Deception Detection: QANTA Diplomacy

Aditya Sahai (MT24009), Shreyas Rajendra Gore (MT24087), Sharad Jain (MT24138)

Abstract

The Deception Detection project also known as QANTA Diplomacy project is a classification task where in-game messages that are shared between players in the game are to be identified as either deceptive or truthful. Goal is to develop model that uses conversational text and metadata to predict deception in the message, with dataset of 17,289 messages. The model is then evaluated using accuracy. It has many implications in the field of NLP, decision making tasks and game theory.Deception Detection also had wide range of applications in field of security in these online era.

1 Problem Definition

The project task is to develop model that detects deceptive messages in communication between players. The task involves analyzing text and metadata to classify a given message is deceptive or truthful messages.

2 Introduction

Deception detection in Diplomacy involves recognizing manipulative or untruthful language used by players to gain strategic advantages. Unlike conventional NLP tasks, deception detection requires modeling speaker behavior, power dynamics, and message history. In this paper, we evaluate four diverse modeling strategies and demonstrate the superiority of a power-aware GNN-based model.

3 Literature Review

Peskov et al. (1) Introduced one of the first datasets targeting deception in a multi-agent, long-form dialogue setting using the game of Diplomacy. As per this paper we found that deception is often context-dependent, influenced not only by linguistic styles but also by the relational dynemics between players, such as prior trust, power imbalances, and conversational history.While their models explored logistic regression and LSTM-based approaches, they noted that deception frequently spans multiple turns and involves dynamic social interactions. Motivated by these findings, we extend their work by incorporating graph-based relational models, which allow us to explicitly encode and reason over speaker-message relations and game structure. Graph neural networks (GNNs), particularly attention-based ones, are well-suited to model such interactions by capturing both message-level semantics and the underlying communication topology.

4 Dataset

We use the Diplomacy Deception Dataset introduced by Peskov et al. (1) which contains 17,289 in-game messages exchanged between players in the strategy game Diplomacy. Each message is annotated by the sender as either truthful or deceptive, and by the receiver for perceived deception, forming a rich supervision setting that captures strategic behavior in long-term alliances and rivalries.

4.1 Data Structure

The dataset includes message-level, speaker-level, and conversation-level metadata. Table 1 summarizes the most relevant fields used in our models.

Field	Description
`message`	Raw text of the in-game message
`speaker` / `receiver`	Countries involved (e.g., England, Turkey)
`sender_labels`	Ground truth label: `true` (truthful) or `false` (deceptive)
`receiver_labels`	Perceived label: `true`, `false`, or `NOANNOTATION`
`game_score`	Current supply center count for the sender
`score_delta`	Sender score minus receiver score
`absolute_message_index`	Index of message in full game timeline
`relative_message_index`	Index of message in the conversation
`season`, `year`	Temporal context (e.g., Fall 1903)

Table 1: Summary of key dataset fields

4.2 Example Message

An example of a message taken from dataset given to us and its associated metadata is shown below:

```
Message: "Let's work together to attack
France — I won't move into your terri-
tory."
Speaker: Germany
Receiver: England
Year: 1901, Season: Spring
Game Score: 3, Score Delta: +1
Sender Label: false (deceptive)
Receiver Label: true (truthful)
Deception Quadrant: Caught
```

This highlights the strategic nature of deception: although the message was deceptive, it was believed to be truthful by the receiver.

4.3 Class Imbalance

The dataset given to us has a significant class imbalance, with truthful messages being much more frequent than deceptive ones. Specifically, deceptive messages constitute approximately only around 5% of the labeled examples. This imbalance makes deception recall a challenging metric and motivates our use of:

- Data augmentation (synonym replacement on deceptive class)
- Weighted loss functions (focal loss or class-weighted cross-entropy)
- Balanced sampling strategies in training

Our models aim to improve deceptive F1 score, not just overall accuracy, to ensure real deceptive messages are detected reliably.

