Deception Detection

Harsh Rajput
2022201

Varun Kumar
2022563

Krishna Shukla
2022254

Abstract

Deception detection in natural language processing (NLP) is a challenging problem with applications in various domains, including gaming, security, and online communication. This project focuses on classifying messages exchanged between players in the strategy game Diplomacy as either deceptive or truthful. Using machine learning and NLP techniques, we aim to build an accurate classification model that can distinguish between strategic deception and honest communication. The project involves data preprocessing, exploratory analysis, feature engineering, and model experimentation, with accuracy as the primary evaluation metric. Our baseline model uses logistic regression, and further improvements will involve advanced text representations and deep learning methods.

1 Introduction

The QANTA Diplomacy project aims to develop a model capable of predicting whether messages exchanged between players in the game Diplomacy are deceptive or truthful. Diplomacy is a strategy-based game where players negotiate, form alliances, and betray each other to achieve victory. Given the strategic nature of the game, detecting deception is crucial for analyzing player behavior and improving AI decision-making.

1.1 Objective

- Build an NLP model that can accurately classify in-game messages as deceptive or truthful.
- Improve classification accuracy using advanced feature engineering and modeling techniques.

2 Related Work

Deception detection and the analysis of agreement and disagreement in textual communication are pivotal areas in natural language processing, with significant implications for understanding strategic interactions and political discourse.

Peskov et al. [2] delve into the intricacies of deception within the strategic board game *Diplomacy*, where players engage in prolonged negotiations, alliances, and betrayals. The authors compiled a dataset of 17,289 messages from 12 games, each annotated by both the sender (indicating the intent to deceive) and the receiver (perceiving the message's truthfulness). This dual-annotation approach provides a nuanced perspective on how deception is constructed and perceived over time. Their analysis revealed that deception often involves a blend of truthful and false statements, strategically employed to build trust before eventual betrayal. Machine learning models incorporating linguistic cues, conversational context, and power dynamics were developed, achieving performance comparable to human players in detecting deception. This study underscores the complexity of deceptive communication and the importance of context in its detection.

Davoodi et al. [1] examine the language of political agreement and disagreement within legislative texts at the U.S. state level. Recognizing that most legislative activity occurs at the state level, the authors constructed a large-scale dataset linking state bills with legislator information, district demographics, and donor data. They introduced a novel task: predicting the vote breakdown of legislative bodies on given bills, considering factors like gender, rural-urban divides, and ideological splits. A shared relational embedding model was proposed to capture interactions between bill texts and their legislative contexts. The study demonstrated that incorporating contextual information significantly enhances the prediction of voting outcomes, highlighting the interplay between language and political alignment.

3 Dataset

The dataset used in this study is derived from the online version of the strategic board game *Diplomacy*, and was introduced by Peskov et al. [2]. It contains 17,289 pairwise messages exchanged between players during 12 different games. Each message is annotated both by the sender (for actual intent to deceive) and the receiver (for suspected deception), providing a dual-perspective ground truth for deception detection.

The data is partitioned into training, validation, and test sets based on entire games: 9 games are used for training, 1 for validation, and 2 for testing.

3.1 Diplomacy game

Diplomacy is a seven-player strategic game set on the eve of World War I, where each player represents a European power (e.g., England, France, Russia, etc.). Players compete to conquer territories across a simplified map of Europe using armies, and the player with the most supply centers (territories) can build additional armies. Since all players begin with equal strength and the game mechanics are deterministic (no randomness or dice rolls), success hinges on forming and breaking alliances. The outcome of any move is based solely on troop counts and positioning, making deception and negotiation central to gameplay.

3.2 Annotation Scheme and Structure

Each message is annotated with two labels:

- **Sender Label (Actual Lie)**: Indicates whether the sender considered the message deceptive (`true`/`false`).
- **Receiver Label (Suspected Lie)**: Indicates whether the receiver suspected the message to be a lie (`true`/`false`/`NOANNOTATION`).

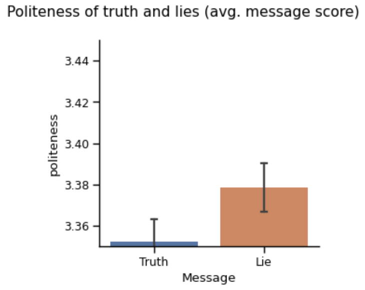
The following fields are included in each messages:

- `speakers`: The sender of the message (e.g., `england`, `russia`).
- `receivers`: The recipient of the message.
- `messages`: The text content of the message, ranging from single words to full paragraphs.
- `sender_labels`: Sender's annotation of whether the message was truthful or deceptive.
- `receiver_labels`: Receiver's perception of the message's truthfulness.
- `game_score`: Number of supply centers controlled by the sender at the time of sending.
- `score_delta`: The difference in supply centers between the sender and the receiver.
- `absolute_message_index`: Global index of the message within the game.
- `relative_message_index`: Index of the message within a specific dialogue.
- `seasons`: The season in which the message was sent (Spring, Fall, Winter).
- `years`: The year associated with the current season (ranging from 1901 to 1918).
- `game_id`: Identifier of the game instance (1 through 12).

