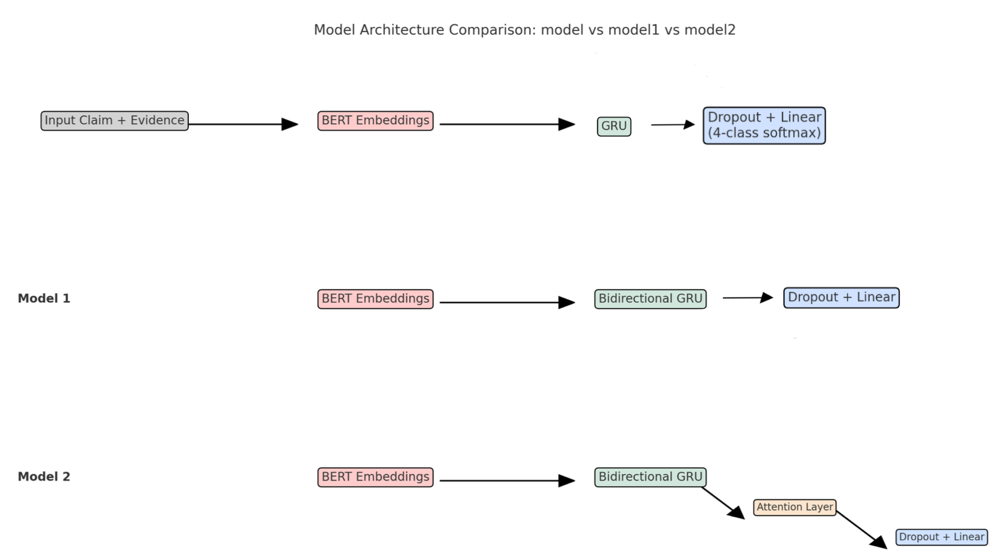
**3.3.3 Classifier on BERT and GRU**

We chose Gated Recurrent Units (GRUs) for their effectiveness in modeling sequence data with lower computational complexity than LSTMs, yet similar performance (Chung et al., 2014). The GRU layer acts as a sequential aggregator over BERT token embeddings, helping the model focus on the relationship between claims and multiple evidence passages over time. We developed a series of models integrating BERT embeddings with GRU-based architectures (**Table 3.3.3-1**).



**Table 3.3.3-1**

All models share a common input structure: the claim and its associated evidence are connected and tokenized using BERT. We extract the hidden state of the [CLS] token or the full sequence output from BERT by the downstream layer’s requirement.

**Base Model: GRU**

This is the baseline model employed a single-layer unidirectional GRU over the BERT embeddings, with a dropout layer and a dense softmax classifier. This combination serves as the minimal recurrent architecture. It allows temporal modeling of the input sequence while keeping BERT's semantic representation. It returns a strong baseline with accuracy = 0.47 and macro F1 = 0.42.

**Model1: BERT + Bidirectional GRU**

To enhance context capture, we replaced the GRU with a bidirectional GRU. This model could capture both past and future contexts in the embedded sequence. The output is pooled and passed through dropout and a linear classification.

**Model2 (Bi-GRU + Attention)**

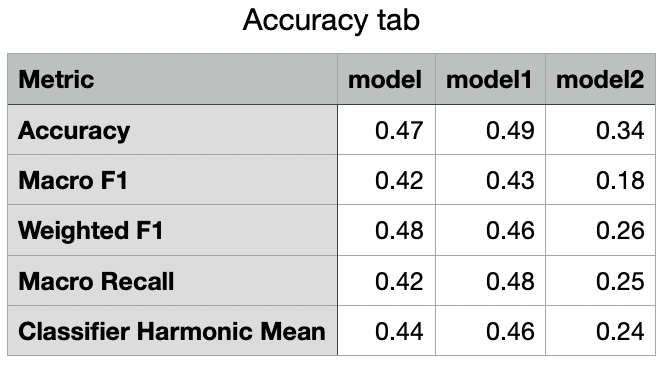
To investigate the effect of nuanced aggregation of evidence phrases, we add a custom attention mechanism over the BiGRU outputs. The attention scores weight the GRU outputs, enabling the model to selectively emphasize important semantic tokens when forming the final representation.