5 Methodology

We experimented with four modeling strategies of increasing complexity and incorporated structured metadata, augmentation, and oversampling techniques to improve deceptive message recall.

5.1 BiLSTM + Attention

This simpler model with a learned embedding layer and only uses the BiLSTM-attention mechanism along with structured metadata. It provides a strong baseline for understanding how much metadata alone can contribute without pretrained contextual embeddings.

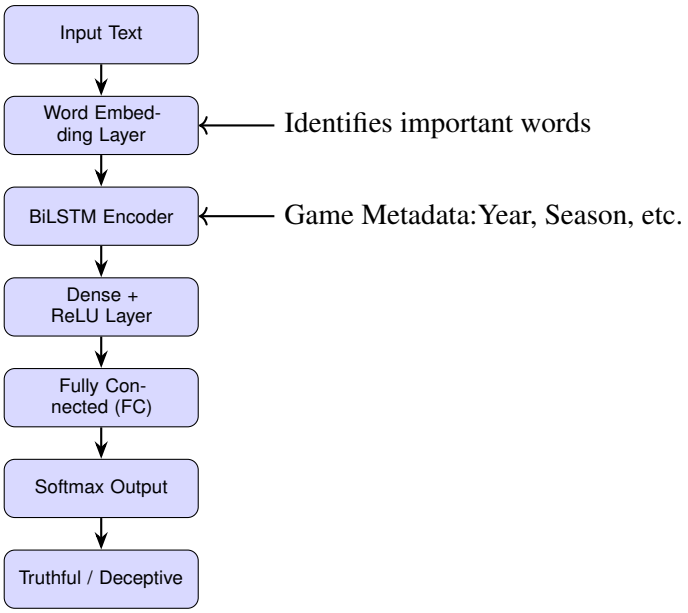

Figure 1: BiLSTM-Based Model Architecture

5.2 BiLSTM + Power + RoBERTa

Tokenized messages are processed through a BiLSTM with attention. Simultaneously, frozen RoBERTa embeddings (CLS token) are extracted for each message. These are combined with metadata features including the season (one-hot encoded), game score delta, message length, year, and message index. The fused representation is passed through a multi-layer perceptron (MLP) for final classification.

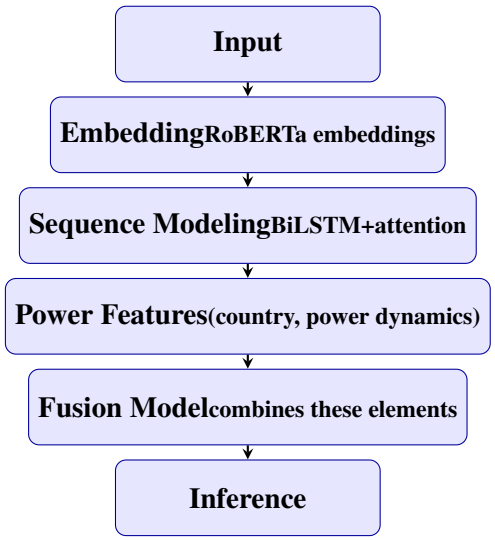

Figure 2: Flowchart of BiLSTM + RoBERTa model with metadata fusion.

5.3 LLM2Vec + GNN

In this approach, we model dialogue messages and their interactions as a heterogeneous graph to detect deceptive communication. Each messagees is represented using a fixed high-dimensional embedding vector precomputed externally (e.g., using DistilBERT or another transformer model). These embeddings are not updated during training.