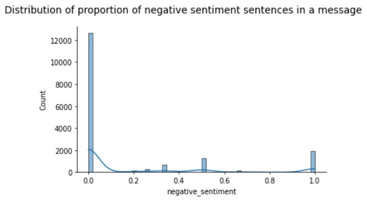
4 Preprocessing and EDA

To prepare the dataset for classification, we performed the following preprocessing steps:

- **Politeness Feature Extraction:** Used the `Politenessr` library to assess politeness levels in messages, as deceptive messages may exhibit different politeness patterns.

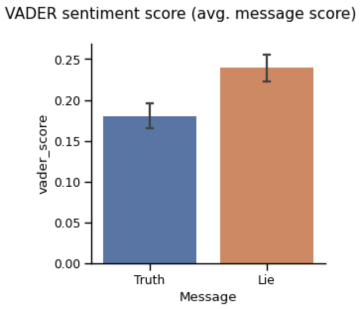


- **Sentiment Analysis with Stanza:** Extracted negative, neutral, and positive sentiment scores at the sentence level using the `Stanza` NLP library, helping to analyze emotional tone variations in deceptive messages.



When a message has negative sentiment it seems to be very negative concluding When a player gets frustrated all the negativity seems to end up in one or multiple very negative messages.

- **Sentiment Scoring with VADER:** Computed an overall sentiment polarity score for each message using VADER, capturing subtle linguistic cues associated with deception.



5 Baseline Models

To establish reference points for evaluating our models, we review a set of baseline approaches reported in the original Diplomacy dataset paper. These baselines span from simple heuristics and feature-based models to more sophisticated neural architectures that incorporate context and power dynamics. They provide valuable insight into the effectiveness of different modeling strategies for deception detection.

5.1 Heuristic and Rule-Based Baselines

- **Random:** A trivial baseline where predictions are made randomly, independent of the input features.

- **Majority Class:** Another simple baseline that always predicts the majority class (typically "truth"), which is especially problematic in imbalanced settings like deception detection. While this may yield high accuracy, its F1 score is typically low due to poor performance on the minority (deceptive) class.

5.2 Feature-Based Models

- **Bag of Words:** This model uses word frequency counts as features and feeds them into a logistic regression classifier. It provides a basic understanding of word-level patterns in deceptive versus truthful messages.

- **Bag of Words + Power:** This variant augments the bag-of-words model with power dynamics features, such as the relative score difference between sender and receiver. Including these features improves performance by accounting for strategic motivations behind deception.

- **Harbingers:** A handcrafted feature-based model that encodes cues associated with betrayal or deception, such as sudden changes in tone, urgency, or unusual expressions.

- **Harbingers + Power:** Combines behavioral linguistic cues with power dynamics to better capture the context in which lies occur.

5.3 Neural Models

- **LSTM:** A basic recurrent neural network trained on message sequences, capturing temporal patterns in player communication.

- **Context LSTM:** Extends the basic LSTM by incorporating the dialog history, allowing the model to use conversational context as additional input.

- **Context LSTM + BERT:** Combines contextual dialog modeling with BERT embeddings for improved language understanding.

- **Context LSTM + Power:** Incorporates power dynamics into the contextual LSTM, helping the model understand the strategic implications of message exchanges.

- **Context LSTM + Power + BERT:** The most advanced baseline, integrating dialog context, power dynamics, and BERT embeddings. This model achieves the highest macro F1 among the baselines and approaches human-level performance in detecting ACTUAL LIES.

5.4 Human Performance

As a reference point, the paper also reports **human baseline performance** for the ACTUAL LIE task. This is based on how often receivers correctly identify the sender's lies. While humans appear to detect a high percentage of lies in raw accuracy, their macro F1 is much lower due to class imbalance. Specifically, humans achieve an F1 score of 22.5 for detecting ACTUAL LIES. No human baseline is reported for the SUSPECTED LIE task, as it lacks objective ground truth.

6 Methodology

Our approach to deception detection is based on a supervised learning framework, where the goal is to classify messages as deceptive or truthful. We evaluate the effectiveness of multiple language representations combined with various classification models.

