Detection of Deceptive Hotel Reviews using Machine Learning Models

Authors

Ankita De (5269326)

Izumi Abril Jimenez-Alatriste (1392778)

Liu Yi-An (7578997)

October 17, 2025

Abstract

This project investigates the detection of deceptive hotel reviews using various machine learning models trained on the *Deceptive Opinion Spam Corpus v1.4*. The dataset includes 800 negative reviews, which are evenly divided between truthful and deceptive samples. Multiple models were implemented, including *Multinomial Naive Bayes*, *Logistic Regression* with L1 (Lasso) penalty, *Single Classification Tree* (non-linear classifier), *Random Forest*, and *Gradient Boosting*. These models were trained using TF–IDF unigram and bigram features, along with hyperparameter tuning via `GridSearchCV` or `RandomizedSearchCV` with cross-validation. The performance of the models was evaluated using accuracy, precision, recall, and F1-score, followed by pairwise McNemar's statistical test to determine whether differences in classification performance were statistically significant.

Our results showed that the *Multinomial Naive Bayes* model has achieved the highest accuracy (88.75%), closely followed by *Gradient Boosting* and *Random Forest*, with no statistically significant difference among the top four models. The *Decision Tree* classifier, however, performed significantly worse ($p < 0.001$). These findings highlight that while ensemble and probabilistic approaches outperform simple tree-based models, the overall improvements among top-performing classifiers are not statistically significant.

Contents

1 Introduction

Online review platforms such as TripAdvisor and Yelp have become an important source of information for consumers. However, their popularity has also led to the rise of opinion spam—fake reviews written to promote or damage a company's reputation. Detecting these deceptive reviews is challenging because they are often crafted to sound authentic and natural.

Previous studies, such as those by Ott *et al.* [1, 2], have shown that linguistic features can help distinguish between truthful and deceptive reviews using linear models like Naive Bayes and Support Vector Machines. In addition, Mukherjee *et al.* [3] and Li *et al.* [4]. examined the use of logistic regression and random forests for large-scale opinion spam detection.

In this study, we extend their work by evaluating whether more flexible, non-linear classifiers can improve detection performance. We compare Multinomial Naive Bayes, L1-regularized Logistic Regression, Classification Trees, Random Forests, and Gradient Boosting on a dataset of 800 hotel reviews (400 truthful and 400 deceptive) focussing specifically on the Negative reviews. In addition to the evaluation of predictive performance using accuracy, precision, recall and F1 score, we also analyze whether addition of bigram features improves the accuracy and examine the most informative linguistic features that are associated with truthful and deceptive reviews. In the end, a pairwise McNemar's statistical test was also performed on the models trained same test data to ensure that the observed difference in performance was not due to random variation. This statistical test has helped us to determine whether any differences in the classification accuracy are statistically significant or not. With the combination of these two analysis, we get a deeper understanding of the relative strengths of different models used and their linguistic characteristics that distinguish genuine from deceptive reviews.

2 Dataset

The dataset utilized in this study is derived from the Deceptive Opinion Spam Corpus v1.4 compiled by Myle Ott *et al.* .[1, 2]. This corpus contains hotel reviews manually collected and labelled as either truthful or deceptive. Each review is about one of 20 Chicago hotels, ensuring a balanced set of genuine and fabricated opinions.

The corpus consists of 400 positive and 400 negative reviews in total, each divided evenly into truthful and deceptive categories. In this project, we focus exclusively on the negative reviews and determine whether they are truthful or deceptive. This results in a total of 800 reviews, evenly split into 400 truthful and 400 deceptive samples.

Several preprocessing techniques were applied, including punctuation removal, lowercasing, tokenization, stopword removal, and lemmatization. These ensured clean and consistent textual input for TF-IDF vectorization.

3 Methods

3.1 Simple Classification Tree

A Decision Tree classifier was implemented to distinguish between truthful and deceptive hotel reviews. This classifier was trained and evaluated using a fold-based validation strategy to ensure robustness and minimize bias. Specifically, the dataset contained five predefined folds: Folds 1–4 were used for training and hyperparameter tuning, while Fold 5 served as an independent test set for the final evaluation. Furthermore, to prevent any possibility of data leakage, `GroupKFold` cross-validation (4 splits) was applied during tuning so that reviews from the same fold were kept together in either the training or validation splits.

