Comparing Transformers vs. Classical ML Models for Fine-Grained Fact-Checking on the LIAR Dataset

Arabind Meher, Uday Pothuri, Alan Uthuppan, Manan Patel

Meet the Team

Member	Core Skills	Deliverables
Arabind Meher	NLP, Data Science, Network Analysis	BERT/RoBERTa fine-tuning, hyper-param search
Uday Pothuri	NLP, Deep Learning, Data Science	Data cleaning scripts, LSTM architecture & training
Alan Uthuppan	GenAI, Sentiment Analysis	Feature engineering, EDA plots
Manan Patel	Neural Networks, Ethical AI	LR baseline, literature review, evaluation dashboard

Project Overview

Misinformation erodes public trust; automatic fact-checking at scale is crucial

Compare cheap classical baselines to modern deep-learning methods (LSTM, BERT, RoBERTa) and quantify gains

Probe whether adding rich handcrafted features + metadata boosts performance over text-only transformers

Goal: build a multi-class classifier that labels each statement with one of six credibility tiers from the LIAR benchmark

Task & Motivation

1

Task: Multi-class text classification with 6 ordinal credibility labels

- pants-on-fire · false · barely-true · half-true · mostly-true · true

2

Inputs

- Statement text (≈ 17 tokens on avg.)
- Context metadata — speaker, party, subject tags, venue, prior ruling counts

3

Output: Softmax-based probability vector → highest-probability label

4

Evaluation: Macro-/weighted-F1 + confusion matrix; focus on minority-class recall

5

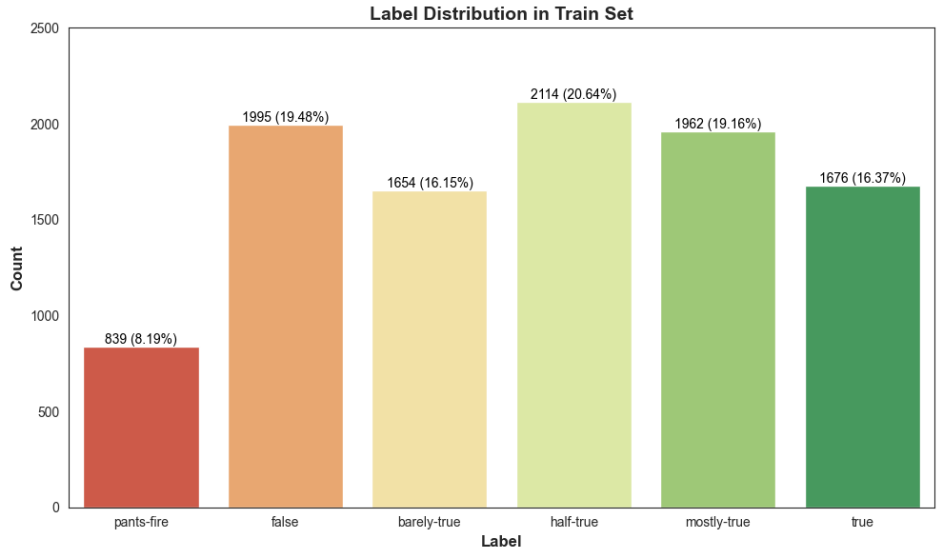
Differs from binary fake-news detection by preserving nuance (e.g., "half-true")

Dataset Overview

- Source:
  - HuggingFace (https://huggingface.co/datasets/ucsbnlp/liar)
  - Kaggle (https://www.kaggle.com/datasets/doanquanvietnamca/liar-dataset)
- Purpose:
  - Benchmark dataset for fake news / misinformation classification
- Size:
  - Total: 12,836 statements
  - Train: 10,269 | Val: 1,284 | Test: 1,283
- Fields:
  - statement (text)
  - label (true, mostly true, half true, mostly false, false, pants-on-fire)
  - speaker, party, context, subject, justification
- Example Entry:
  - “The economy is growing faster than ever before” → Label: false