6.1 Feature Representations

We experiment with four different language models to encode messages into fixed-length vector representations:

- **RoBERTa**: A robustly optimized BERT variant trained on large-scale corpora, known for strong performance across many NLP tasks.
- **BERT**: The Bidirectional Encoder Representations from Transformers model, which provides contextualized embeddings of text.
- **MiniLM**: A lightweight transformer model that offers faster computation with competitive performance, making it suitable for resource-constrained environments.
- **GloVe**: A static word embedding model trained on global word co-occurrence statistics, used here by averaging the embeddings of words in each message.

For transformer-based models (RoBERTa, BERT, MiniLM), we use the CLS token embedding as the representation of the entire message. For GloVe, we compute the mean of the embeddings for all tokens in a message.

6.2 Classification Heads

Each of the above representations is paired with four types of classification models:

- **Logistic Regression (LR)**: A linear model that outputs probabilities using a sigmoid function, serving as a simple and interpretable baseline.
- **Support Vector Machine (SVM)**: A maximum-margin classifier effective for high-dimensional spaces.
- **Random Forest (RF)**: An ensemble of decision trees trained via bagging, useful for capturing nonlinear decision boundaries.
- **Multilayer Perceptron (MLP)**: A fully connected neural network with one hidden layer, capable of modeling more complex relationships in the data.

7 Evaluation Metric

The primary evaluation metric for this task is **accuracy**, which measures the proportion of correctly classified messages.

$$\text{Accuracy} = \frac{\text{Correctly classified messages}}{\text{Total messages}} \tag{1}$$

However, accuracy alone can be misleading due to the strong class imbalance in the dataset: fewer than 5% of messages are labeled as lies in both the `ACTUAL_LIE` and `SUSPECTED_LIE` settings [2]. As such, a classifier predicting all messages as truthful could still achieve high accuracy without learning meaningful deception signals.

To address this issue, we use the **macro-averaged F1 score** as a more reliable metric, particularly for evaluating the detection of `ACTUAL_LIE`.

$$\text{F1}_{\text{macro}} = \frac{1}{2}\left(\text{F1}_{\text{truth}} + \text{F1}_{\text{lie}}\right) \tag{2}$$

Human performance on this task has been measured as a baseline: while humans correctly identify 88.3% of lies, the corresponding Lie F1 score is only 22.5 due to the skewed distribution.

8 Results and Analysis

Table 1 presents the macro F1 scores obtained from each combination of language model and classification head on the `ACTUAL_LIE` detection task. Given the imbalanced nature of the dataset—with deceptive messages making up less than 5% of the total—it is crucial to rely on metrics like macro F1 rather than accuracy to gain a more faithful representation of model performance. Our experimental results reveal several interesting trends across both model types and classifier architectures.

8.1 Comparison of Language Models

Among the four language models used—BERT, RoBERTa, MiniLM, and GloVe—**BERT** emerged as the top performer overall. Specifically, BERT paired with Logistic Regression achieved the highest macro F1 score of **53.6**, indicating that BERT's deep contextualized embeddings are well-suited for detecting nuanced deception cues embedded in player messages.

RoBERTa also showed strong performance, particularly when combined with the MLP classifier, reaching a macro F1 of **52.4**. RoBERTa's architecture, which benefits from more extensive pretraining on a diverse corpus, seems to provide an edge in capturing contextual subtleties that are relevant to deception.

MiniLM, while smaller and more efficient, still managed to hold its ground with respectable scores. It achieved a macro F1 of **50.5** when paired with both the Random Forest and Logistic Regression classifiers. This highlights MiniLM's capability as a lightweight alternative, offering competitive results despite its reduced size and training footprint.

On the other hand, **GloVe**—a static word embedding model—demonstrated consistently weak performance across all classifiers, with all combinations yielding a macro F1 of **48.5**. This flat result underscores the limitations of static embeddings in tasks requiring contextual understanding and subtle pragmatics, both of which are crucial in identifying lies in dialogue.

8.2 Comparison of Classification Heads

From the perspective of classification algorithms, the **MLP (Multi-Layer Perceptron)** classifier consistently provided strong or best-in-class performance across several language models. Its ability to model nonlinear relationships likely plays a key role in its effectiveness, especially when handling high-dimensional contextual embeddings.

Random Forest classifiers also showed competitive performance, particularly with RoBERTa and MiniLM, both achieving macro F1 scores of **50.5**. Random Forest's ensemble approach appears to be beneficial in capturing varied decision boundaries, even with limited and imbalanced data.