- **Message Representation:** Each message is represented by a 2048-dimensional fixed embedding vector, augmented with six handcrafted metadata features: absolute and relative message indices, year, score, score delta, and message length. These features are normalized using MinMax scaling and concatenated to the message embeddings.
- **Player Encoding:** Players (speakers and receivers) are one-hot encoded and padded to match the message feature dimension. Each player node is projected through an MLP into the same space as the message embeddings.
- **Graph Construction:** The heterogeneous graph consists of:
  - **Message-to-Message Edges:** Temporal connections between consecutive messages.
  - **Speaker-to-Message and Receiver-to-Message Edges:** Capture communication patterns and roles of players in the conversation.
- **Feature Projection:** Both message and player embeddings are projected into a shared 512-dimensionals space using a two-layer MLP with ReLU activation.
- **Graph Neural Network:** Two GATConv layers are applied to the fused embeddings to capture contextual and structural dependencies within the message-player graph.
- **Classification Head:** The output node features are passed through a final MLP to perform binary classification (deceptive or truthful).
- **Training Objective:** The model is trained using `BCEWithLogitsLoss` with class weighting to handle label imbalance. Evaluation is done using Macro F1, accuracy, and deceptive-class F1 scores.

This setup allows modeling of both message-level sementic features and graph-based speaker-receiver interactions, improving performance on minority class detection.

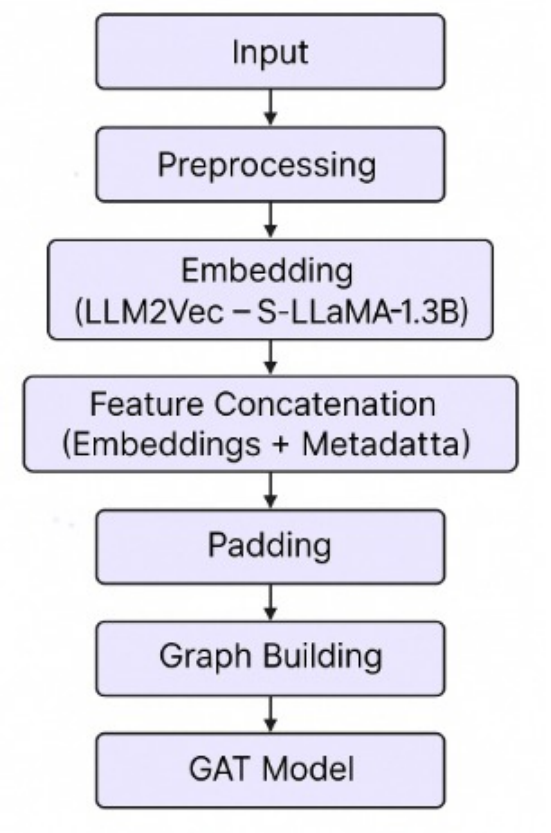

Figure 3: Model Architecture of LLM2Vec with GNN

5.4 MLDM- Final Model Architecture

Our best-performing model combines DistilBERT CLS embeddings i.e Multi Level Deception Model with:

- Dialogue act predictions (from a learned linear head)
- Power difference embeddings (from score delta)
- Graph encoding using GATs across speaker-message history

The fused representations are passed through an MLP classifier.

Metadata Integration

All models (except GNN-only variants) utilize structured metadata to supplement text representations. These include:

- **Power difference** (score_delta)
- **Game year and season**
- **Message length and position** (absolute/relative index)
- **Speaker and receiver identity** (encoded where applicable)

Augmentation and Oversampling

To address class imbalance, we used data augmentation and oversampling techniques:

- We applied synonym replacement using the `nlpaug` library (`wordnet` source) on deceptive class messages.
- Each deceptive message was augmented once, and the original + augmented examples were oversampled to match the number of truthful messages.
- The final training set was a balanced combination of truthful, deceptive, and augmented deceptive messages, randomized before training.

This strategy improved model stability and boosted F1 performance for the minority (deceptive) class.

