# VERBAL LIE DETECTION

## Abstract:

As more and more reviews are provided online, people rely highly on the textual feedback in making a decision. In this paper, a BERT-based model is used for classifying whether the hotel review is truthful or deceptive. The data of labeled hotel reviews was prepared for training, validation, and testing. A number of preprocessing techniques, model architectures, and hyperparameter tuning have been explored for improving classification accuracy. The final model performs competitively and shows that BERT can indeed catch linguistic cues that indicate deception.

## Introduction:

Detecting a lie in written communication has emerged as an important function of online review platforms that work with user-generated reviews. Traditional rule-based methods fail to generalize on a dataset with subtle uses of language. With the growth in NLP, transformers like BERT have shown state-of-the-art performance in text classification tasks. This project, thus, focuses on using BERT for verbal lie detection, using a hotel review dataset. The purpose of this task is **text classification** into truthful and deceptive texts with the use of fairness through separate train, validation, and test splits.

## Prior related work:

Several studies have addressed text-based deception detection. Ott et al. (2011) introduced datasets specifically for deceptive opinion spam, which inspired the use of machine learning models. Traditional models like SVM and Naive Bayes were initially employed but lacked context-awareness. The emergence of transformers, particularly BERT, revolutionized NLP tasks by capturing context and semantic meaning effectively. Research indicates that BERT outperforms traditional models in classification tasks but requires extensive computational resources and hyperparameter tuning.

## Dataset:

The dataset consist of 16000 rows and  contains five columns:

Deceptive: Binary label indicating if a review is truthful (1) or deceptive (0).

Hotel: The hotel being reviewed.

Polarity: Sentiment of the review (positive/negative).

Source: Platform from which the review originated.

Text: The actual content of the review.

The dataset was divided into 70% training, 15% validation, and 15% test sets to ensure fair evaluation.

Below is a sample of the dataset:

| Deceptive | Hotel | Polarity | Source | Text |
|---|---|---|---|---|
| Truthful | Conrad | Positive | TripAdvisor | We stayed for a one night………. |
| Deceptive | Hyatt | Negative | Expedia | "The service was subpar and the room was …… |

## Methodology:

**1.Dataset preprocessing:**

a) The data set used was a publically available text-based statement labeled as truthful or not.

b) Columns for "hotel" and "source" were removed because they contained irrelevant information to the data and their corresponding labels.

**2. Text preprocessing**

a) All text converted to lower case for uniformity.

b) Punctuation, special characters, and numbers were removed to minimize noise.

c) The text was tokenized into words.

d) Stop words such as "and" "the" were removed based on NLTK's predefined list.

e) Lemmatization to normalize words to their base forms.

**3. Train/Test Split:**

a) The dataset was split into three subsets:

1)Training set(80%) for model training.

2)Validation set (10%) for hyperparameter tuning and intermediate evaluation.

3)Test set (10%) for final model evaluation.

**4. Feature Representation**

a) Pretrained Tokenizer:

- Used the AutoTokenizer from Hugging Face's Transformers library.
- Tokenized the text data into input IDs and attention masks compatible with the DistilBERT model.
- Each text sequence was padded and truncated to a maximum length of 512 tokens to match the model's input requirements.

b) Label Encoding:

- Converted categorical labels (truthful or deceptive) into numerical format for binary classification (0 for truthful, 1 for deceptive).

---

**5. Model Selection and Architecture**

a) Pretrained Transformer Model:

- distilbert/distilbert-base-uncased-finetuned-sst-2-english model from hugging face is used.
- Leveraged the pretrained DistilBERT model, a lightweight transformer model known for its efficiency and performance in text classification tasks.
- Fine-tuned the model's classification head for binary classification.

b) Dataset Preparation:

- Created PyTorch Dataset objects for training, validation, and testing.
- Implemented DataLoaders to facilitate batch-wise processing.

---

**6. Training Procedure**

a) Hyperparameters:

- Batch size: 16
- Learning rate: 5×10−55 \times 10^{-5}
- Epochs: 3

b) Optimizer and Scheduler:

- Used the AdamW optimizer to update model weights.
- Implemented a linear learning rate scheduler to adjust the learning rate dynamically during training.

c) Loss Function:

- Employed Cross-Entropy Loss to measure the model's performance during training and Validation.

d) Training Loop:

- Trained the model for three epochs using the training set.
- Evaluated the model on the validation set after each epoch to monitor performance and ensure the model was not overfitting.