Data Distribution

- Bar chart of label counts (train/val/test combined)
- Observations:
  - "true" and "mostly true" are less frequent
  - "pants-on-fire" is rare → requires class weighting or balancing
- Token Length Distribution:
  - Histogram of number of tokens per statement
  - Most statements are short → median ~10–15 tokens
- Challenges:
  - Short text → less context for Transformer models



Data Cleaning

Initial Column Selection:

- Retained columns: label, statement, subject, speaker, party_affiliation, context

Handling Missing Values:

- Missing values filled with "unknown" in all splits

Statement Cleaning:

- Converted to lowercase
- Removed punctuation
- Replaced hyphens with spaces
- Removed stopwords
- Removed extra whitespace → created statement_clean column

Metadata Cleaning:

- Converted all categorical columns (except label, statement) to lowercase
- Example columns cleaned: subject, speaker, party_affiliation, context

Additional Cleanup:

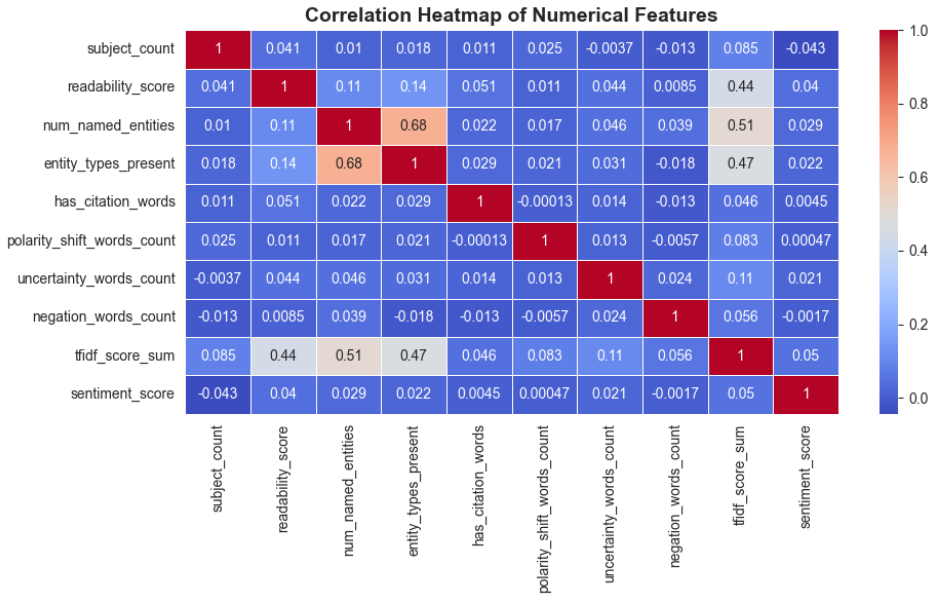
- Replaced original statement with cleaned version
- Created subject_count: number of topics listed in subject

Feature Engineering

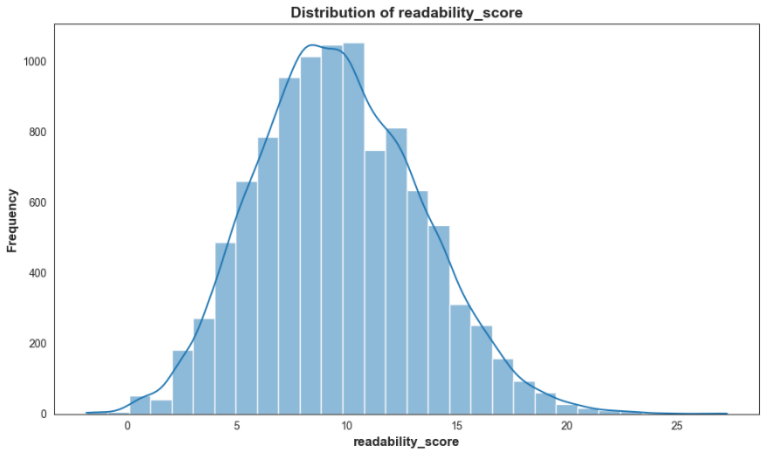
Readability Features:	• readability_score: Flesch-Kincaid grade level of statement
NER-based Features:	• num_named_entities: number of named entities • entity_types_present: number of unique entity types
Citation Words:	• has_citation_words: binary indicator if citation-like phrases present (e.g., "according to", "sources say")
Sentiment:	• sentiment_score: VADER compound sentiment score of cleaned statement
TF-IDF Summary:	• tfidf_score_sum: sum of TF-IDF weights for cleaned statement
Polarity/Uncertainty/Negation Features:	• polarity_shift_words_count: e.g., "however", "but" "although“ • uncertainty_words_count: e.g., "might", "could", “ • possibly"negation_words_count: e.g., "not", "never", "no“
Final Export:	• Feature columns added to train.csv, validation.csv, test.csv for model training

