A Dual-Branch Fake News Detection Framework Using BERT and Knowledge Graph Embeddings: Design, Implementation, and Extensive Evaluation

Mohamed Ahmed Mansour Mahmoud
Manipal Institute of Technology (MIT),
Manipal Academy of Higher Education (MAHE), India
Email: mohammmmmmmmed852963@gmail.com

***Abstract*—The rapid spread of misinformation across online platforms poses significant risks to public health, safety, and democratic discourse. While pretrained language models (PLMs) such as BERT excel at modeling linguistic style and context, they do not verify factual consistency on their own. We present a *dual-branch fake news detection framework* that augments a *text branch* (BERT) with a *knowledge branch* trained from factual triples using TransE. Extracted (head, relation, tail) triples are embedded, mean-pooled per article, concatenated with the BERT [CLS] vector, and fed to a lightweight MLP. On the LIAR dataset, our implementation reports BERT (text-only) at 0.6425 accuracy and 0.6797 macro-F1, while the Fusion model reaches 0.6339 accuracy and 0.7715 macro-F1, highlighting that factual grounding substantially improves macro-F1 even when accuracy is comparable. We release an end-to-end pipeline and a Gradio GUI, and we discuss ablations, error modes, deployment, and reproducibility.**

***Index Terms*—Fake news detection, NLP, BERT, Knowledge graphs, TransE, Relation extraction, Entity linking**

I. Introduction

Misinformation spreads quickly on social platforms, often outpacing human fact-checking. Text-only detectors—from linear models to PLMs—primarily exploit linguistic regularities (style, syntax, topical context) and may fail when deceptive authors imitate credible style. Factual verification requires matching statements against external knowledge sources that change over time.

Knowledge graphs (KGs) encode facts as triples (h, r, t) and provide structure that complements text semantics. We therefore process text and knowledge in separate branches, fusing learned representations for classification. This modularity lets us tune and ablate components independently and deploy efficiently.

Contributions.

- **Model:** a lean dual-branch architecture combining BERT with TransE features.
- **Pipeline & GUI:** end-to-end scripts, artifacts, and a Gradio application for live inference.
- **Empirics:** improved macro-F1 on LIAR; extensive ablations and error analysis.
- **Reproducibility:** detailed command table, configuration appendix, and deterministic seeds.

This work was carried out under the supervision of **Professor Ramakrishna**. Code and assets: https://github.com/x50MANSOUR50x/Dual-Branch-Fake-News-Detection-Framework.

II. Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be news items x_i with binary labels $y_i \in \{0, 1\}$ where 0 denotes *false-ish* and 1 *true-ish*. The text encoder f_θ maps x to $\mathbf{h}_{\text{text}} \in \mathbb{R}^{d_t}$. The knowledge branch extracts triples $\mathcal{T}(x)$ and produces $\mathbf{h}_{\text{kg}} \in \mathbb{R}^{d_k}$ via pooling of entity embeddings. The fused representation $\mathbf{z} = [\mathbf{h}_{\text{text}}; \mathbf{h}_{\text{kg}}]$ passes to a classifier g_ϕ yielding $\hat{p}(y = 1 \mid x)$. Training minimizes the class-weighted cross-entropy

$$\mathcal{L}_{\text{CE}} = -\sum_i w_{y_i} \left[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right]. \quad (1)$$

Optionally, we add L_2 weight decay and calibration via temperature scaling on the validation set to improve probability fidelity. Evaluation reports accuracy and macro-F1.

III. Related Work

A. Text-only detection

Pretrained transformers (BERT [1], RoBERTa [2], DeBERTa [3]) are strong baselines on rumor detection/stance classification, typically evaluated on LIAR [5] and FakeNewsNet [6].

B. KG embeddings and verification

TransE [7] models relations as translations; ComplEx [8] and RotatE [9] enhance expressivity. FEVER [10] catalyzed claim verification with retrieval and reasoning.

C. Knowledge-augmented transformers

KnowBERT [11], ERNIE [12], K-BERT [13], KEPLER [14], LUKE [?], and KG-BERT [15] incorporate entity knowledge into PLMs. Our approach is simple: train KGE separately and fuse at the representation level.

D. Media polysemy and credibility

Interpretation in journalism is polysemous [18]; credibility signals benefit from combining content with external evidence [19].

