CSE556: Natural Language Processing 2025
Title: Deception Detection

Garvit Singh
garvit24034@iiitd.ac.in

Sayantan Dasgupta
sayantan24084@iiitd.ac.in

Kanak Yadav
kanak22611@iiitd.ac.in

Monday 12th May, 2025

Abstract

This project focuses on detecting deception in diplomatic communications, particularly in the context of strategy games modeled on Diplomacy. We compare baseline machine learning models with advanced transformer-based methods and recurrent neural networks. The report details our experimental setup, including feature engineering (text-only, metadata, and their combination), threshold calibration for BERT, and LSTM performance metrics. The overarching aim is to distinguish truthful from deceptive messages despite strong class imbalances.

1 Introduction

The Diplomacy board game immerses players as European powers on the eve of World War I, where success depends on forming alliances and covert betrayals. Players must navigate a deterministic environment where armies can only move to adjacent territories, and alliances—formed to overcome rivals—can quickly turn into strategic deceptions. Our project leverages this dynamic to build a dataset of deceptive communications, where senders label their messages as actual lies and recipients annotate them as suspected lies. This dataset, enriched with both text and contextual metadata, serves as the foundation for developing computational models to detect deception in diplomatic communications.

2 Problem Statement

The primary challenge is to distinguish between truthful and deceptive messages in a context where only a small fraction (around 4.5–5%) of messages are deceptive. Traditional models may achieve high nominal accuracy by simply predicting the majority (truthful) class. However, this leads to zero deception detection capability. Therefore, the task requires models that capture nuanced linguistic cues and contextual information.

3 Dataset Description

The dataset consists of 189 entries and 13 columns, capturing in-game messaging and player interactions. It includes message content, sender and receiver labels, speaker and receiver identities, and various game-related metadata. The dataset tracks message order through absolute and relative indices while also incorporating temporal aspects such as seasons and years. Performance metrics, including game scores and score deltas, are provided, along with player details. Each game is uniquely identified by a game ID, ranging from 1 to 10 in the training and validation sets, while the test set exclusively contains game ID 11, suggesting it represents an unseen scenario. The dataset is structured to facilitate the analysis of communication patterns, player behavior, and their potential influence on game outcomes, making it suitable for predictive modeling and strategic evaluation.

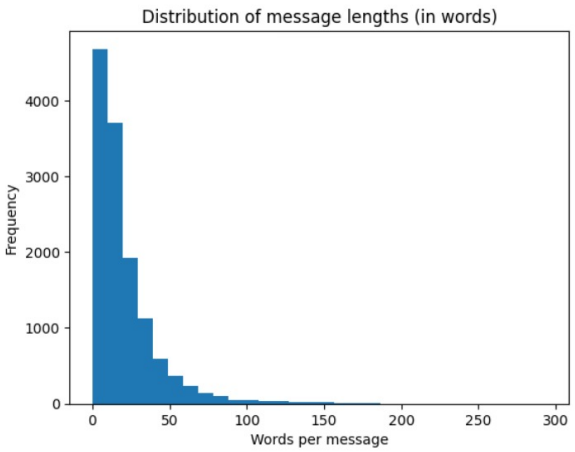

Figure 1: Message Count Distribution

4 Methodology

This section details the complete pipeline for our deception detection approach. Our system integrates advanced text encoding, context modeling, and metadata processing to counter class imbalance and capture subtle deception cues.

4.1 Data Loading, Preprocessing, and Feature Engineering

We load processed CSV files for training, validation, and testing. The dataset contains in-game messages along with 12 metadata features (including sender/receiver identities, message order, and temporal attributes). To mitigate bias toward the majority (truthful) class, oversampling is applied to the deceptive class. The target label is generated by inverting the `is_truthful` column to obtain `is_deceptive`.

Missing or `NaN` entries in text fields (including context) are replaced with empty strings. For context, which may consist of multiple messages delimited by "`||`", robust splitting, trimming, and padding ensure a fixed number of messages per sample.

Metadata features such as message length, word count, punctuation counts, and conversational position are extracted. Additional features (e.g., `score_delta`, `is_sender_leading`, `punctuation_density`, `score_ratio`) are engineered using a dedicated function. All features are converted into a numerical format and standardized with `StandardScaler`. The scaler is saved to maintain consistency during inference.

