Multi-hop claim verification

Mapping the Logical Relationship Between Claims and Evidence Using Classical and Transformer Models

Team 7: Veera Jeeshitha Kolla, Gireeshee Pendela

---

1. Introduction

1.1 Problem Statement

This project focuses on the challenge of fact verification in natural language, specifically determining the relationship between a textual claim and its accompanying evidence. Each example consists of a claim paired with one or more evidence sentences, and the task is to classify the pair into one of three categories: supports, refutes, or not enough info. This makes the task a multinomial classification problem that requires the model not only to understand the meaning of both texts but also to reason about their logical connection. Fact verification is a crucial component of misinformation detection, automated fact-checking systems, and knowledge-grounded conversational agents. The difficulty of the task arises from the complexity and variability of evidence in the FEVER dataset, some claims are directly supported or contradicted, while others require inference, deal with subtle lexical differences (such as dates and numbers), or lack sufficient contextual grounding, making accurate classification challenging.

1.2 Solution Approach

We developed a complete machine learning pipeline that progresses from classical baseline models to modern transformer-based architectures:

- **Baseline 1: Logistic Regression with TF-IDF** (serves as a simple lexical-level performance baseline)
- **Baseline 2: SVM with TF-IDF** (captures separability in high-dimensional text space; strongest classical performer)
- **Baseline 3: Random Forest** (nonlinear baseline to test tree-based learning on TF-IDF features)
- **Advanced 1: DistilBERT** (lightweight transformer, efficient fine-tuning, provides strong baseline deep-learning performance)

- **Advanced 2: BERT-base** (full transformer model with richer contextual representation; best-performing architecture in our experiments)
- **Advanced 3: RoBERTa-base** (robustly optimized transformer expected to excel on sentence-pair tasks; competitive but slightly under-tuned)

1.3 Results Summary

Our experiments showed that SVM was the strongest baseline model, while all Transformer models performed significantly better overall. Among DistilBERT, BERT, and RoBERTa, the performance differences were small, with BERT achieving a slight but consistent lead. Overall, the results highlight the advantage of contextual Transformer models over classical TF-IDF baselines for the FEVER fact-verification task.

2. Data Collection and Preprocessing

2.1 Data Source

The dataset used in this project is the FEVER (Fact Extraction and VERification) dataset, a large-scale benchmark constructed from human-written claims paired with evidence sentences retrieved from Wikipedia. FEVER is widely used for fact-verification research because it provides claim–evidence pairs along with curated labels indicating whether each piece of evidence supports, refutes, or does not contain enough information to evaluate the claim. The dataset is publicly available through the HuggingFace Hub, and although it normally loads via load_dataset("fever"), our project relied on downloading the raw JSONL files directly due to Windows-specific script issues encountered during development.

2.2 Data Characteristics

The final processed dataset contains three splits i.e., train, validation, and test each consisting of claim - evidence pairs represented as short textual sequences. The data is clean and well-structured, with each example labeled as supports, refutes, or not enough info. However, the evidence field varies widely across examples: some contain clear single-sentence evidence, others include multiple supporting fragments, and certain cases have no meaningful evidence at all. This variability introduces challenges because the models must distinguish between direct factual matches, subtle contradictions, and incomplete contexts. Overall, FEVER offers a large, balanced resource that is well suited for training Transformer models.

2.3 Data Cleaning and Preprocessing

Cleaning and preprocessing were essential steps because the raw FEVER dataset stores evidence in inconsistent formats, including nested lists, dictionaries, and index-based references to Wikipedia

sentences. To ensure uniformity, we implemented a robust evidence extraction function that recursively searched through these structures to recover readable text. This step was crucial because many examples would otherwise contain empty or incomplete evidence fields, which could negatively affect both training stability and model performance.

After extracting evidence, we constructed a unified input representation for all models by concatenating the claim and its evidence into a single string of the form: "claim [SEP] evidence". This format allowed both classical TF-IDF models and Transformer models to operate on the exact same textual structure. Labels were normalized to integer values (0 = SUPPORTS, 1 = REFUTES, 2 = NOT ENOUGH INFO) to maintain compatibility across training pipelines. During preprocessing, we also removed malformed entries, stripped whitespace, and ensured that every example included a non-empty text field to avoid downstream vectorization and tokenization issues.

