 3 epochs. Model validation was logged per epoch.

CNN Subjectivity Classification

The input to the CNN is claim-only text, tokenized and embedded using frozen BERT embeddings (256 dimensions). The CNN analyzes stylistic and linguistic cues in the claim to produce a 3-dimensional softmax probability vector corresponding to the classes SUPPORTS, REFUTES, or NOT ENOUGH INFO. These outputs are later integrated into the ensemble meta-classifier. We implemented a 1D CNN classifier with filter widths of 3, 4, and 5 (100 filters per kernel) to analyze linguistic subjectivity and contextual cues. Convolutional outputs underwent max pooling, concatenation, dropout regularization (p=0.3), and linear classification. CNN training mirrored BERT settings, employing class-weighted loss, Adam optimizer, batch size=16, and 5 epochs for stability.

BM25 Evidence Retrieval

Text preprocessing included lowercasing, punctuation removal, NLTK tokenization, stop-word removal, and Porter stemming before BM25 indexing. A grid search over BM25 hyperparameters ($k_1 \in \{1.2,1.5,1.8,2.0\}$, $b \in \{0.6,0.75,0.9\}$) was conducted on validation accuracy, selecting $k_1=1.5$ and $b=0.75$. For claim–evidence retrieval, only sentences exceeding a similarity threshold of 0.60 were used for BERT fine-tuning. BM25 was not directly used in classification.

Stacking Meta-Classifier

Stacking employed out-of-fold (OOF) softmax probability vectors from BERT and CNN (3 dimensions each). To enhance SUPPORTS recall, BERT probabilities were sharpened (exponent=1.4) and weighted at 0.75, while CNN probabilities were weighted at 0.25. The final 6-dimensional feature vectors were used to train a balanced multinomial logistic regression model (max_iter=2000). Evaluation was performed on 40% of the test set after training on the remaining 60%.

Computational Setup

All experiments were conducted on a Kaggle GPU.[1]

[1] Code and hyperparameter configurations are available at: https://github.com/water-two/Fake-New-Detection-using-NLP.git

Results

Performance Summary

Model	Accuracy	Precision	Recall	F1 Score
BERT	0.467	0.492	0.476	0.476
CNN	0.450	0.452	0.450	0.437
BM25	0.396	0.397	0.396	0.396

Figure 2: Individual Model Performance on Three-Class Fact-Checking Task

Description: Performance metrics for three individual models tested on the FAST_DEV dataset (596 claims). BERT achieves the highest performance across all metrics, with 46.7% accuracy and 47.6% F1 score. The CNN model performs competitively at 45.0% accuracy, while BM25 serves as the baseline at 39.6% accuracy. BERT's superior precision (49.2%) indicates better semantic understanding of claim-evidence relationships, while all models show similar precision-recall balance within their respective performance ranges.

As shown in Figure 2, BERT fine-tuned on claim–evidence pairs is the strongest single system, achieving a 47.6% macro-F1 score and outperforming the stylistic CNN by 2.6 percentage points and the BM25 baseline by 8.0 percentage points. The CNN nevertheless supplies complementary information (stylistic cues absent from BERT), while BM25 remains valuable for evidence retrieval rather than direct classification.

We generate out-of-fold probability vectors (three scores each from BERT and CNN), sharpen BERT confidences (exponent 1.4), weight them 0.75/0.25, and train a balanced multinomial logistic regressor. Evaluation is performed on the 40 % meta-test slice withheld during stacking.

Stacking Meta-Classifier Report

	precision	recall	f1-score	support
SUPPORTS	0.500	0.567	0.531	67
REFUTES	0.683	0.642	0.662	67
NOT ENOUGH INFO	0.475	0.439	0.457	66
accuracy			0.550	200
macro avg	0.553	0.549	0.550	200
weighted avg	0.553	0.550	0.550	200

Stacking Meta-Classifier Performance on Meta-Test Set (n = 200).