**Training Details**

All models were trained using the Adam optimizer. The cross-entropy loss was used for the 4-class classification. We applied early stop on the validation accuracy to prevent overfitting. We also fine-tuned all models for a minimum of 15 epochs using a GPU-enabled Colab environment.

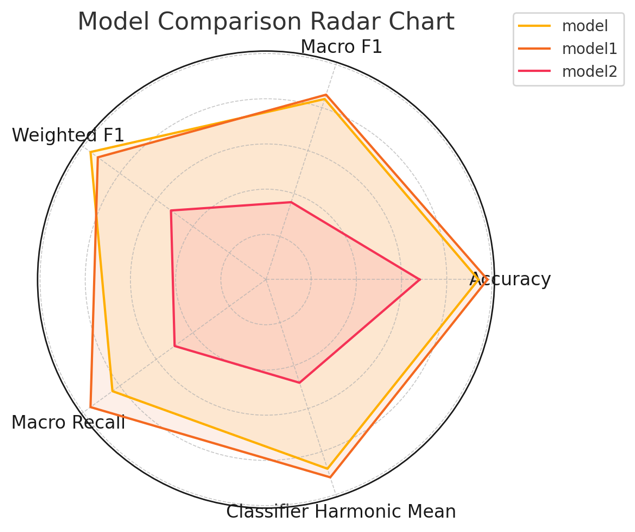
**Evaluation and Analysis**

We adopted standard evaluation metrics here:



**Figure 3.3.3-1** presents the performance radar plot comparing the three models. The base model performed reasonably well, validating the effectiveness of GRU. We found that Model 1 consistently outperformed others across all metrics. This indicate bidirectionality significantly improves understanding of claim-evidence relationships. In Model 2, despite aggregates attention, it still performed worst. This may be due to overfitting and instability in attention weights on the limited data size.

The error analysis from the confusion matrix (**Figure 3.3.3-2**) revealed that most misclassifications occurred in the ‘DISPUTED’ and ‘NOT\_ENOUGH\_INFO’ classes, we made a hypothesis against this problem: the model may have over-relied on clean, correctly retrieved evidence during training, resulting in poor generalization when faced with noisy or imperfectly retrieved evidence at inference time.



**Figure 3.3.3-1**

图表

描述已自动生成

**Figure 3.3.3-2**

**3.3.4 Sentence Transformer with MLP Classifier**

We also explored a non-recurrent architecture for claim classification, by using SentenceTransformers for semantic encoding of claim-evidence pairs with a multi-layer perceptron (MLP). This approach was chosen for its computational efficiency and its ability to model sentence-level similarity directly, without requiring recurrent or attention-based sequence modeling.

**Baseline Model: Sentence-BERT + MLP**

This pipeline uses a SentenceTransformer model (MiniLM-L6-v2) to encode each claim-evidence pair into a single dense embedding vector, and then passed to an MLP composed The resulting 384-dimensional vector is then passed through a multi-layer perceptron (MLP) with ReLU activation and dropout layer, followed by a softmax layer for 4-class classification.

**Architecture flow:**

* Input: [claim\_text] + [evidence\_text]
* Sentence-BERT encoding
* MLP: Linear 🡪 ReLU 🡪 Dropout 🡪 Linear 🡪 Softmax

**Improved Model: Claim-Only Training + Nearest-Neighbor Voting**

To reduce noise from irrelevant evidence, the improved model uses only claim text during training, which omites vidence entirely. This helps the model focus on linguistic patterns within claims and avoid overfitting to misleading evidence. At inference, we encode test claims using Sentence-BERT and retrieve the k most similar training claims (via cosine similarity). Final prediction is made through majority voting on their labels, leveraging semantic proximity for classification.