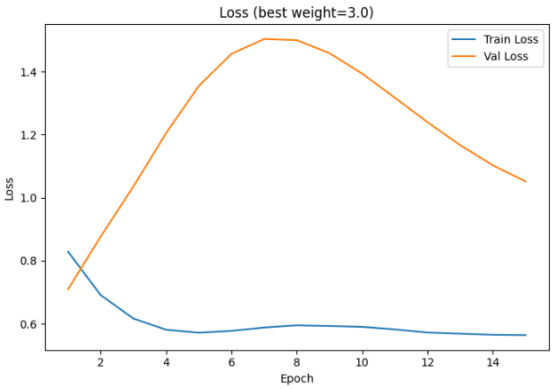

Figure 4: Training and validation loss/F1 curves over epochs.

6 Experimental Setup Issues And Resolvement

All experiments were conducted in Kaggle notebooks using the standard GPU environment (Tesla T4, 16GB RAM, 2 CPUs). Due to session resets in Kaggle, certain libraries had to be reinstalled at the start of each runtime:

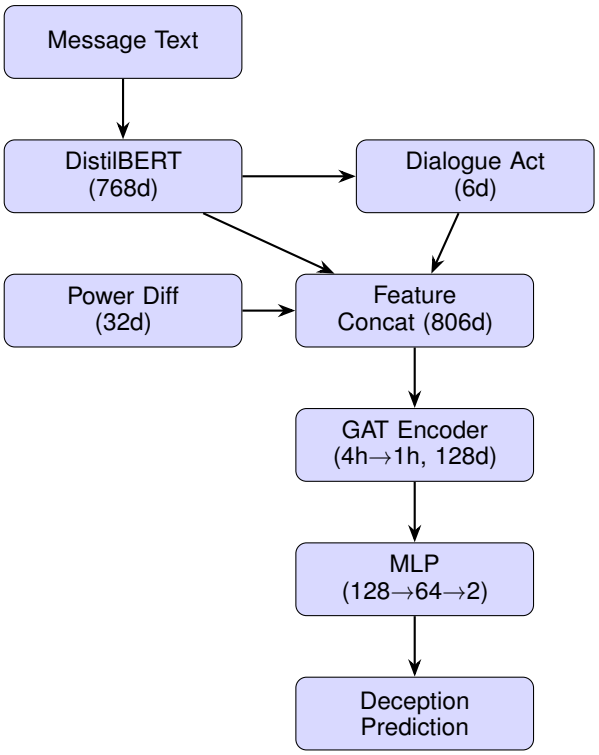

Figure 5: MLDM architecture: A graph-based deception detection model combining BERT embeddings, dialogue acts, and power difference features processed through GAT layers.

- `torch-geometric` for graph neural network components
- `nlpaug` for synonym-based data augmentation

Our approach relies heavily on rich sentence-level representations using transformer models. Initially, we experimented with full BERT-based encoders; however, this led to frequent `CUDA out of memory (OOM)` errors on the GPU due to the size of the model and batch processing requirements.

To address this, we adopted the **LLM2Vec** strategy:

- DistilBERT CLS embeddings were generated on the **CPU** in batches using inference mode.
- These embeddings were then saved to disk and reused across model training runs to avoid re-computation.

This embedding caching technique significantly reduced GPU memory usage and allowed us to train complex graph-based models (such as GAT) efficiently. By separating embedding extraction from model training, we were able to scale experimentation and avoid reprocessing raw text repeatedly.

Furthermore, speaker and receiver identities were encoded as one-hot vectors and projected using an MLP to align their dimension with message embeddings. All message and player vectors were then combined to construct a heterogeneous graph. The final training pipeline included graph construction, embedding fusion, and graph encoding using stacked GAT layers.

7 Evaluation Metrics

We evaluate the performance of each model using the following metrics:

- **Accuracy:** Measures the overall correctness of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Macro F1 Score:** The unweighted average of F1 scores across all classes, giving equal importance to both classes regardless of imbalance.

$$MacroF1 = \frac{1}{C}\sum_{i=1}^{C} F1_i, \quad whereF1_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \quad (2)$$

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

- **F1 Score for Deceptive Class:** Specifically evaluates performance on the deceptive class (positive class).

- **F_β Score (with $\beta = 1.5$):** A variant of F1 that emphasizes recall more than precision.