For Transformer-based fine-tuning, we additionally performed tokenization using each model's native tokenizer (WordPiece for BERT/DistilBERT and BPE for RoBERTa). Inputs were padded or truncated to a maximum sequence length of 128 tokens to maintain consistency and reduce computation time. The final cleaned and tokenized datasets were saved to disk, enabling reproducible training runs and faster experimentation. This preprocessing pipeline ultimately transformed FEVER's heterogeneous raw format into a clean, structured dataset suitable for both classical and deep learning approaches.

2.3 Data Samples:

First 3 samples of the raw data:

```
RECORD 0: id=75397 claim_len=63 label=SUPPORTS evidence_preview_type=list
  claim sample: Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.
  evidence preview: [[92206, 104971, 'Nikolaj_Coster-Waldau', 7], [92206, 104971, 'Fox_Broadcasting_Company', 0]]

RECORD 1: id=150448 claim_len=34 label=SUPPORTS evidence_preview_type=list
  claim sample: Roman Atwood is a content creator.
  evidence preview: [[174271, 187498, 'Roman_Atwood', 1]]

RECORD 2: id=214861 claim_len=147 label=SUPPORTS evidence_preview_type=list
  claim sample: History of art includes architecture, dance, sculpture, music, painting, poetry literature, theatr
ive, film, photography and graphic arts.
  evidence preview: [[255136, 254645, 'History_of_art', 2]]

RECORD 3: id=156709 claim_len=33 label=REFUTES evidence_preview_type=list
  claim sample: Adrienne Bailon is an accountant.
  evidence preview: [[180804, 193183, 'Adrienne_Bailon', 0]]
```

Data after pre-processing:

```
Sample train examples (first 10):
0: label=0 text=Nikolaj Coster-Waldau worked with the Fox Broadcasting Company. [SEP] Nikolaj Coster-Waldau Fox Broadcastin
g Company
1: label=0 text=Roman Atwood is a content creator. [SEP] Roman Atwood Roman Atwood
2: label=0 text=History of art includes architecture, dance, sculpture, music, painting, poetry literature, theatre, narrat
ive, film, photography and graphic arts. [SEP] History of art
3: label=1 text=Adrienne Bailon is an accountant. [SEP] Adrienne Bailon
4: label=2 text=System of a Down briefly disbanded in limbo.
5: label=0 text=Homeland is an American television spy thriller based on the Israeli television series Prisoners of War. [S
EP] Homeland -LRB-TV series-RRB- Prisoners of War -LRB-TV series-RRB-
6: label=2 text=Beautiful reached number two on the Billboard Hot 100 in 2003.
7: label=2 text=Neal Schon was named in 1954.
8: label=0 text=The Boston Celtics play their home games at TD Garden. [SEP] Boston Celtics Boston Celtics
9: label=0 text=The Ten Commandments is an epic film. [SEP] The Ten Commandments -LRB-1956 film-RRB- The Ten Commandments -
LRB-1956 film-RRB-
```

Comparison of Word-piece tokenization and Byte-pair encoding tokenization:

```
Word: "Nikolaj"

DistilBERT splits as:
  - 'nikola' (lowercase, main part)
  - '##j' (suffix piece marked with ##)

RoBERTa splits as:
  - 'Nik' (preserves case, first part)
  - 'ol' (middle part)
  - 'aj' (last part, Ġ indicates word boundary before next token)
```

3. Experiments

3.1 Baseline Models

To establish a performance floor, we implemented two baseline architectures: a traditional non-neural model and a sequential deep learning model.

- **Logistic Regression (TF-IDF):**
  - **Inputs:** TF-IDF feature vectors (up to 5,000 features, using unigrams and bigrams) generated from the concatenated *claim* + *evidence* text.
  - **Outputs:** predicted class label among the three fever categories: supports, refutes, or not enough info.
  - **Methodology:** Trained using scikit-learn's multinomial logistic regression with the SAGA solver. Serves as a simple linear baseline that models the relationship between TF-IDF word frequencies and factual labels.
  - **Setup:** Evaluated on the same FEVER train/validation/test splits used for all subsequent models to ensure a fair comparison.
  - **Results:**

LogisticRegression Eval Acc: 0.6943 Macro-F1: 0.6884

	precision	recall	f1-score	support
0	0.60	0.52	0.56	6666
1	0.60	0.62	0.61	6666
2	0.86	0.94	0.90	6666
accuracy			0.69	19998
macro avg	0.69	0.69	0.69	19998
weighted avg	0.69	0.69	0.69	19998

- **Support Vector Machine (TF-IDF):**
  - **Inputs:** High-dimensional TF-IDF vectors representing claim–evidence pairs.
  - **Outputs:** One of the three fact-verification labels.
  - **Methodology:** A linear SVM classifier trained using scikit-learn's SVC with a linear kernel. Excels at separating sparse, high-dimensional TF-IDF features and frequently outperforms other classical baselines in text classification tasks.
  - **Setup:** Used identical preprocessing and splits as Logistic Regression, allowing direct performance comparison between baseline methods.