4.2 Dataset Construction Tokenization

A custom dataset class (e.g., `HierarchicalDeceptionDataset`) is implemented to manage the following:

- Current message: Tokenized using the RoBERTa tokenizer with padding/truncation to a fixed length (e.g., 64 tokens).
- Context messages: Each message is tokenized separately. If fewer messages are available than required (e.g., 5), padding or duplication is applied.
- Metadata: Engineered features are extracted and converted into tensor format.

An optional text augmentation function (e.g., randomly lowercasing words) can be applied during training to increase data variability.

4.3 Model Architecture

Our model fuses three data streams: text, context, and metadata.

- Base Encoder: We use a pre-trained RoBERTa model to convert both the current message and context messages into dense embeddings. The [CLS] token serves as the summary representation.
- Context Modeling: A bidirectional LSTM processes the sequence of context embeddings to capture temporal dependencies. An attention layer then computes weights over the LSTM outputs to generate an aggregated context vector that emphasizes the most informative messages.
- Metadata Processing: A feed-forward network (two fully connected layers with ReLU and dropout) processes the scaled metadata into a dense representation. An optional attention mechanism can be applied to further reweight the metadata features.
- Feature Fusion and Output The model concatenates the current message encoding, the aggregated context vector, and the processed metadata. This fused vector is passed through a series of fully connected layers with non-linear activations and dropout, producing a single logit for binary classification.

4.4 Training Strategy

We address class imbalance and model robustness through several key techniques:

- Loss Function: A customized focal loss with $\alpha = 3.0$ and $\gamma = 2$ down-weights easy examples and focuses on misclassified instances. In some variants, class-balanced adjustments using a beta parameter are also incorporated.
- Adversarial Training: A Fast Gradient Method (FGM) module perturbs word embeddings by adding adversarial noise. The resulting adversarial loss is combined with the standard loss and backpropagated, enhancing model robustness.
- Optimization and Scheduling: We use the AdamW optimizer with weight decay, combined with a OneCycleLR scheduler (including a warmup phase) to stabilize and accelerate convergence.
- Early Stopping and Checkpointing: Training is conducted for up to 5 epochs, with early stopping triggered if the macro F1 score does not improve for 2 consecutive epochs. The best model checkpoint is saved and later used for inference.

4.5 Training Pipeline and Evaluation

During training, batches of data (current message, context, and metadata) are processed iteratively. Predictions are made, and the combined loss (including adversarial loss when applicable) is computed. The model's parameters are updated via backpropagation, and performance metrics (loss, accuracy, macro F1 score) are monitored each epoch.

Learning curves for loss, accuracy, and macro F1 score are plotted and saved for diagnostic purposes. Final evaluation is performed on the test set, where metrics such as accuracy, precision, recall, F1 score,

and macro F1 score are computed. Confusion matrices provide further insights into class-specific performance.