In contrast, **SVM** and **Logistic Regression** performed somewhat inconsistently. While they produced moderate scores for most configurations, their linear nature might have limited their expressiveness. That said, it is notable that **BERT + Logistic Regression** yielded the highest macro F1 score overall, suggesting that strong embeddings can sometimes compensate for simpler classifiers.

8.3 Overall Insights

The experimental findings reinforce the importance of both representation and classification in the task of deception detection. Transformer-based models such as BERT and RoBERTa clearly outperform traditional word embeddings, highlighting the critical role of contextual understanding in this domain. Furthermore, model performance appears to be sensitive to the choice of classifier: pairing rich language models with more expressive classifiers like MLP can lead to notable gains.

Despite these improvements, the absolute values of macro F1 remain modest, with the highest score barely exceeding 53. This modest performance reflects the inherent difficulty of the deception detection task—particularly in a game like Diplomacy, where lies are deeply embedded in strategic reasoning and conversational nuance. The severe class imbalance also plays a role, making it challenging for models to generalize well to the minority class.

Going forward, combining language models with discourse-level context, incorporating dialog history, and experimenting with advanced loss functions or data augmentation techniques may provide paths toward better performance on this complex task.

9 Conclusion

Table 1: Macro F1 scores for each model–classifier combination on the `ACTUAL_LIE` detection task.

Model	Log. Reg.	RF	SVM	MLP
RoBERTa	48.5	50.5	48.5	52.4
BERT	53.6	51.0	49.0	49.0
MiniLM	48.5	50.5	48.5	48.5
GloVe	48.5	48.5	48.5	48.5

9.1 Learning from the Project

Throughout the course of this project, we gained valuable knowledge and hands-on experience in the fields of natural language processing, machine learning, and applied research. Some of the key takeaways include:

- **Understanding Real-World NLP Challenges:** We worked with a complex and highly nuanced dataset from the Diplomacy game, which exposed us to real human communication patterns, including strategic deception and alliance-building.

- **Working with Annotated Dialog Data:** We learned how to preprocess and structure dialog-based datasets with annotations for ACTUAL_LIE and SUSPECTED_LIE, understanding both sender and receiver perspectives in human conversations.

- **Dealing with Class Imbalance:** One of the most important challenges was handling a highly imbalanced dataset, where only a small percentage of messages were labeled as deceptive. This taught us how metrics like macro F1 can provide a fairer evaluation compared to accuracy, and how loss functions and model training strategies need to be adapted accordingly.

- **Exploring and Comparing NLP Models:** We experimented with both traditional word embedding models (GloVe) and modern transformer-based models (BERT, RoBERTa, MiniLM). We observed firsthand the improvements contextual embeddings provide for capturing subtle linguistic cues associated with lies.

- **Experimenting with Classifiers:** We integrated multiple classification heads including Logistic Regression, Random Forests, SVMs, and MLPs to observe how different architectures affect model performance when paired with various embeddings.

- **Model Evaluation and Interpretation:** We developed an understanding of how to design fair experiments, tune hyperparameters, and evaluate models in a way that reflects real-world performance, especially in sensitive tasks like deception detection.

- **Team Collaboration and Project Management:** Through task division, documentation, and weekly progress discussions, we also improved our skills in teamwork, communication, and managing the end-to-end machine learning pipeline in a research context.

These experiences have strengthened our foundation in applied NLP research and equipped us with practical tools for future machine learning and AI projects.

9.2 Member Contribution

- **Harsh Rajput (2022201):** Responsible for dataset preprocessing, implementing the classification pipelines, and training baseline models using GloVe and MiniLM.

- **Krishna Shukla (2022254):** Focused on integrating transformer-based models (BERT and RoBERTa), fine-tuning, and performance analysis.

- **Varun Kumar (2022563):** Led the documentation, LaTeX report structuring, and contributed to evaluation, visualization, and final analysis of results.

All members contributed equally to discussion, literature review, and weekly progress updates.

9.3 Future Work

There are several promising directions to extend this project:

- Incorporating temporal and conversational context using sequence-based models or graph neural networks.

- Exploring contrastive learning or multi-task learning to jointly model ACTUAL and SUSPECTED lies.

- Addressing class imbalance through advanced sampling techniques or synthetic data augmentation.

- Fine-tuning domain-specific transformer models with a focus on pragmatic and discourse-level features of deception.

This project lays a solid foundation for future research in deception detection and conversation modeling in strategic environments.

References

[1] Maryam Davoodi, Xuan Li, Matthew Lease, and Byron C. Wallace. 2020. Understanding the language of political agreement and disagreement in legislative texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[2] Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It takes two to lie: One to lie and one to listen. In *Association for Computational Linguistics*.