For preprocessing, all the reviews were first loaded and then labelled as 0 for truthful and 1 for deceptive, each associated with its corresponding fold identifier. During vectorization, text preprocessing steps such as lowercasing and stopword removal (using the English stopword list from `scikit-learn`) were automatically handled by the TF–IDF vectorizer. This representation converts each document into a weighted feature vector, where each feature reflects the relative importance of a word within the corpus.

Two Decision Tree models were developed:

1. **Unigram:** using single-word TF–IDF features.
2. **Unigram + Bigram:** combining unigram TF–IDF features from the best model after hyperparameter finetuning with additional bigram TF–IDF features using a `FeatureUnion`. The combined model was also hyperparameter optimized. The relative contribution of unigram and bigram features was also tuned through `transformer_weights`.

The particular usage of the features in the best hyperparameter optimized unigram model was done to have a fair comparison between the two model and to check whether addition of bigram features approves the model accuracy or not. Model hyperparameters were optimized using `RandomizedSearchCV`, which randomly searches for the best hyperparameters with 30 random configurations using the F_1 score as the performance metric. This search method was particularly chosen as it efficiently explores a broad range of

parameters while remaining computationally feasible.

The hyperparameters tuned were as follows: `max_depth` to limit tree growth and prevent overfitting; `min_samples_split` and `min_samples_leaf` to ensure a sufficient number of samples per split or leaf for stability; `ccp_alpha` and `min_impurity_decrease` for pruning weak branches to regularize complexity; three different splitting criteria were compared for measuring node impurity, namely `gini`, `entropy`, and `log_loss`; and lastly, for vectorization, `min_df`, `max_df`, and `max_features` were tuned to balance vocabulary size and noise reduction.

For reproducibility, all models used balanced class weights and a fixed random seed (42).

3.2 Multinomial Naive Bayes

A Multinomial Naive Bayes classifier was employed within a two-stage experimental workflow to evaluate the impact of different feature combinations on model performance.

The process begins with text preprocessing, including tokenization, stop-word removal, and lemmatization, to extract meaningful linguistic features. Preprocessing was applied to improve model performance, as the initial experiments showed that many tokens had little predictive value. Terms such as "...", "'s", "“", "'re", "n't", "”””, and "i.e." were observed to appear frequently but carry minimal semantic meaning. By treating these as stop words, the models were able to focus on more informative terms, resulting in significantly better predictive accuracy.

Subsequently, two distinct feature sets are used to train and evaluate the models. The first is a baseline model that uses only unigram features. The second is a model utilizing a mixed vocabulary of both unigrams and bigrams. This design allows for comparison between a simple word frequency model and one that captures local contextual information. The entire experimental workflow, from feature engineering to model training and evaluation, is encapsulated within a reproducible Scikit-learn pipeline to ensure the robustness and consistency of the results.

Two distinct feature sets were used:

1. **Unigram:** This model utilizes single-word features, extracted using `CountVectorizer` with an `ngram_range` of (1,1).
2. **Unigram + Bigram:** This model utilizes a mixed vocabulary of single-word and two-word phrase features, extracted using a single `CountVectorizer` with an

`ngram_range` of (1,2). In this setup, unigrams and bigrams compete for inclusion in the final feature set based on their frequency.

Hyperparameter tuning was performed using `GridSearchCV` with `StratifiedKFold` cross-validation (`n_splits=4`) on the training folds to optimize model performance. The hyperparameter space for both experiments included:

1. `vect__max_features`: [500, 1000, 2000, 3000, 3500, 4000
2. `clf__alpha`: [0.01, 0.05, 0.1, 0.5, 1.0]

For `vect__max_features`, a range from 500 to 4000 was chosen to evaluate how different vocabulary sizes affect model performance, balancing the need to capture meaningful words while avoiding excessively sparse feature matrices that could slow down training. For `clf__alpha` in Multinomial Naive Bayes, values between 0.01 and 1.0 were used to examine the effect of Laplace smoothing, as fake reviews often contain uncommon or highly specific words that could otherwise be underrepresented.

StratifiedKFold ensures that each fold maintains the same class distribution as the original dataset.

Model performance was evaluated on the held-out test set (fold 5) using standard metrics: precision, recall, F1-score, and accuracy. Additionally, the top five most indicative features for both fake and genuine reviews were identified based on the classifier's feature log probabilities to provide insight into the model's decision-making process.

3.3 Logistic Regression with Lasso Penalty

A Logistic Regression classifier with an L1 (Lasso) penalty was employed within a two-stage experimental workflow to evaluate the impact of different feature combinations on model performance.