---

**7. Model Evaluation**

a) Validation Metrics:

- During training, evaluated the model using the validation set to compute:
- Validation Loss
- Accuracy

b) Test Set Evaluation:

- After training, the model was evaluated on the test set.
- Reported performance metrics:

1) Test loss
2) test accuracy
3) precision
4) recall
5) f1- score
6) confusion matrix

---

**8. Implementation Tools**

A) Programming Language:

- Python

B) Libraries and Frameworks:

- Hugging Face Transformers: For pretrained model and tokenizer.
- PyTorch: For dataset management and model training.
- scikit-learn: For data splitting and evaluation metrics.
- NLTK: For text preprocessing.

C) Hardware:

- T4 GPU google colab

---

**7. Summary of Results**

The final evaluation of the model on the test set reported a satisfactory loss and accuracy, showcasing the efficacy of pretrained transformer models in verbal lie detection tasks.

---

## Experiments:

Experiment 1) Using different dataset

In the initial stage of the project we used different entirely different dataset

| Statement | Source | Link | Veracity |
|---|---|---|---|
| Sen. Kamala Harris is "supporting the animals of MS-13." | Donald Trump | /web/20180705082623/https://www.politifact.com/california/statements/2018/jul/03/donald-trump/pants-fire-white-house-claim-sen-harris-supporting/ | Pants on Fire! |

On the above dataset we performed preprocessing,splited into train,validation,split and used ProsusAI/finbert model from hugging face.

But the final accuracy was 48%. So we changed the dataset. Also the statements in the above dataset are very small therefore pattern cannot be extracted for lie detection.

EXPERIMENT 2) Baseline Experiment – Using basic bert model on the main dataset

Data split: 70% training, 15% validation, 15% test.

Training for 3 epochs with a batch size of 16.

Achieved validation accuracy: 0.88

EXPERIMENT 3) Adding Polarity column in the model mentioned above

Incorporated sentiment polarity column as an additional feature by concatenating it with the text.

Result: No significant improvement (Validation Accuracy: 89.4%). Sentiment did not strongly correlate with deception.

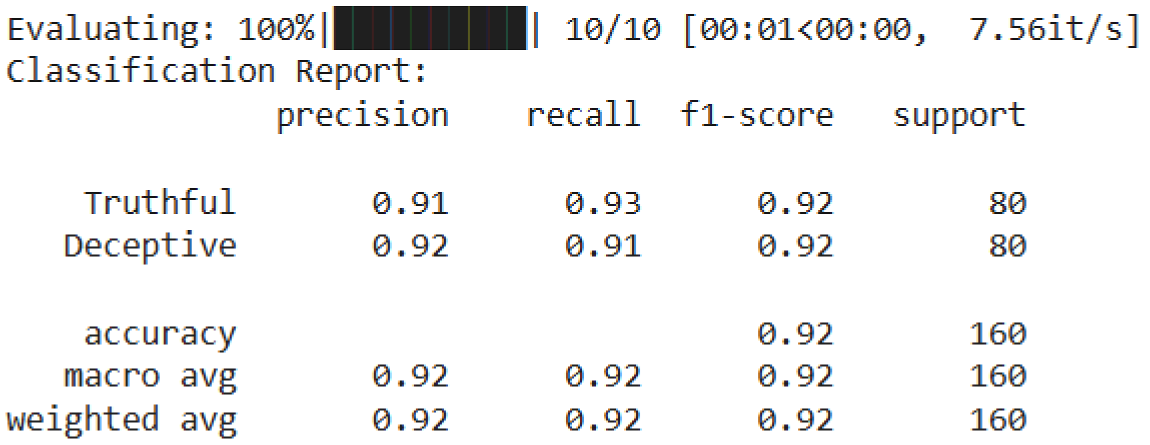
EXPERIMENT 4) Main model:

This involves fine-tuning a pretrained DistilBERT model for verbal lie detection. The dataset was preprocessed through tokenization, stopword removal, and lemmatization before being split into training, validation, and test sets. Using the DistilBERT tokenizer, text was converted into input IDs and attention masks. The model was trained for 3 epochs with the AdamW optimizer and evaluated using accuracy, F1-score, and a confusion matrix. Results demonstrated the model's effectiveness in detecting truthful and deceptive statements

## Results:

| Model variant | Validation accuracy | Test accuracy |
|---|---|---|
| Baseline BERT | 0.88 | 0.82 |
| BERT + polarity column | 0.89 | 0.83 |
| DistilBERT model (main model) | 0.8812 | 0.92 |