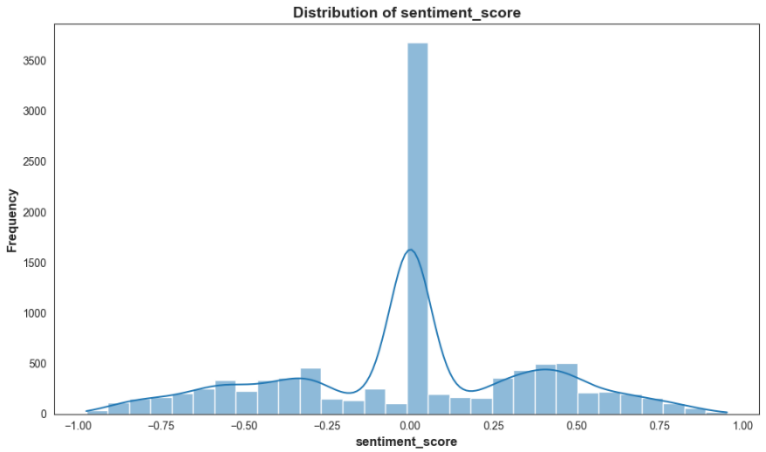
Exploratory Data Analysis

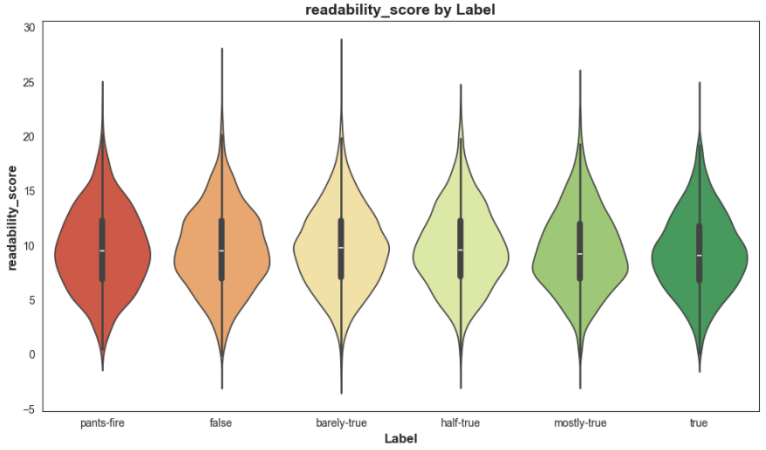
- Key observation:
  - num_named_entities and entity_types_present are strongly correlated (0.68) — more named entities naturally lead to more unique entity types.
  - readability_score moderately correlates with:
    - tfidf_score_sum (0.44): higher readability may result in more distinctive terms picked up by TF-IDF.
    - num_named_entities (0.11) and entity_types_present (0.14) — possibly because more readable text tends to mention more entities.
  - All other feature pairs show low correlation (< 0.5)