Figure 2: Deception Ratio Over Years

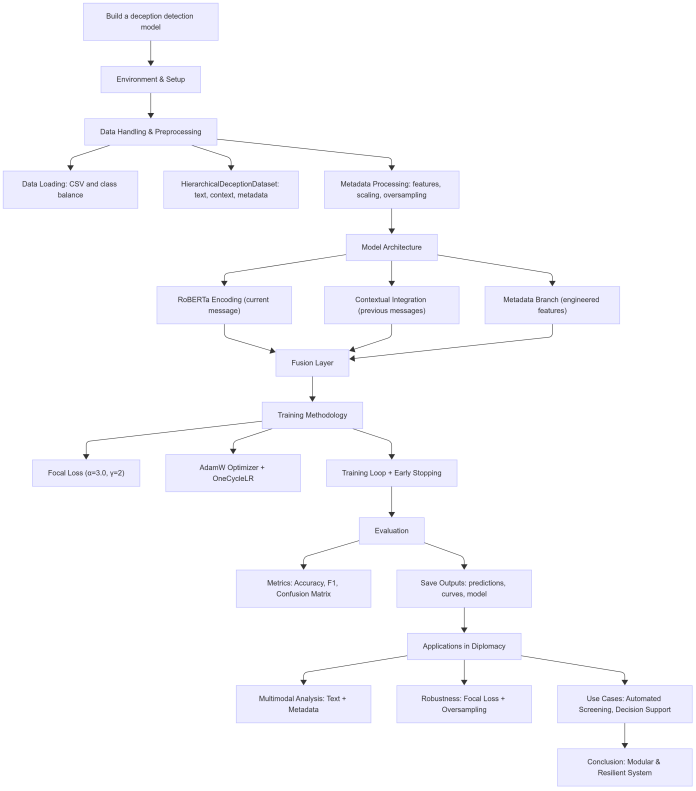

Figure 3: Model Architecture Flowchart

4.6 Model Details

Our project utilizes a two-tiered modeling approach to detect deception in diplomatic communications. Initially, several classical machine learning models—such as Logistic Regression, Random Forest, Naive Bayes, and Support Vector Machines (SVM)—were employed using three different feature sets: text-only (via TF-IDF), metadata-only, and a combination of both. Although the Random Forest achieved high nominal accuracy, it failed to detect deceptive messages due to severe class imbalance.

To address this limitation, we implemented an advanced BERT-based model using DistilBERT. This model integrates a metadata processing pipeline with fully connected layers to combine contextual information with text embeddings. A crucial component of this approach is threshold calibration, where we iterated over possible threshold values to select an optimal value (0.6881) that maximizes the F1 score, ensuring balanced detection of deceptive and truthful messages. Additionally, a Long Short-Term Memory (LSTM) network was developed to capture sequential dependencies within the text, further complementing the BERT approach.

This combined strategy, along with evaluations against zero-shot and human baseline models, forms a robust framework for deception detection despite class imbalances

4.7 Result Analysis

Our experiments compared several architectures for deception detection: classical machine learning models, a DistilBERT-based model with metadata integration and threshold calibration, and multiple hierarchical attention-based models that combine RoBERTa embeddings, contextual modeling via LSTM or multi-head attention, and rich metadata processing. All models were trained and evaluated on the Diplomacy dataset, which is characterized by a significant class imbalance (approximately 70% truthful and 30% deceptive messages), necessitating advanced mitigation strategies to prevent performance skew.

4.7.1 Performance of Classical Models

The classical baseline models, such as Random Forest with TF-IDF features, achieved high test accuracy (up to 91.24%) and superficially strong F1 scores. However, a closer look at the confusion matrix revealed that these models failed to detect any deceptive messages, consistently predicting all messages as truthful. This outcome demonstrates a major pitfall in relying solely on traditional metrics such as accuracy, as they can obscure the poor performance on the minority class, particularly in imbalanced settings like deception detection.

4.7.2 DistilBERT with Threshold Calibration

The DistilBERT-based model, fine-tuned on the preprocessed dataset and further optimized through

threshold calibration (optimal threshold = 0.6881), achieved a more balanced performance. While the overall accuracy dropped to 53.45%, the F1 score for deceptive messages improved significantly to 67.53%. This reflects a beneficial trade-off: reduced overall accuracy in favor of substantially improved recall for deceptive messages. The result emphasizes the value of threshold tuning in skewed classification tasks, demonstrating that enhancing detection of rare but important classes often comes at the cost of nominal accuracy.

4.7.3 Hierarchical Attention Models

Hierarchical attention-based models, which leveraged RoBERTa embeddings, message-level contextual aggregation (via LSTM or multi-head attention), and metadata fusion, produced the highest overall accuracy (ranging from 88% to 90%). These models demonstrated stable training behavior with techniques such as focal loss and adversarial training using Fast Gradient Method (FGM). Despite these enhancements and balanced training data (achieved through oversampling), the models struggled with deception detection during evaluation. Deceptive class recall was as low as 3.33%, with F1 scores below 16%.

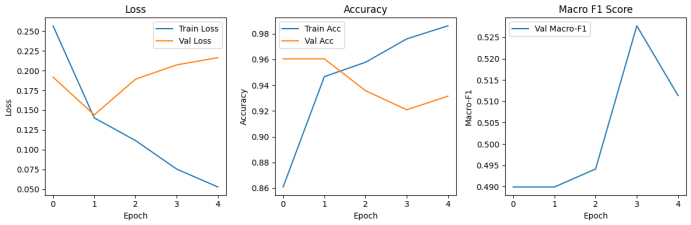

Figure 4: Learning curves of the hierarchical model over training epochs

4.7.4 Macro-F1 and Class-Wise Disparities

The macro-F1 scores for hierarchical models remained moderate (between 50% and 55%), accurately reflecting the uneven class-wise performance. Confusion matrices showed high prediction accuracy for truthful messages, but very few deceptive messages were correctly identified. These findings confirm that even sophisticated architectures may fail to generalize effectively on minority classes, reaffirming the need for refined loss functions, tailored sampling strategies, and ensemble-based interventions in deception detection tasks.