The process begins with text preprocessing—including tokenization, stop-word removal, and lemmatization—to extract meaningful linguistic features. Preprocessing was applied because initial experiments showed that many tokens had little predictive value, such as "...", "'s", "“", "'re", "n't", "”””, and "i.e.", which occur frequently but carry minimal semantic information. Treating these as stop words allowed the model to focus on more informative terms, resulting in improved predictive performance.

Subsequently, two distinct feature sets are used to train and evaluate the models. The first

is a baseline model utilizing only unigram features. The second is a model utilizing a mixed vocabulary of both unigrams and bigrams. This design allows for comparison between a simple word-frequency model and one that captures local contextual information. The entire experimental workflow, from feature engineering to model training and evaluation, is encapsulated within a reproducible Scikit-learn Pipeline to ensure the robustness and consistency of the results.

Two distinct feature sets were employed:

1. **Unigram:** This model utilizes single-word features, extracted using `CountVectorizer` with an `ngram_range` of (1,1).
2. **Unigram + Bigram:** This model utilizes a mixed vocabulary of single-word and two-word phrase features, extracted using a single `CountVectorizer` with an `ngram_range` of (1,2). In this setup, unigrams and bigrams compete for inclusion in the final feature set based on their frequency.

Hyperparameter tuning was performed using `GridSearchCV` with `StratifiedKFold` cross-validation (`n_splits=4`) on the training folds to optimize model performance. The hyperparameter space for both experiments included:

1. `vect__max_features`: [500, 1000, 2000, 3000, 3500, 4000], chosen to explore the effect of vocabulary size on model performance, balancing the inclusion of informative terms with computational efficiency.
2. `clf__C`: [0.001, 0.01, 0.1, 1, 10] (inverse of regularization strength), selected to investigate different levels of L1 regularization and prevent overfitting while retaining meaningful coefficients for feature interpretation.

`StratifiedKFold` ensures that each fold maintains the same class distribution as the original dataset, which helps produce a more reliable estimate of model performance.

Model performance was evaluated in the held-out test set (fold 5) using standard metrics: precision, recall, F1 score, and accuracy. Additionally, the top five most indicative features for both fake and genuine reviews were identified by inspecting the model's coefficients (`clf.coef_`).

3.4 Random Forest

A Random Forest classifier was implemented to distinguish between truthful and deceptive reviews, integrating both unigram and bigram textual representations to assess whether the ensemble approach provides advantages over single-tree baselines. The model was developed within a fold-based experimental framework to ensure fairness, robustness, and comparability across classifiers. Specifically, **Folds 1–4** were used for model training and hyperparameter optimization, while **Fold 5** served as an independent test set for final evaluation. To minimize the risk of information leakage, the model leveraged its intrinsic Out-of-Bag (OOB) estimation during tuning, thus eliminating the need for additional cross-validation splits and preserving the independence of the test data.

Model Variants

Two variants of the Random Forest model were developed to investigate the effect of lexical granularity on classification performance. The Unigram model employed single-word features extracted via a `CountVectorizer` with `ngram_range = (1,1)`, and dimensionality was reduced through **chi-squared feature selection (`SelectKBest`)**, retaining up to $k = 3000$ features. The **Unigram + Bigram model** extended this representation by incorporating contiguous two-word sequences (`ngram_range = (1,2)`) under the same feature budget, allowing a controlled comparison between single and contextual lexical representations. This approach follows prior work on deceptive opinion detection that emphasizes the importance of local context in identifying linguistic cues of deception.

Random Forest Architecture and Hyperparameter Optimization

The RandomForest algorithm constructs an ensemble of decision trees, $\{h_t(x)\}_{t=1}^{T}$, each trained on a bootstrap sample of the training data while considering a random subset of features at each split. The final prediction is obtained through majority voting across the ensemble:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\}.$$

This dual randomization—data bagging and feature sub-sampling—decorrelates the individual trees, thereby reducing variance and improving generalization.

Key hyperparameters tuned include:

- `n_estimators`: number of trees in the ensemble,

- `max_depth`: maximum depth of each tree,
- `max_features`: number of features considered per split,
- `min_samples_split` and `min_samples_leaf`: minimum samples per node,
- `bootstrap = True`: enabling Out-of-Bag evaluation.