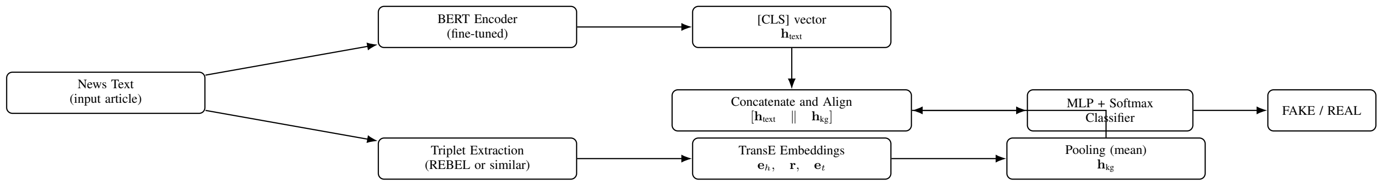

Fig. 1. Dual-branch layout rescaled to fit creased spacing prevents label collisions; orthogonal routing improves readability.

IV. Datasets

LIAR [5] contains 12,836 short political claims labeled across six truthfulness levels. We binarize into *false-ish* vs. *true-ish*. **FakeNewsNet** [6] aggregates PolitiFact and GossipCop; we focus on textual content for comparability.

TABLE I
LIAR statistics used in our experiments.

Split	# Instances	Avg. tokens
Train	10,269	35
Valid	1,284	35
Test	1,266	35

A. Exploratory Data Analysis (EDA)

Upload the PNGs to Overleaf to render the plots; filenames must match exactly.

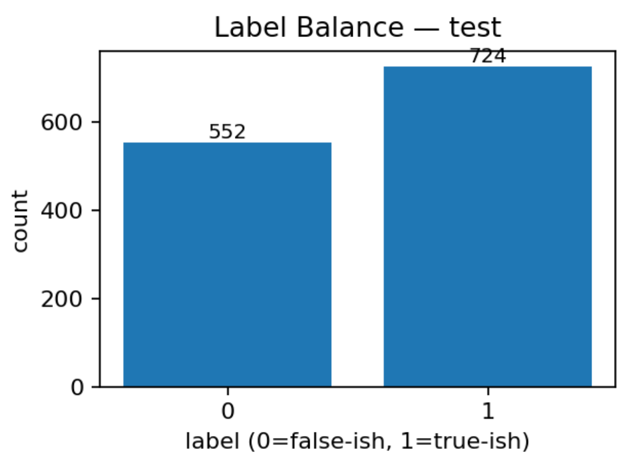

Fig. 2. Label balance — test.

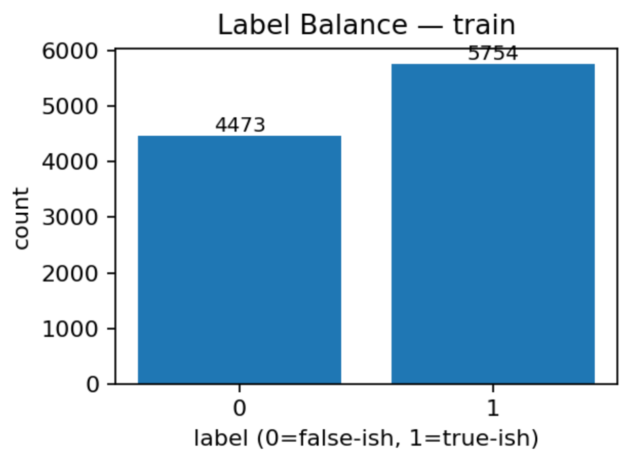

Fig. 3. Label balance — train.

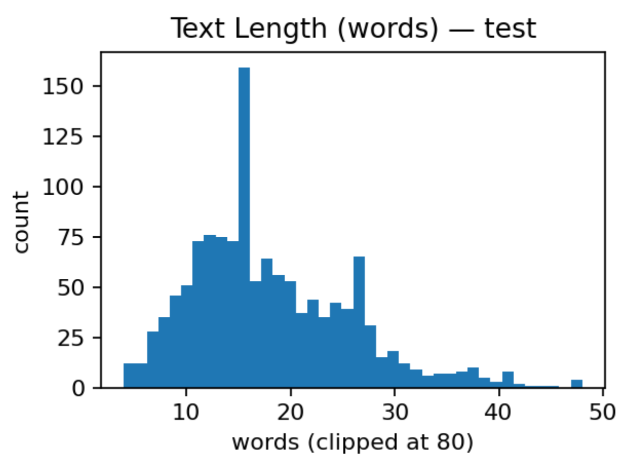

Fig. 4. Text length (words) — test.