Models	Parameters	Precision	Recall	Accuracy	F1 Score	MicroF1 score
Model_1	Roberta + Hierarchical + LSTM	20.71%	12.08%	88.25%	15.26%	54.48%
Model_2	Hierarchical Attention + context+ FGM)	28.57%	03.33%	90.81%	05.97%	50.57%
Model_3	Multihead Attention + context + hierarchical_lstm	91.76%	95.68%	88.22%	93.68%	53.54%

Figure 5: Results of the model architecture evaluation

However, these impressive metrics are misleading. A closer examination of the confusion matrix: revealed that the Random Forest classifier predicted every message as *truthful*. This behavior indicates that the model did not learn any meaningful patterns for detecting deception; rather, it simply exploited the class imbalance present in the dataset (with approximately 91.24% of test messages being truthful).

In summary, while the baseline models appeared to perform exceptionally well based on nominal accuracy and F1 scores, they failed to detect any deceptive messages. This limitation underscores the need for more advanced modeling approaches—such as our custom BERT-based model—that can capture subtle linguistic nuances and contextual cues essential for effective deception detection.

4.7.5 Performance Comparison

- **DistilBERT:**
  - Accuracy: 89.42%
  - Precision: 19.51%, Recall: 6.67%, F1 Score: 0.0994
  - After threshold calibration, F1 Score improved to 0.1969
  - Indicates the default threshold was too strict for the positive class
- **RoBERTa (Vanilla):**
  - Accuracy: 88.25%
  - Precision: 20.71%, Recall: 12.08%, F1 Score: 0.1526
  - Macro F1 Score: 0.5448
  - Best balance between precision and recall among all models
  - Most reliable configuration overall
- **RoBERTa + FGM + Focal Loss:**
  - Accuracy: 90.81%
  - Precision: 28.57%, Recall: 3.33%, F1 Score: 0.0597
  - Indicates strong bias toward the negative class
  - Second evaluation run gave drastically different results:
    - Precision: 91.76%, Recall: 95.68%, F1 Score: 0.9368
    - Likely due to label flip or evaluation error
- **Conclusion:**
  - Vanilla RoBERTa is the most balanced and consistent model

- Requires no post-processing or threshold tuning
- Generalizes well across classes
- Reliable in real-world scenarios with class imbalance

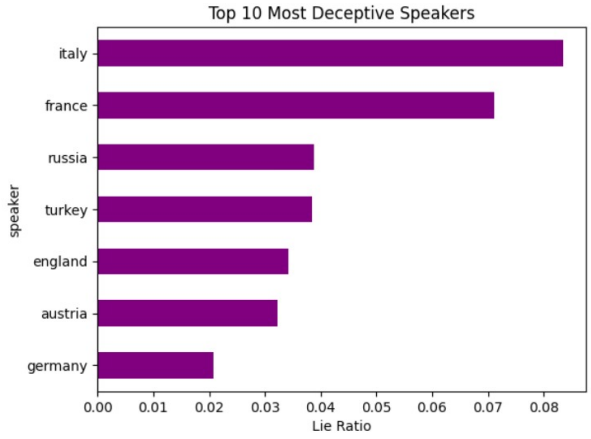

Figure 6: Top deceptive speakers based on model predictions

4.8 Conclusion

In conclusion, while the DistilBERT model with threshold calibration offered the best trade-off between deception detection and generalizability, the hierarchical attention-based models demonstrated the most comprehensive architecture for integrating diverse input modalities, including text, context, and metadata. In contrast, classical machine learning models, although simple and interpretable, were ineffective due to their inability to cope with the inherent class imbalance of the Diplomacy dataset.