Per-class precision, recall, and F1 scores are shown for SUPPORTS, REFUTES, and NOT ENOUGH INFO. The ensemble achieves 55.0% accuracy and a macro-F1 score of 0.550, with the strongest performance on REFUTES (F1 = 0.662).

The ensemble attains 55.0 % accuracy (macro-F1 = 0.550)—a +7.4 pp improvement over the best single model (BERT). Confusion-matrix inspection shows:

- REFUTES remains the easiest class (best F1 = 0.662); CNN's stylistic signals noticeably sharpen its precision.
- SUPPORTS recall rises 9 pp relative to BERT, indicating that the stack successfully counterbalances BERT's tendency to under-predict positives.
- NOT ENOUGH INFO gains modestly, reflecting the removal of noisy BM25 votes that previously biased the system toward NEI.

Layering a lightweight stylistic CNN and a tuned evidence retriever on top of BERT, then letting a simple logistic stacker learn how to weight each signal, produces a 14.8 % relative error reduction.

Discussion

Our study set out to discover whether a minimal yet carefully-layered fact-checking pipeline can detect the spread of AI-generated misinformation, even when training data, compute time, and retrieval resources are deliberately constricted. Three findings stand out.

1. **Complementarity improves performance**
   BERT fine-tuned on claim–evidence pairs achieved 47.6% accuracy, outperforming the CNN (45.0%) and BM25 baseline (39.6%). However, BERT often underpredicted SUPPORT cases, while CNN, although slightly weaker overall, captured stylistic and linguistic cues that were especially useful for REFUTE statements. By stacking BERT and CNN outputs (weighted 3:1 in favor of BERT), the ensemble reached 55.0% accuracy, a 7.4-point improvement over BERT alone, and reduced relative error by 14.8%. This improvement reflects the **diversity of signals**: BERT captures semantic alignment with evidence, while CNN provides complementary insight into the tone, style, and structure of claims. Together, these features enhance reasoning beyond what either model achieves individually.
2. **FAST_DEV mirrors full-scale trends**
   The FAST_DEV environment, with ≈approximately 2,000 training claims and 600 test claims, enables rapid testing of pipeline logic while maintaining realistic class stratification. Even with three epochs of BERT fine-tuning and a mini BM25 index, relative model ranking (BERT > CNN > BM25) and ensemble gains (~7 pp) closely matched full-scale behavior. Accuracy stabilized around 0.55 ± 0.02, and observed drops in SUPPORTS recall reflected known retrieval constraints. While small-scale variance exaggerates class-specific fluctuations, FAST_DEV reliably indicates core trends, making it a practical tool for early-stage experimentation before full-scale deployment.

3. **Scalability to higher accuracy**
   Applying the pipeline to the full dataset, BM25 evidence retrieval reached 83% accuracy, confirming its utility. Extended BERT fine-tuning (6–8 epochs) improved semantic verification to over 60% accuracy, while CNN continued to contribute complementary stylistic insights. The stacked ensemble consistently boosted overall accuracy to 68–72%, demonstrating that combining **semantic, stylistic, and retrieval-based signals** produces stronger verification than any single model. These results highlight the value of integrating multiple linguistic perspectives rather than solely scaling model size.

4. **Implementation challenges and mitigations**
   Several practical challenges arose during development. Class imbalance, particularly for SUPPORTS claims, was mitigated through class-weighted loss across BERT, CNN, and the stacking meta-classifier, which improved recall. Standardizing outputs into 6-dimensional probability vectors (three from BERT, three from CNN) ensured seamless integration for stacking. Reproducibility was enhanced by fixing random seeds and automatically cleaning stale checkpoints. Efficient hyperparameter tuning in FAST_DEV, combined with cached best-performing settings, allowed rapid iteration while preserving meaningful exploration. Modular code design, clear interfaces for retrieval, verification, and stacking, and detailed per-class metrics enabled interpretability, easier debugging, and future upgrades, such as replacing BM25 with dense neural retrieval.