**Training Details**

We trained the model using categorical cross-entropy loss and Adam optimizer with a learning rate of 1 × 10-4, batch size of 32. And we used early stopping based on validation loss to mitigate overfitting. Training was conducted in under 10 minutes per fold on a standard Colab GPU.

**Evaluation and Analysis**

As visualized in the confusion matrix (**Figure 3.3.4-1**), the baseline model struggled heavily with **‘**SUPPORTS’ vs ‘NOT\_ENOUGH\_INFO’:

* Only 22 of 68 SUPPORTS instances were correctly classified.
* 46 SUPPORTS samples were misclassified as ‘NOT\_ENOUGH\_INFO’.

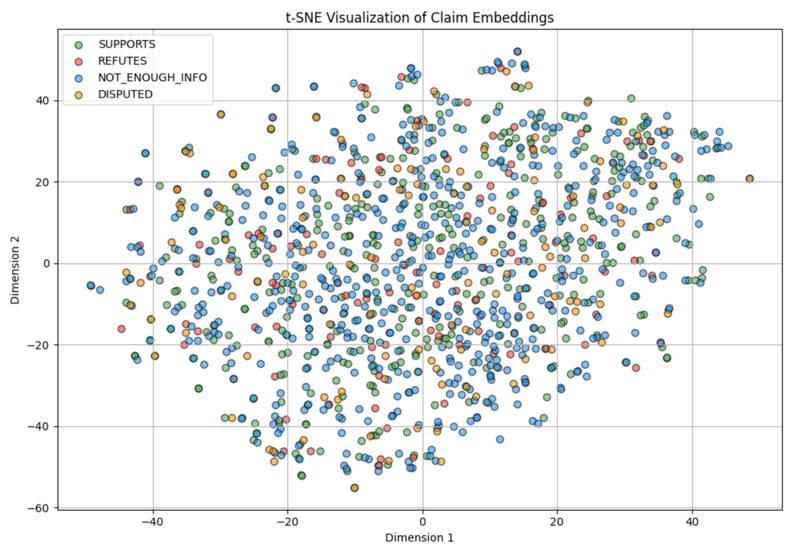
This indicates a bias toward predicting insufficient information when the semantic boundary is subtle. This behavior aligns with our hypothesis before for other types of classifiers. The dense sentence embeddings, while effective for semantic similarity, may lose fine-grained token-level signals needed for nuanced factual inference. Embeddings from different classes are not well-separated, particularly between ‘SUPPORTS’, ‘NOT\_ENOUGH\_INFO’, and ‘DISPUTED’. Class overlap suggests that the Sentence-BERT encoder does not distinctly capture class-specific features in this task context. Only claims with extremally confirmed evidence are classified as ‘SUPPORTS’.

图表

描述已自动生成

**Figure 3.3.4-1**

The improved model exhibited slightly more stable behavior and better class-specific precision in ‘REFUTES’ and ‘SUPPORTS’ compared to the baseline. From the t-SNE visualization (**Figure 3.3.4-2**) we can observed that, all points of 4 classes are heavily intermixed, so there’s no clear, class-specific clusters we expected. This means embeddings alone are not linearly separable for classification and semantic features are shared across classes, so simple models could be confusing on such datasets.



**Figure 3.3.4-2**

**3.3.4 Discussion**

One of the main challenges we identified during all classifiers was the model's over-reliance on clean and ideal evidence during training, which often led to overfitting and poor generalization in real-world settings where retrieved evidence may be irrelevant or misleading. To address this, we propose a general strategy: noisy evidence injection.

Most fact-checking systems assume that retrieved evidence during testing is of relatively same quality to training annotations. However, the truth is not so in practice. By injecting retrieval-like noise into training, we can better simulate real-world input conditions and teach the model to distinguish between truly informative and distractive evidence passages. For example, for each claim in the training set, we can increase the correct evidence with a few passages that seems incorrect (e.g., top-k retrieved items not labeled as ground truth). The ratio of correct to incorrect evidence (e.g., 1:2) can be tuned.