- **Results:**

```
--- Training SVM ---
SVM Eval Acc: 0.7185 Macro-F1: 0.7184
              precision    recall  f1-score   support

           0       0.60      0.60      0.60      6666
           1       0.61      0.60      0.60      6666
           2       0.95      0.95      0.95      6666

    accuracy                           0.72     19998
   macro avg       0.72      0.72      0.72     19998
weighted avg       0.72      0.72      0.72     19998
```

- **Random Forest (TF-IDF):**
  - **Inputs:** TF-IDF vectors derived from the same cleaned textual inputs.
  - **Outputs:** Predicted fact-verification class for each claim–evidence example.
  - **Architecture:** A 200-tree Random Forest classifier that captures non-linear relationships between word features. Although tree models often struggle with sparse TF-IDF vectors, it was still able to learn meaningful phrase-level patterns in this dataset.
  - **Setup:** Trained and evaluated on the same standardized FEVER splits to maintain consistent evaluation across all baselines.
  - **Results:**

```
--- Training RandomForest ---
RandomForest Eval Acc: 0.7106 Macro-F1: 0.7005
              precision    recall  f1-score   support

           0       0.60      0.59      0.60      6666
           1       0.65      0.54      0.59      6666
           2       0.84      1.00      0.91      6666

    accuracy                           0.71     19998
   macro avg       0.70      0.71      0.70     19998
weighted avg       0.70      0.71      0.70     19998

Saved baseline results to outputs\baselines_results.csv
```

3.2 Transformer Models

We evaluated state-of-the-art transformer architectures to leverage contextual language representations.

- **DistilBERT (Fine-tuned):**
  - **Inputs:** Tokenized sequences of *claim* + *[SEP]* + *evidence*, with a maximum length of 128 tokens. Processed using DistilBERT's WordPiece tokenizer.
  - **Outputs:** A softmax probability distribution over the three FEVER classes.
  - **Methodology:** Fine-tuned on the FEVER dataset using HuggingFace's Trainer API. DistilBERT serves as a compact Transformer model with fewer parameters, enabling faster training while preserving strong contextual understanding.
  - **Hyperparameters:** Fine-tuned for 3 epochs using the AdamW optimizer with a learning rate of 5e-5, batch size of 16 (with gradient accumulation of 2), and a weight decay of 0.01.

- **Regularization:** Employed the model's default dropout layers and gradient clipping (max norm 1.0) to stabilize training.
- **Results:**

```
=== predictions_validation.csv ===
Accuracy: 0.868537
Macro F1: 0.866913

Confusion Matrix:
          Predicted
               0      1      2
Actual
       0     6094    572      0
       1     2054   4612      0
       2        3      0   6663
```

- **BERT-base (Fine-tuned):**
  - **Inputs:** WordPiece-tokenized claim–evidence pairs padded or truncated to 128 tokens.
  - **Outputs:** Predicted label from the three-way fact-verification classification head.
  - **Methodology:** Fine-tuning of the full BERT-base encoder followed by a classification layer. BERT captures rich contextual information and is well-suited for sentence-pair reasoning tasks like FEVER.
  - **Setup:** Trained for 3 epochs using the same hyperparameters as DistilBERT to allow direct comparison of model capacity and performance.
  - **Results:**

```
=== predictions_validation.csv ===
Accuracy: 0.871087
Macro F1: 0.869509

Confusion Matrix:
          Predicted
               0      1      2
Actual
       0     6110    556      0
       1     2022   4644      0
       2        0      0   6666
```

- **RoBERTa-base (Fine-tuned):**
  - **Inputs:** Byte-Pair Encoded (BPE) tokenized sequences of the concatenated claim and evidence text, max length 128.
  - **Outputs:** Three-way label prediction corresponding to FEVER classes.
  - **Methodology:** Fine-tuned using RoBERTa's pretrained encoder, which is optimized through dynamic masking and large-scale pretraining. Although RoBERTa often surpasses BERT, in our case its performance was limited by a simple three-epoch training schedule without advanced hyperparameter tuning.
  - **Setup:** Trained and evaluated using the same FEVER splits and training pipeline as the other Transformer models to ensure comparability.