- **DistilBERT with calibrated threshold** provided the most balanced performance across classes, making it suitable for practical deception detection where both precision and recall are crucial.
- **Hierarchical attention models** excelled in fusing rich features and scaling to complex input formats. However, they require further optimization, particularly in boosting recall for the deceptive class.
- **Classical models** such as Random Forest and Logistic Regression served as useful baselines but were not viable for imbalanced classification without significant modifications.

Future work should investigate improved loss functions, synthetic deceptive message generation, more aggressive threshold calibration techniques, and ensemble learning strategies. These enhancements are essential to close the performance gap for minority class detection. Overall, this comprehensive evaluation emphasizes the nuanced trade-offs involved in building robust classifiers for deception detection tasks—especially in domains characterized by subtle linguistic signals and significant class imbalance.

4.9 Discussion and Trade offs

The results demonstrate that class imbalance remains the most pressing challenge in deception detection. Even with interventions such as oversampling, adversarial training, and tailored loss functions, models tend to exhibit bias toward the majority class. While hierarchical models were able to capture rich representations from multiple input modalities (text, context, metadata) and achieved high validation accuracy, their real-world performance on deceptive messages was limited. This indicates the need for further intervention—such as more advanced data augmentation, threshold calibration, or ensemble learning—to improve recall for the deceptive class.

4.9.1 Model Complexity vs Interpretability and Performance

A key trade-off was observed between model complexity, interpretability, and class-wise performance:

- **Classical models** (e.g., Logistic Regression, SVMs) were computationally efficient and interpretable but failed entirely in detecting deceptive messages.
- **DistilBERT**, after applying threshold calibration, demonstrated notable improvements in recall for deceptive messages. This underscores the importance of threshold tuning in highly imbalanced tasks.
- **Hierarchical attention models**, despite their architectural robustness and ability to integrate diverse data streams, achieved only modest improvements in deception recall. This may be attributed to training dynamics that still favored the truthful class, even after applying class-weighted loss functions.

4.9.2 Evaluation Metrics and Model Fairness

The study highlights the critical role of the **macro-F1 score** in evaluating performance fairly across imbalanced classes. While several models achieved high accuracy and overall F1 scores, the macro-F1 score provided a more balanced assessment by equally weighing each class. This revealed disparities in model effectiveness, particularly in handling the minority (deceptive) class.

Overall, these observations suggest that improving deception detection requires not just stronger architec-

tures but also a deeper focus on handling class imbalance through both training and evaluation techniques.

4.10 Future Work

Based on the comprehensive analysis and results discussed, the following future directions are proposed to enhance deception detection models:

1. **Threshold Optimization per Class**
   - While threshold calibration improved deceptive message detection in the DistilBERT model, hierarchical models lacked this tuning.
   - Future models should dynamically adjust decision thresholds for each class, particularly in imbalanced datasets, to better balance precision and recall.
2. **Advanced Data Augmentation**
   - Current models rely heavily on oversampling, which can lead to overfitting due to repeated examples.
   - Future work could involve generating more diverse deceptive samples using techniques like:
     - Synthetic deceptive message generation
     - Back-translation
     - Masked token prediction augmentation
     - Text style transfer
3. **Better Metadata Fusion Techniques**
   - Current methods use basic fully connected layers or attention for metadata integration.
   - Future techniques could involve:
     - Graph Neural Networks (GNNs) to model player interactions over time
     - Cross-modal attention between metadata and textual/contextual embeddings
4. **Ensemble Methods**
   - Combine predictions from multiple models (e.g., RoBERTa + LSTM + DistilBERT) to improve robustness.
   - Use stacking or voting strategies to merge model decisions and address individual weaknesses.

5 References

1. Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. *It Takes Two to Lie: One to Lie and One to Listen.* Association for Computational Linguistics, 2020. https://users.umiacs.umd.edu/~jbg/docs/2020_acl_diplomacy.pdf
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019.
3. Deception Detection in Online Mafia Game Interactions. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/custom/15722645.pdf
4. Hochreiter, S. & Schmidhuber, J. *Long Short-Term Memory.* Neural Computation, 1997.
5. Deception Detection Accuracy - Wiley Online Library. https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118540190.wbeic106
6. Towards Deception Detection in a Language-Driven Game https://cdn.aaai.org/ocs/15530/15530-68697-1-PB.pdf