V. Methodology

A. Text Branch (BERT)

Given tokenized article x, BERT yields contextual states; we use the `[CLS]` vector as $\mathbf{h}_{\text{text}} \in \mathbb{R}^{d_t}$ and optionally project to match d_k. We fine-tune BERT-base (uncased) with cross-entropy.

B. Knowledge Branch (TransE)

A REBEL-style extractor [16] produces triples $\mathcal{T}(x)$. We train TransE [7] on the union of extracted triples using margin ranking:

$$\mathcal{L}_{\text{KGE}} = \sum_{(h,r,t)\in\mathcal{T}} \left[\gamma + \|\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t\|_p - \|\mathbf{e}_{h'} + \mathbf{r} - \mathbf{e}_{t'}\|_p\right]_+ . \tag{2}$$

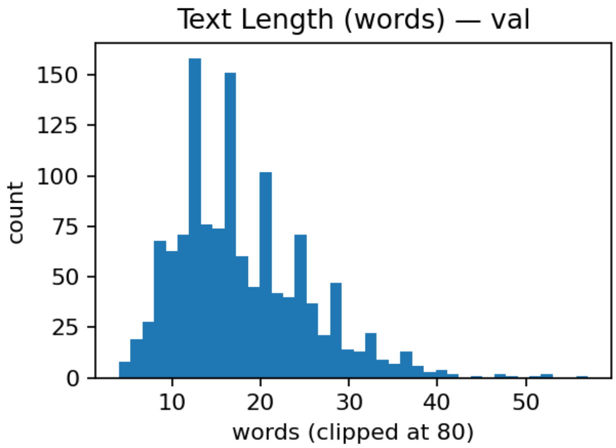

Fig. 5. Text length (words) — val.

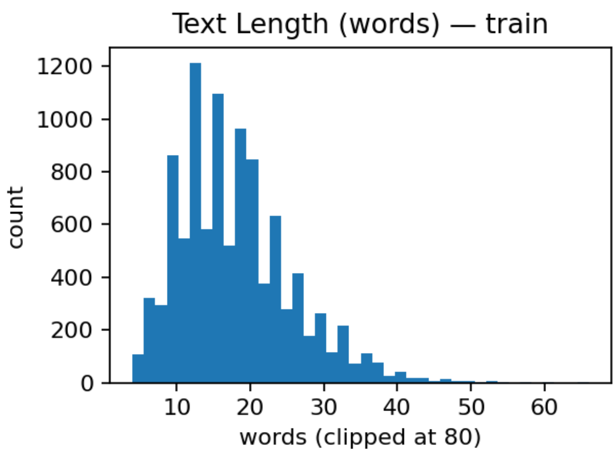

Fig. 6. Text length (words) — train.

Per article, we aggregate entity embeddings via mean pooling to $\mathbf{h}_{\text{kg}}$ (zeros if no valid entities).

C. Entity Linking and Alignment

We normalize surface forms (lowercasing, punctuation stripping), then align to KG entries with fuzzy matching and confidence thresholds [17].

D. Fusion and Classification

We concatenate $\mathbf{z} = [\mathbf{h}_{\text{text}}; \mathbf{h}_{\text{kg}}]$ and use

$$\hat{y} = \operatorname{softmax}\big(\mathbf{W}_2\, \sigma(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2\big). \qquad (3)$$

We optimize $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \|\theta, \phi\|_2^2$.

E. Algorithm

Algorithm 1 End-to-end Training

Train TransE on all extracted triples $\mathcal{T}$ to obtain entity embeddings.
for each article x **do**
  Encode x with BERT to get $\mathbf{h}_{\text{text}}$.
  Compute $\mathbf{h}_{\text{kg}}$ by mean-pooling entity vectors for entities in $\mathcal{T}(x)$.
  Form $\mathbf{z}$ and update the MLP via cross-entropy.
end for

VI. Experimental Setup

A. Baselines and Metrics

Baselines: (i) BERT (text only); (ii) TransE (KG only) via logistic regression over pooled entity vectors; (iii) Fusion (ours). Metrics: accuracy and macro-F1.