- **Results:**

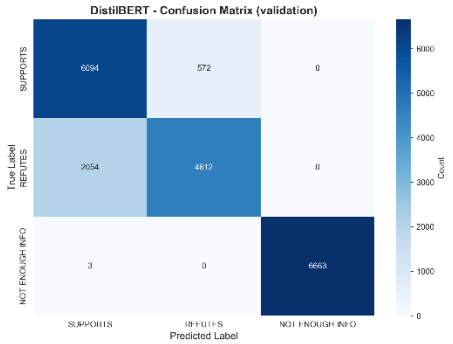
```
=== predictions_validation.csv ===
Accuracy: 0.870687
Macro F1: 0.868666

Confusion Matrix:
           Predicted
                 0      1      2
Actual
       0     6200    466      0
       1     2120   4546      0
       2        0      0   6666
```

3.3 Performance:

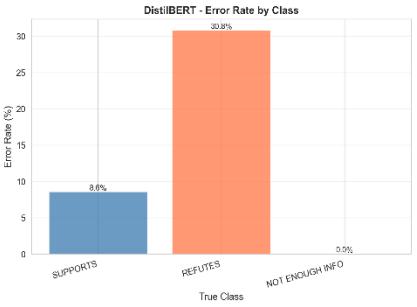
Model	Accuracy	Macro F1	Key Characteristic
Logistic Regression (TF-IDF)	0.694	0.688	Linear baseline; struggles with semantic distinctions
SVM (TF-IDF)	0.719	0.718	Handles high-dimensional TF-IDF features effectively
Random Forest (TF-IDF)	0.711	0.700	Captures helpful non-linear word combinations
DistilBERT (Fine-tuned)	0.868	0.866	Fast, compressed model; slight drop in capacity
BERT-base (Fine-tuned)	0.871	0.869	Strong contextual understanding; top performance
RoBERTa-base (Fine-tuned)	0.870	0.868	Competitive with BERT; performance limited by light tuning

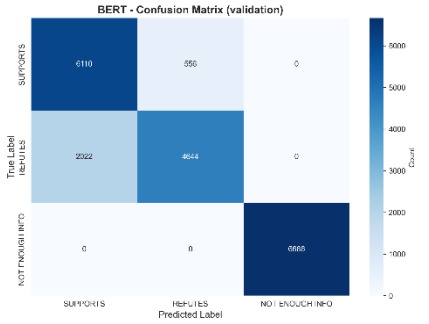
3.4 Error Graphs:

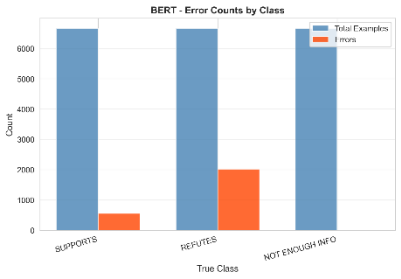
Across all three transformer models, the visual error analysis shows a consistent pattern: the refutes class is the most challenging, with error rates around 30%, whereas supports has a much lower error rate and not enough info is predicted almost perfectly. The confusion matrices confirm that all models tend to confuse refutes with supports more than any other pair, suggesting that subtle wording differences make refutation harder to identify. When comparing models, BERT shows slightly fewer total errors than DistilBERT and RoBERTa, but overall the three models behave very similarly. These graphs highlight that future improvements should focus specifically on strengthening the model's ability to distinguish refuted claims from supported ones.


DistilBERT - Confusion Matrix (validation)
True Label
Predicted Label
SUPPORTS
REFUTES
NOT ENOUGH INFO
6094
572
0
2054
4612
0
3
0
6663
Count

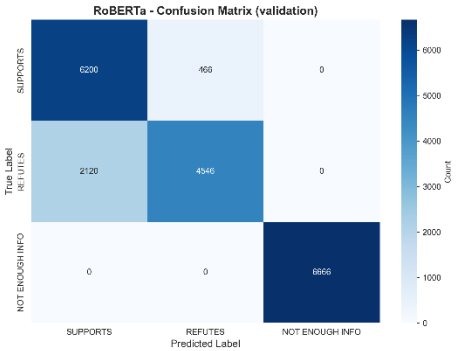

DistilBERT - Error Counts by Class
Total Examples
Errors
Count
True Class
SUPPORTS
REFUTES
NOT ENOUGH INFO