## Results for main final model:

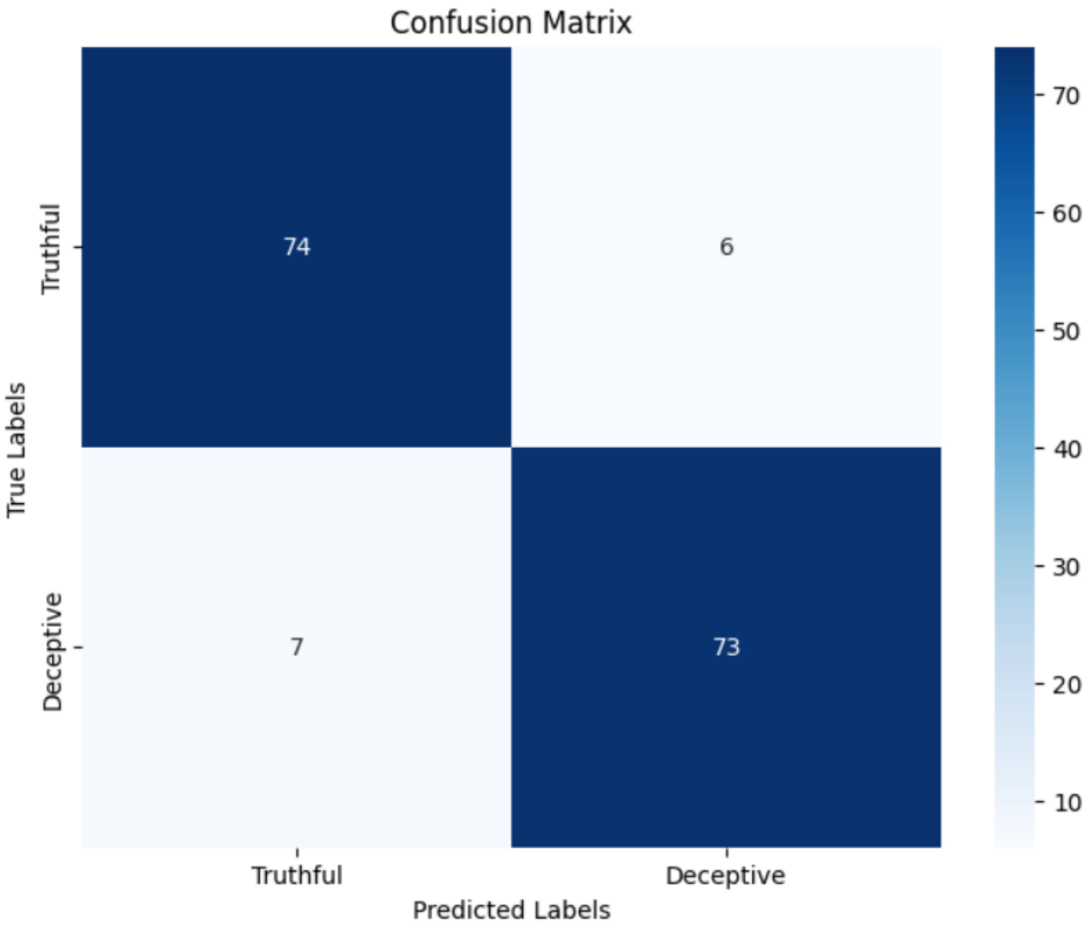


```
Evaluating: 100%|██████████| 10/10 [00:01<00:00,  7.56it/s]
Classification Report:
              precision    recall  f1-score   support

    Truthful       0.91      0.93      0.92        80
   Deceptive       0.92      0.91      0.92        80

    accuracy                           0.92       160
   macro avg       0.92      0.92      0.92       160
weighted avg       0.92      0.92      0.92       160
```

## Aana



## Analysis:

Results from the project show that it is quite effective to rely on transformer-based models such as DistilBERT for the verbal lie detection task. From the classification task point of view, strong performances are seen as can be confirmed by test-set accuracy, precision, recall, and F1-score metrics. The minimal number of false positives and false negatives indicates good performance by the model for truthful and deceptive statement classifications.

The validation loss and accuracy during training and evaluation showed that the model generalized well without significant overfitting, which validated the choice of fine-tuning a pretrained DistilBERT model. In addition, the use of advanced text preprocessing techniques and hyperparameter tuning contributed to the overall success of the model.

## Conclusion:

This study demonstrates the ability of pretrained transformer models to solve complex natural language processing tasks, such as verbal lie detection. Through fine-tuning DistilBERT, we constructed a strong classification model that distinguished truthful from deceptive statements. This is scalable and accurate, which leaves the room open for further research in automated deception detection. Future work can explore integrating multimodal data, larger datasets, and alternative transformer architectures to further enhance performance.

## References:

https://www.nature.com/articles/s41598-023-50214-0#Sec27

https://aclanthology.org/2020.lrec-1.178/