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \quad (4)$$

These metrics are selected to provide a balanced evaluation across all classes, with special attention to the minority class (deceptive messages), where traditional accuracy may be misleading.

8 Results and Analysis

Model	Macro F1	Accuracy
BiLSTM + Attention	0.47	0.67
BiLSTM + RoBERTa	0.49	0.68
LLM2Vec + GNN	0.53	0.81
MLDM	**0.54**	**0.83**

Table 2: Model performance comparison across standard metrics.

Comparative Analysis

Table 8 presents the comparative results of four models with increasing architectural complexity. We observe a clear trend where performance improves with the integration of pretrained embeddings, graph modeling, and metadata.

- **BiLSTM + Attention** establishes a neural baseline using learned word embeddings and sequential modeling. It achieves modest performance with a Macro F1 of 0.48 and 67% accuracy. The model struggles with the minority deceptive class due to lack of contextual or structural awareness.

- **BiLSTM + RoBERTa** introduces contextualized token embeddings from the `roberta-base` model. This leads to marginal gains (Macro F1 of 0.49 and accuracy of 68%), suggesting pretrained language features are beneficial but insufficient alone.

- **LLM2Vec + GNN** leverages frozen DistilBERT CLS embeddings combined with metadata and a graph neural network. It models message-player relations and improves performance significantly—Macro F1 rises to 0.53, and accuracy reaches 81%.

- **MLDM (Final Model)** Multi Level Deception Model integrates multiple components: BERT-based embeddings, dialogue act heads, power difference encodings, and speaker-message graph structure processed through GAT layers. This fusion leads to the best performance, achieving 0.54 Macro F1 and 83% accuracy.

The results demonstrate that integrating pretrained language models with relational graph modeling and metadata fusion yields robust performance for deception detection. The MLDM model particularly excels in recognizing deceptive cues in low-resource, imbalanced data scenarios.

9 Conclusion

In this work, we presented and compared several architectures for deception detection in Diplomacy messages. Our key findings are as follows:

- Sequential models like BiLSTM provide a strong baseline but struggle to capture nuanced social cues without contextual embeddings.

- Incorporating pretrained embeddings (e.g., RoBERTa and DistilBERT) improves performance, especially when fused with structured metadata.

- Graph-based architectures outperform sequential baselines by modeling message-message and player-message relationships.

- Our MLDM model, which fuses DistilBERT embeddings, dialogue acts, and power-aware GNN encodings, achieves the best overall results.

- Data augmentation, class balancing, and threshold tuning further improve performance on the underrepresented deceptive class.

These results underscore the importance of combining linguistic, structural, and strategic context for robust deception detection.

10 Future Work

While our proposed MLDM model demonstrates strong performance in deception detection within the Diplomacy domain, some cases remain open for future research and development like:

- **Dynamic Graph Modeling:** Currently, message graphs are constructed statically based on speaker interactions. Future work could explore dynamic graph construction using temporal graph neural networks to better capture evolving alliances and betrayals.

- **Expanded Metadata Usage:** Additional metadata such as action orders, territory movements, and alliance formations could be incorporated to provide richer contextual grounding for deception cues.

References

[1] D. Peskov, B. Cheng, A. Elgohary, and J. Barrow, "It takes two to lie: One to lie, and one to listen," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Online, 2020, pp. 3811–3854. doi: 10.18653/v1/2020.acl-main.353.

[2] A. S. Constâncio, D. F. Tsunoda, H. F. N. Silva, J. M. da Silveira, and D. R. Carvalho, "Deception detection with machine learning: A systematic review and statistical analysis," *PLOS ONE*, vol. 18, no. 2, Feb. 2023. doi: 10.1371/journal.pone.0281323.

[3] V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proc. 2015 ACM Int. Conf. Multimodal Interaction*, Seattle, WA, 2015, pp. 59–66. doi: 10.1145/2818346.2820758.