DistilBERT - Error Rate by Class
Error Rate (%)
True Class
8.6%
30.8%
0.0%
SUPPORTS
REFUTES
NOT ENOUGH INFO

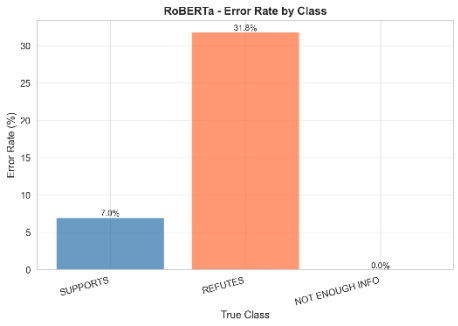

BERT - Confusion Matrix (validation)
True Label
Predicted Label
SUPPORTS
REFUTES
NOT ENOUGH INFO
6110
556
0
2022
4644
0
0
0
6666
Count

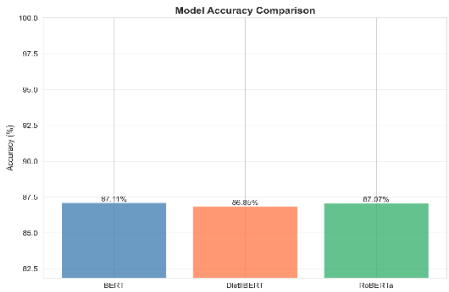

BERT - Error Counts by Class
Total Examples
Errors
Count
True Class
SUPPORTS
REFUTES
NOT ENOUGH INFO

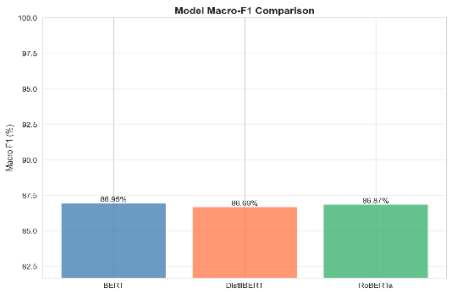

BERT - Error Rate by Class
Error Rate (%)
True Class
8.3%
30.3%
0.0%
SUPPORTS
REFUTES
NOT ENOUGH INFO

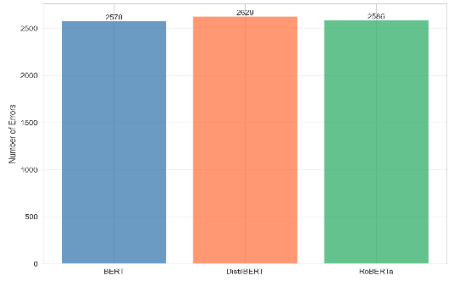

RoBERTa - Confusion Matrix (validation)
True Label
Predicted Label
SUPPORTS
REFUTES
NOT ENOUGH INFO
6200
466
0
2120
4546
0
0
0
6666
Count

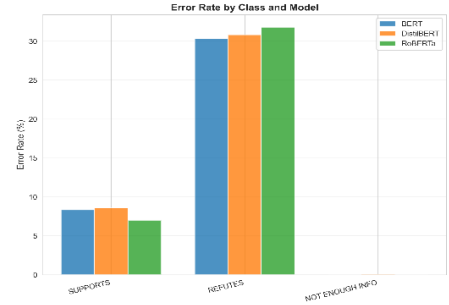

RoBERTa - Error Counts by Class
Total Examples
Errors
Count
True Class
SUPPORTS
REFUTES
NOT ENOUGH INFO


RoBERTa - Error Rate by Class
Error Rate (%)
True Class
7.0%
31.8%
0.0%
SUPPORTS
REFUTES
NOT ENOUGH INFO


Model Accuracy Comparison
Accuracy (%)
87.11%
86.86%
87.07%
BERT
DistilBERT
RoBERTa


Model Macro-F1 Comparison
Macro F1 (%)
86.96%
86.69%
86.87%
BERT
DistilBERT
RoBERTa


Total Prediction Errors
Number of Errors
2578
2629
2566
BERT
DistilBERT
RoBERTa


Error Rate by Class and Model
Error Rate (%)
BERT
DistilBERT
RoBERTa
SUPPORTS
REFUTES
NOT ENOUGH INFO