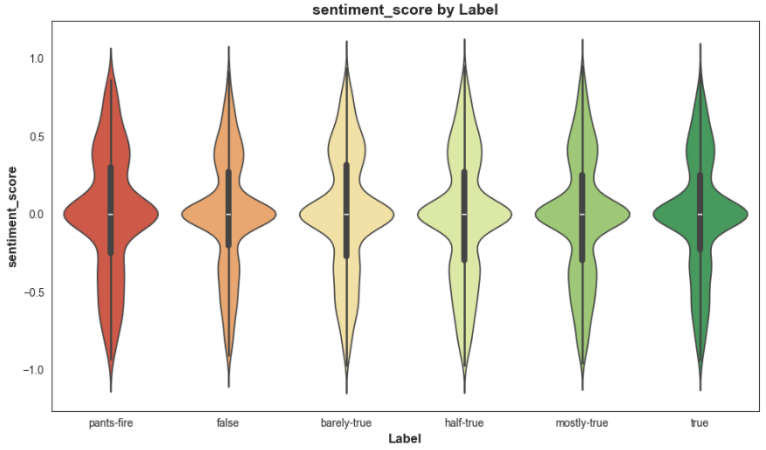


Exploratory Data Analysis









Related Work & Baselines

Earlier LIAR studies: logistic regression + n-grams; CNN/LSTM variants; not yet strong on rare classes

Our baselines:

- Logistic Regression + TF-IDF over statement text & metadata (GridSearchCV)
- Vanilla LSTM with GloVe embeddings + numeric & categorical features concatenated

These establish our "classical" ceilings

Model Architectures

- Logistic Regression:
  - ColumnTransformer → TF-IDF, One-Hot, StandardScaler → LR (multinomial)
- Feature-Fusion LSTM:
  - Text → Embedding → Bi-LSTM
  - Numeric features (10) → dense
  - Categorical features (4) → learned embeddings
  - Concatenate → FC + Softmax
- Transformers:
  - BERT-base-uncased and RoBERTa-base; final pooled token → classifier head
  - Max len = 128; lr = 2e-5, 10 epochs, early stopping on val F1
- **Why Transformers?**
  - Bidirectional context, pre-training on 3B+ tokens → better capture subtle cues (e.g., negations)

Logistic Regression Process

- **Objective**: Build a classical ML baseline using Logistic Regression with engineered features.
- **Data Loading**:
  - Loaded preprocessed train data, validation, test datasetsets with (15 features, 6-class target): label_id.
- **Preprocessing**:
  - **Text**: statement → TfidfVectorizer
  - **Categorical**: subject, speaker, party_affiliation, context, → OneHotEncoder
  - **Numeric**: 10 engineered features → StandardScaler
  - Combined using ColumnTransformer
- **Pipeline**:
  - Preprocessor + Logistic Regression (balanced, weights, multinomial, lbfgs, solver, random_state=691)).
- **Hyperparameter Tuning**:
  - GridSearchCV (5-fold CV, weighted F1 scoring)
  - Tuned: TF-IDF max_features (5000, 10000, 20000), ngram_range ((1,1), (1,2), (1,3)), LR C, (0.01, 0.1, 0.1, 1, 10)
- **Evaluation Setup**:
  - Fit on training settrain, evaluated on validation/test with accuracy, weighted F1, confusion matrices (normalized heatmap saved).

LSTM Process

- **Objective**: Build a feature-fusion LSTM baseline leveraging text, metadata, and engineered features.
- **Data Loading**:
  - Loaded preprocessed train, validation, and test datasets (15 features, 6-class target: label_id).
- **Preprocessing**:
  - **Text**: Tokenized statement, built vocabulary, encoded with GloVe embeddings (100D).
  - **Categorical**: subject, speaker, party_affiliation, context → LabelEncoder.
  - **Numeric**: 10 engineered features → StandardScaler.
  - Custom LIARDataset and DataLoader (batch size 64, dynamic padding).
- **Model**:
  - GloVe embedding → LSTM (hidden_dim=128) → Concatenate with numeric features and categorical embeddings (16D each) → FC layers (Linear→ReLU→Dropout→Linear).
- **Training**:
  - CrossEntropyLoss, Adam (lr=1e-4), ReduceLROnPlateau scheduler, early stopping (patience=5).
  - Up to 100 epochs, random seed 691, GPU if available.
- **Evaluation Setup**:
  - Classification report and confusion matrix (heatmap) on validation/test sets.