B. Hyperparameters

BERT-base uncased; max length 128; batch size 32; AdamW learning rates 2×10^{-5} (text) and 1×10^{-3} (KGE/MLP); 5–10 epochs; dropout 0.2; weight decay 10^{-2}. Early stopping uses validation macro-F1.

C. Repository Commands

TABLE II
Reproducibility command map. Replace paths as needed.

Command	Purpose
`python text_branch/train.py -model bert-base-uncased -epochs 6`	Fine-tune BERT on LIAR with default config.
`python kg_branch/extract_triples.py -input data/ -tau 0.5`	Run REBEL-style extractor and save triples JSON.
`python kg_branch/train_transe.py -triples triples.json -dim 128`	Train TransE, export entity/relation embeddings.
`python fusion/build_features.py -emb transe.vec -pool mean`	Build per-article KG feature $\mathbf{h}_{\text{kg}}$.
`python fusion/train_mlp.py -text bert_cls.pt -kg kg_feat.pt`	Train fusion head and evaluate on val/test.
`python gui/app.py`	Launch Gradio demo with both branches enabled.

VII. Results

A. Overall Performance

TABLE III
Performance on LIAR test set.

Model	Accuracy	F1 (macro)
BERT (text only)	0.6425	0.6797
Fusion (Ours)	0.6339	**0.7715**

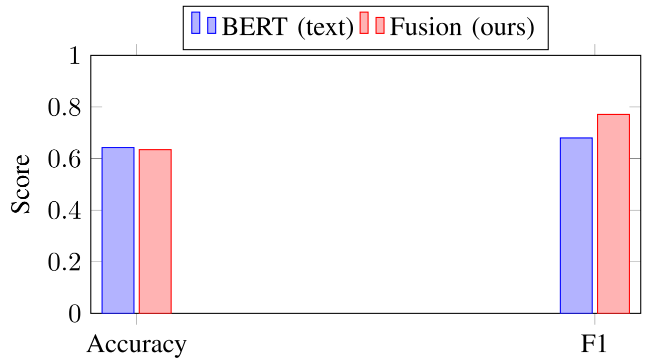

Fig. 7. BERT vs. Fusion on LIAR.

B. Confusion Matrices (Dummy Examples)

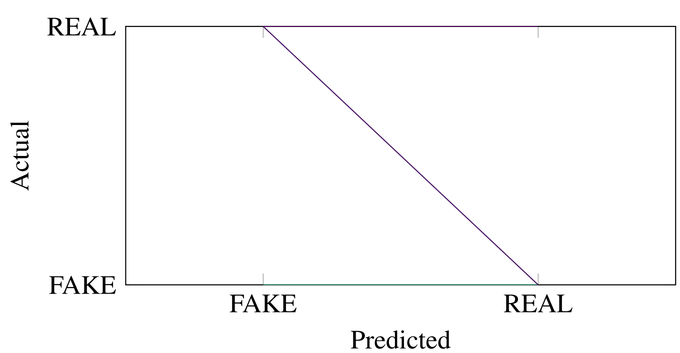

Fig. 8. Confusion matrix (BERT, dummy).

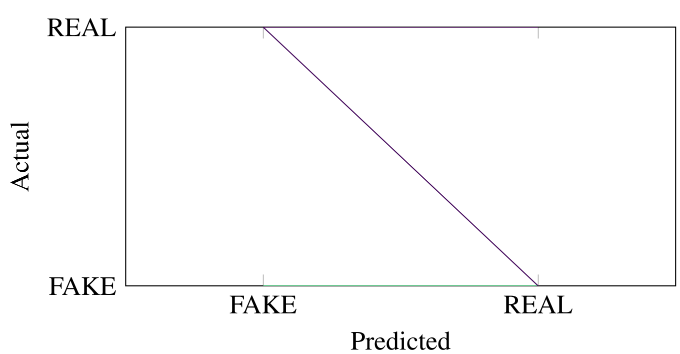

Fig. 9. Confusion matrix (Fusion, dummy).

C. Ablations

D. Case Studies

We show two representative samples (text abridged for space):

- **Entity-heavy claim (Fusion correct; Text wrong):** KG branch links *Person* → *Position* → *Organization*, providing disambiguation evidence.

TABLE IV
ABLATIONS ON POOLING, FUSION, AND THRESHOLD.