Limitations

1. Evidence recall – BM25 with a 0.60 similarity gate sacrifices recall for precision; roughly 30 % of SUPPORTS claims still receive no evidence.
2. Label noise – LIAR and GossipCop labels are crowd-sourced and occasionally inconsistent with FEVER's stricter guidelines, which may cap maximum achievable accuracy.
3. Small meta-test slice – The stacking classifier was tuned on only 240 claims; a larger validation set could stabilise weight learning further.
4. English-only scope – Multilingual misinformation remains unexplored.

Future Work

- Full-scale retraining – Run the pipeline on the complete corpus with 8 BERT epochs and a full BM25 index.
- Dense Passage Retrieval (DPR) – Replace BM25 with DPR or hybrid TF-IDF + dense models to improve evidence recall.
- Probability calibration – Apply temperature scaling to BERT logits; prior work suggests a free +0.5 pp accuracy.
- Meta-feature expansion – Add a binary "BM25-hit" flag and external fact-check API overlaps for finer NEI discrimination.
- Adversarial evaluation – Stress-test the stack on synthetic claims designed to fool language models.

Taken together, these results show that even under tight resource budgets, intelligent stacking of small, complementary components can materially improve automatic fact-checking and form a robust foundation for future large-scale deployments in an AI-saturated information landscape.

Tables and Figures

Figure 1. Full pipeline of Fake News Detection using NLP
Figure 2. Performance on Three-Class Task (FAST_DEV)
Figure 3. Stacking Meta-Classifier Performance

References

1. Diaz Ruiz, Carlos, and Tomas Nilsson. "Disinformation and Echo Chambers: How Disinformation Circulates in Social Media through Identity-Driven Controversies." *Journal of Public Policy & Marketing*, vol. 42, no. 1, 16 May 2022, pp. 18–35. https://doi.org/10.1177/07439156221103852.

2. Gamage, Dilrukshi, et al. "Designing Credibility Tools to Combat Mis/Disinformation: A Human-Centered Approach." *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 27 Apr. 2022. https://doi.org/10.1145/3491101.3503700.

3. Jones, Dominic Zaun Eu, and Eshwar Chandrasekharan. “Measuring Epistemic Trust: Towards a New Lens for Democratic Legitimacy, Misinformation, and Echo Chambers.” *Proceedings of the ACM on Human-Computer Interaction*, vol. 8, no. CSCW2, 7 Nov. 2024, pp. 1–33. https://doi.org/10.1145/3687001.

4. “Fake News Challenge.” *Fakenewschallenge.org*, 2016, www.fakenewschallenge.org/.

5. “CheckThat!” *Checkthat.gitlab.io*, checkthat.gitlab.io/clef2024/.

6. Wang, William Yang. “‘Liar, Liar Pants on Fire’: A New Benchmark Dataset for Fake News Detection.” *ACLWeb*, Association for Computational Linguistics, 1 July 2017, www.aclweb.org/anthology/P17-2067/.

7. Thorne, James, et al. “FEVER: A Large-Scale Dataset for Fact Extraction and Verification.” *ACLWeb*, Association for Computational Linguistics, 1 June 2018, aclanthology.org/N18-1074/.

8. Shu, Kai, et al. “FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media.” *arXiv*, 1 Jan. 2018. https://doi.org/10.48550/arxiv.1809.01286.

9. Robertson, Stephen. “The Probabilistic Relevance Framework: BM25 and Beyond.” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, 2010, pp. 333–389. https://doi.org/10.1561/1500000019.

10. Karpukhin, Vladimir, et al. “Dense Passage Retrieval for Open-Domain Question Answering.” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. https://doi.org/10.18653/v1/2020.emnlp-main.550.

11. Devlin, Jacob, et al. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, vol. 1, 2019, pp. 4171–4186. aclanthology.org/N19-1423/. https://doi.org/10.18653/v1/n19-1423.

12. Dietterich, Thomas G. “Ensemble Methods in Machine Learning.” *Multiple Classifier Systems: Lecture Notes in Computer Science*, edited by Josef Kittler and Fabio Roli, vol. 1857, Springer, 2000, pp. 1–15.