BERT Process

- **Objective**: Fine-tune BERT for text-only classification on the LIAR dataset.
- **Data Loading**:
  - Loaded preprocessed train, validation, and test datasets (statement text, 6-class target: label_id).
- **Preprocessing**:
  - **Text**: Tokenized statement using BertTokenizer (from bert-base-uncased, max length 128, padded/truncated).
  - Custom LIARBertDataset and DataCollatorWithPadding for batching.
- **Model**:
  - BertForSequenceClassification (from bert-base-uncased, 6 classes) with custom CrossEntropyLoss.
  - 12-layer transformer, classification head on [CLS] token.
- **Training**:
  - Hugging Face Trainer: lr=2e-5, 10 epochs, batch size 16/32, early stopping on validation loss.
  - Adam optimizer, weight decay 0.01, random seed 691.
- **Evaluation Setup**:
  - Accuracy, weighted F1, precision, recall on validation/test sets; confusion matrix heatmap.

RoBERTa Process

- **Objective**: Fine-tune a hybrid RoBERTa model with text, metadata, and engineered features for classification on the LIAR dataset.
- **Data Loading**:
  - Loaded preprocessed train, validation, and test datasets (statement text, 4 categorical, 10 numeric features, 6-class target: label_id).
- **Preprocessing**:
  - **Text**: Tokenized statement using RobertaTokenizer (from roberta-base, max length 128, padded/truncated).
  - **Categorical**: subject, speaker, party_affiliation, context encoded with LabelEncoder.
  - **Numeric**: 10 engineered features scaled with StandardScaler.
  - Custom HybridRoBERTaDataset and DataCollatorWithPadding for batching.
- **Model**:
  - HybridRoBERTaClassifier: RoBERTa (roberta-base) [CLS] token + categorical embeddings (16D each) + numeric features → FC layers (842→256→6).
- **Training**:
  - Hugging Face Trainer: lr=2e-5, 10 epochs, batch size 16/32, early stopping on validation loss.
  - Adam optimizer, weight decay 0.01, random seed 691.
- **Evaluation Setup**:
  - Accuracy, weighted F1, precision, recall on validation/test sets; test set confusion matrix heatmap.

Training Process & Results

Model	Validation F1 (Weighted Average)	Test F1 (Weighted Average)	Validation Accuracy	Test Accuracy
Logistic Regression	0.28	0.26	0.29	0.26
LSTM	0.27	0.23	0.29	0.25
BERT	0.19	0.18	0.25	0.24
RoBERTa	0.24	0.26	0.27	0.28

Limitations

Data Limitations

- The LIAR dataset consists of short political statements—real-world misinformation is often longer and more nuanced.
- The dataset lacks dynamic context (e.g., how posts evolve over time or how users interact with them).

Model Limitations

- BERT and RoBERTa are powerful, but still primarily capture linguistic cues.
- They do not inherently leverage social or propagation-based signals (which are crucial in misinformation spread).

Generalizability

- Trained models may not generalize well to other domains (e.g., health misinformation, non-political news).
- Platform-specific characteristics (Twitter, Facebook, YouTube) are not modeled.

Multimodality

- Current project focuses on text only — no image, video, or metadata fusion.
- Many modern misinformation instances are multimodal.

Future Work

Expand Dataset Scope

- Incorporate larger and more diverse datasets beyond LIAR.
- Collect time-evolving data (to study propagation and temporal features).

Multimodal Models

- Combine text with images, videos, and external knowledge sources.
- Explore architectures like **multimodal transformers**.

Incorporate Social Signals

- Leverage graph-based models to integrate user interactions and propagation patterns.
- Study how misinformation spreads through communities.

Cross-Platform Adaptation

- Develop domain-adaptive models to handle content across different platforms.
- Fine-tune models for specific platform characteristics (e.g., meme-heavy content on Instagram vs. tweet threads on Twitter).

Explainability

- Improve model transparency to aid human moderators.
- Use explainable AI techniques to highlight why content was flagged as misinformation.

THANK YOU