GROUP 12

Team Members: *Ben Baker and Ben Barrow*
Github: *https://github.com/BenTheBacker/NLU---Project*
Course: *COMP34812 Natural Language Understanding*

Evidence Detection in Claim-Evidence Pairs (Track B)

Using a BiLSTM (Non-Transformer) and a DeBERTa-v3 (Transformer) Model

REFERENCES & ACKNOWLEDGMENTS

1. *DeBERTa-v3: He et al., 2021. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention." arXiv:2111.09543*
2. *Hyperopt: Bergstra et al., 2013. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures."*
3. *GloVe Embeddings: Pennington et al., 2014. "Glove: Global Vectors for Word Representation."*
4. *NLTK: Bird et al., 2009. "Natural Language Toolkit."*

Acknowledgments:
Kaggle P100 GPU resources, Hugging Face Transformers library, and the authors of open-source frameworks.
Course staff for organizing the ED track dataset and helpful instructions.
Images Designed by Freepik

01. Introduction

Task:

We address the Evidence Detection (ED) shared task. Given a **claim** and a **piece of evidence**, the goal is to determine whether the evidence is **relevant** to the claim (label = 1) or **not relevant** (label = 0). This classification task has important real-world applications, such as **fact-checking**, **argument mining**, and **information retrieval**.

Data:

- ~24,800 claim-evidence pairs for training (plus our augmented samples).
- ~6,000 pairs for development/validation.
- A separate hidden **test set** (labels unavailable to participants until final evaluation).

Constraints:

- **Closed task:** No additional external datasets allowed.
- We **augmented** the training data using synonym replacement from WordNet.

02. Our Two Approaches

1) BiLSTM + GloVe (Category B: Non-Transformer Deep Learning)

Motivation:

- Traditional **RNN-based** architectures (LSTMs) still perform competitively on sequence classification tasks.
- We used **GloVe** word embeddings to capture pre-trained semantic knowledge.
- **Attention mechanism** aggregates contextual information across the entire sentence pair.

Key Features:

- **Embedding layer** initialized with **300-dimensional GloVe vectors**.
- **Bidirectional LSTM** (BiLSTM) to capture past and future context.
- **Attention layer** to focus on the most relevant tokens.
- **Optional Focal Loss** to mitigate class imbalance.
- **Hyperparameter search** via Hyperopt (TPE) for learning rate, batch size, epochs, focal gamma, label smoothing.

Data Augmentation:

- **Synonym replacement** with a 15% chance to duplicate and replace 1 random word.
- Helps introduce slight variability and combat overfitting.

Implementation Outline:

1. Tokenize claim + evidence using a **BERT-based tokenizer** (for consistency in indexing).
2. Align BERT tokenizer's vocab with GloVe embeddings (fallback to random vectors for missing tokens).
3. Feed the token IDs into BiLSTM; extract final hidden states or apply attention.
4. Classify with a simple **Linear layer** → 2 classes (0 or 1).
5. Evaluate on the dev set using **F1 (weighted)** and **Accuracy**.

2) DeBERTa-v3 Base (Category C: Transformer-Based Approach)

Motivation:

- **DeBERTa-v3** (by Microsoft) is known for strong performance on various NLP tasks.
- Self-attention with **Disentangled attention** and **Enhanced Mask Decoder**.
- Outperforms many standard BERT-family models on benchmarks.

Key Features:

- **Pre-trained** on large corpora; we fine-tune on our ED dataset.
- **Optional Focal Loss** or **Label Smoothing** to handle noisy labels and mitigate imbalance.
- **Hyperopt** to explore learning rate, epochs, batch size, etc.

Implementation Outline:

1. Use **microsoft/deberta-v3-base** from Hugging Face.
2. Tokenize (claim, evidence) with **AutoTokenizer**, max length = 128.
3. Replace the final layer's standard cross-entropy with either **Focal Loss** or **Label Smoothing**.
4. Train for 2–4 epochs, evaluate on dev.
5. Save the best model (lowest validation loss or highest F1).

03. Hyperparameter Tuning

We used **Hyperopt** with the **Tree-structured Parzen Estimator (TPE)** algorithm.

- **learning_rate ∈ log-uniform(1e-5, 5e-4)**
- **epochs ∈ {2, 3, 4}**
- **batch_size ∈ {4, 8, 16}**
- **focal_loss gamma ∈ [1.0, 5.0]**
- **label_smoothing ∈ [0.0, 0.2]**

Search Space

Objective: Maximize **weighted F1** on dev set.

04. Results & Evaluation

We report **weighted F1** (primary), **Accuracy**, and a **classification report** on the dev set.

Metric	Value
F1 (weighted)	**0.8235**
Accuracy	**0.8272**
Loss (Val)	**0.2086**

Classification Report

- **Class 0**: precision=0.8634, recall=0.9041, f1=0.8833
- **Class 1**: precision=0.7142, recall=0.6262, f1=0.6673

Key Observations

- Attention mechanism boosted recall slightly for the **relevant** class.
- Focal loss or label smoothing helped handle the imbalance.

BiLSTM + GloVe

Metric	Value
F1 (weighted)	**0.8894**
Accuracy	**0.8873**
Loss (Val)	**0.1149**

Classification Report

- **Class 0**: precision=0.9458, recall=0.8955, f1=0.9199
- **Class 1**: precision=0.7602, recall=0.8659, f1=0.8096

Key Observations

- The transformer-based approach yields higher overall F1.
- Adding **Focal Loss** or slight **Label Smoothing** further improves performance.

DeBERTa-v3

05. Discussion

Comparing Solutions

- ***BiLSTM*** *solution is simpler, smaller (~270MB), and trains faster in practice.*
- ***DeBERTa-v3*** *is more powerful, achieving ~0.8894 F1 on dev vs. ~0.8235 for BiLSTM.*

Data Augmentation

- *Synonym replacement introduced variety but occasionally produced awkward phrases.*
- *Potential next step: more sophisticated augmentation (e.g., back-translation).*

Limitations

- ***Domain shift:*** *The model may perform poorly on specialized or domain-specific claims.*
- ***Vocabulary mismatch:*** *GloVe or subword tokens may not capture niche terms.*
- *Focal Loss requires* ***tuning gamma*** *carefully to avoid over-emphasizing minority classes.*

06. Conclusion & Future Work

Both solutions effectively address Evidence Detection; the DeBERTa-v3 approach outperforms the BiLSTM on dev. However, BiLSTM remains competitive if compute resources are limited.

Future Work:

- Investigate advanced data augmentation (e.g., mask-based or back-translation).
- Try domain adaptation for specialized datasets.
- Explore post-hoc explainability methods (e.g., attention visualization, LIME, or SHAP).