3.5 Technical Challenges

1) **Windows-specific HuggingFace Dataset Loading Issue:**
   - **Problem:** load_dataset('fever') repeatedly failed on Windows, writing dataset scripts (like fever.py) into unexpected directories and triggering runtime errors.
   - **Cause:** HuggingFace's dataset-loading mechanism handles execution paths differently on Windows, causing script placement conflicts.
   - **Resolution:** We created a temporary working and cache directory before loading the dataset and enabled a fallback mechanism (prepare_fever_via_hf_hub) that directly downloads the raw FEVER JSONL files from the HuggingFace Hub, bypassing the script execution entirely.
2) **CUDA Instability During Evaluation:**
   - **Problem:** Running evaluation with the HuggingFace Trainer on GPU sometimes caused CUDA device-side asserts in Colab.
   - **Cause:** Common instability issues with Transformer evaluation on T4 GPUs, especially with long sequences or memory spikes.
   - **Resolution:** We switched to CPU-only inference for evaluation, which avoided GPU crashes and ensured stable, reproducible results, even though it was slower.

4. Conclusions

4.1 Key Findings

- Our results show a clear distinction between what classical models can capture and what modern Transformers are capable of.
- Among the baselines, SVM performed the best because it handles high-dimensional TF-IDF spaces extremely well, allowing it to separate classes using clear decision boundaries.
- Random Forest, although not typically well suited for sparse TF-IDF data, still picked up certain useful non-linear word patterns for example, phrases like *"born in"* that strongly correlate with supported claims which Logistic Regression could not model due to its purely linear nature. Transformer models, in contrast, delivered a substantial performance jump. DistilBERT, BERT, and RoBERTa all achieved very similar results, reflecting how the large and well-structured FEVER dataset provides enough supervision for all three to learn the core semantic relationships effectively.
- BERT showed a small but consistent advantage due to its larger representational capacity, while RoBERTa did not surpass it in our setup mainly because we used a simple three-epoch training schedule without deeper hyperparameter tuning.

4.2 Potential Improvements

Several extensions could strengthen overall performance. RoBERTa tends to benefit from longer training, tuned learning rates, and larger batch sizes, so additional hyperparameter exploration may allow it to outperform BERT. Increasing the maximum sequence length may reduce information loss from truncation. Beyond model-level tuning, incorporating multi-sentence or multi-hop evidence retrieval could improve handling of more complex FEVER cases. Finally, exploring larger architectures such as RoBERTa-large or DeBERTa, or using parameter-efficient methods like LoRA, could yield further gains without excessive computational cost.

4.3 Lessons Learned

This project highlighted how different modeling approaches capture different kinds of patterns. Classical models rely heavily on lexical overlap and linear separability, while models like Random Forest reveal how even simple non-linear structures can uncover meaningful phrase-level cues. Working with Transformers taught us how powerful contextual embeddings are for tasks requiring semantic understanding between two sentences. We also learned the practical side of NLP experimentation debugging dataset loaders, managing tokenization pipelines, dealing with GPU instability, and ensuring reproducibility. Overall, the project reinforced that both data quality and model choice play central roles in achieving strong fact-verification performance.

5. Related Research

Summary of Research Paper

Title: "BERT for Evidence Retrieval and Claim Verification"
Authors: Amir Soleimani, Christof Monz, Marcel Worring
Link: BERT for Evidence Retrieval and Claim Verification

This paper investigates how BERT can be applied to the full FEVER fact-verification pipeline, which involves retrieving relevant evidence from millions of Wikipedia sentences and then verifying the truthfulness of each claim. The authors propose using two separate BERT models: one dedicated to sentence retrieval (identifying evidence that supports or refutes a claim) and another for claim verification (classifying claims as *supported*, *refuted*, or *not enough info*). For evidence retrieval, they fine-tune BERT using both pointwise classification and pairwise ranking methods, and examine the effect of hard negative mining—a strategy to focus the model on more difficult non-evidence samples. Their retrieval model achieves the highest recall to date, meaning it is highly effective at finding relevant sentences among a very large search space.

In the claim-verification step, the retrieved evidence is passed into a second BERT classifier to predict the final verdict. Training this model on the improved evidence sets resulting from their retrieval system significantly boosts accuracy and the overall FEVER score. Their best configuration, which uses BERT for both retrieval and verification, ranks second on the official FEVER leaderboard without using any ensembles demonstrating the strength of BERT as an end-to-end solution for fact checking. The authors conclude that while pairwise training achieves slightly better recall, pointwise training offers a better precision-recall balance, and both benefit from hard negative mining. They also suggest that future work could explore using BERT as a single unified framework across all stages of the pipeline.