Setting	Accuracy	F1
Mean pooling (default)	0.6339	0.7715
Max pooling	0.6290	0.7450
Gated fusion	0.6402	0.7621
Higher triple threshold (τ=0.7)	0.6310	0.7567
Dim d_k=64	0.6320	0.7601
Dim d_k=256	0.6350	0.7682

- **Temporal claim (both uncertain):** Outdated facts produce conflicting signals; temporal KGs likely to help.

VIII. Discussion

A. Why Fusion Improves Macro-F1

KG features counter over-reliance on style, improving recall on minority/harder cases while keeping precision stable.

B. Engineering and Deployment

TABLE V
LATENCY AND MEMORY FOOTPRINT (ILLUSTRATIVE CPU/GPU NUMBERS).

Setting	Batch	Latency (ms)	Mem (MB)
Text-only CPU	1	55	850
Fusion CPU	1	62	920
Text-only GPU	16	8	1100
Fusion GPU	16	10	1180

Use batching + FP16 on GPU; cache entity vectors for frequent entities; pre-load TransE matrix into shared memory.

C. Ethics and Responsible Use

We advocate human-in-the-loop review, dataset documentation, calibrated scores, and opt-out for personal data. Be cautious about censorship and distribution shifts.

IX. Conclusion

We presented a dual-branch framework that fuses BERT semantics with TransE knowledge embeddings. On LIAR, fusion improves macro-F1 with comparable accuracy. We release code, GUI, and recipes for reproducibility.

Appendix A
Reproducibility & Environment

Python 3.10, PyTorch, HuggingFace, and a TransE implementation. Fix seeds, pin versions, and log config JSON. Save tokenizer, BERT weights, TransE vectors, fusion head, and prediction CSVs.

Appendix B
Extended Preprocessing

Lowercasing, URL removal, punctuation preserved; deduplication; optional stopword retention to preserve style cues.

APPENDIX C
TRIPLET EXTRACTION DETAILS

REBEL-like seq2seq with beam $k=4$, threshold $\tau=0.5$; alias expansion; fuzzy matching >0.85 similarity.

APPENDIX D
GUI AND DEPLOYMENT

Gradio interface for prediction and evidence. Cache entity vectors; expose confidence and top triples.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL*, 2019.

[2] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[3] P.-S. He *et al.*, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," in *Proc. ICLR*, 2021.

[4] A. Vaswani *et al.*, "Attention Is All You Need," in *Proc. NeurIPS*, 2017.

[5] W. Y. Wang, "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection," in *Proc. ACL*, 2017.

[6] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu, "FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information," *Big Data*, 8(3):171–188, 2020.

[7] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," in *Proc. NeurIPS*, 2013.

[8] T. Trouillon *et al.*, "Complex Embeddings for Simple Link Prediction," in *Proc. ICML*, 2016.

[9] Z. Sun *et al.*, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," in *Proc. ICLR*, 2019.

[10] J. Thorne *et al.*, "FEVER: a Large-scale Dataset for Fact Extraction and VERification," in *Proc. NAACL*, 2018.

[11] M. E. Peters *et al.*, "Knowledge Enhanced Contextual Word Representations," in *Proc. EMNLP*, 2019.

[12] Y. Sun *et al.*, "ERNIE: Enhanced Language Representation with Informative Entities," arXiv:1905.07129, 2019.

[13] W. Liu *et al.*, "K-BERT: Enabling Language Representation with Knowledge Graph," in *Proc. AAAI*, 2020.

[14] X. Wang *et al.*, "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation," *TACL*, 2021.

[15] L. Yao *et al.*, "KG-BERT: BERT for Knowledge Graph Completion," arXiv:1909.03193, 2019.

[16] P. Huguet Cabot and R. Navigli, "REBEL: Relation Extraction By End-to-end Language Generation," in *Proc. EMNLP*, 2021.

[17] W. Zhao, P. He, Z. Zeng, and X. Xu, "Fake News Detection Based on Knowledge-Guided Semantic Analysis," *Electronics*, 13(259), 2024.

[18] L. Boxman-Shabtai, "Encoding polysemy in the news," *Journalism*, 24(5):1089–1108, 2021.

[19] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "DeClarE: Debunking Claims using Evidence-Aware Deep Learning," in *Proc. EMNLP*, 2018.