3.5 Gradient Boosting

A Gradient Boosting classifier was implemented to distinguish between truthful and deceptive reviews, following the same, fold base framework adopted likewise for the other models. Specifically, **Folds 1–4** were used for model training and hyperparameter tuning, while **Fold 5** served as an independent test set to ensure fair evaluation. Gradient Boosting constructs an additive ensemble of shallow decision trees, where each successive learner corrects the residual errors of the preceding ensemble. This iterative, stage-wise process regulated by the learning rate, the **number of estimators**, and the **tree depth**progressively refines the model's predictions, striking a balance between bias reduction and variance control to improve overall generalization.

Data Access and Labeling

`iter_review_files` streams review files from either a folder or a ZIP archive of the dataset. `load_dataframe` parses the terminal path components to recover the fold identifier (`fold1`–`fold5`) and the label directory (`truthful_from_Web` vs. `deceptive_from_MTurk`), mapping these to $\{0, 1\}$ via `LABEL_MAP`. The resulting DataFrame (`fold`, `label`, `text`, `path`) is summarized to verify corpus size, class balance, and fold coverage prior to feature construction.

Text Representation and Feature Budget `vectorize_and_select` builds a sparse document term matrix using `CountVectorizer` with a configurable n-gram policy and a minimum document-frequency filter. For the Unigram model, `ngram_range=(1,1)` and `min_df=2` with `k_uni=2000`; for the Unigram+Bigram model, `ngram_range=(1,2)` with `min_df=3` and `k_unibi=4000`. In both cases, χ^2-based selection via `SelectKBest(chi2)` enforces a fixed feature budget, returning the selected vocabulary (`feats_selected`) to facilitate interpret ability. This then yields so that $(X_{\text{train}}, X_{\text{test}})$ are directly comparable across variants under a controlled dimensionality.

Hyperparameter Tuning, Optimization, and Cross-Validated Model Selection

Hyperparameter optimization was performed via a systematic grid search using `grid_search_gbc`, which applied 5-fold stratified cross-validation (`StratifiedKFold`, k=5). Each parameter configuration was assessed by its mean cross-validation accuracy, and the configuration achieving the highest score was retained as optimal. The parameter grid explored was defined as:

$$
\begin{aligned}
&\texttt{n\_estimators} && \in \{200, 400\},\\
&\texttt{learning\_rate} && \in \{0.05, 0.1\},\\
&\texttt{max\_depth} && \in \{2, 3\},\\
&\texttt{subsample} && \in \{0.7, 1.0\},\\
&\texttt{max\_features} && \in \{\texttt{None}, \texttt{"sqrt"}\}.
\end{aligned}
$$

Per candidate, GradientBoostingClassifier is evaluated `cross_val_score`with scoring="accuracy".

4 Results

4.1 Simple Classification Tree

The Decision Tree classifiers were evaluated using `GroupKFold` cross-validation for hyper-parameter tuning and an independent test fold (Fold 5) for final validation. This approach ensures consistent, non-overlapping evaluation between the training and testing sets. The metrics used to evaluate the models were accuracy, precision, recall, and F_1 score.

After fine-tuning, the unigram Decision Tree achieved higher performance than the Decision Tree with both unigram and bigram features. Table 4.1 summarizes the evaluation metrics on the test fold for both models.

Table 4.1. Decision Tree Performance

Model	Accuracy	Precision	Recall	F1-score
Unigram	0.681	0.68	0.68	0.68
Unigram + Bigram	0.65	0.65	0.65	0.65

For the unigram model, truthful reviews were classified slightly more accurately than deceptive ones, with recall values of 0.70 and 0.66, respectively. For the unigram and bigram model, the truthful class showed a recall of 0.71, while the deceptive class dropped slightly to 0.59. Overall, the unigram model was more accurate, achieving an accuracy of 68.1%, compared to the Decision Tree with unigram and bigram features, which achieved an accuracy of 65%.

To enhance interpretability, feature importance from the Decision Tree of the unigram model and point-biserial correlations between feature presence and class labels were analyzed. Only the unigram model was used for this analysis, as it achieved the highest accuracy among the two models. The top five features that indicate deceptive and truthful reviews are reported in Table 4.2.

Table 4.2. Top Indicative Terms (Decision Tree)

Fake Terms	Importance	Corr.	Genuine Terms	Importance	Corr.
chicago	0.470	0.419	called	0.028	-0.050
decided	0.085	0.219	–	–	–
turned	0.081	0.159	–	–	–
east	0.067	0.047	–	–	–
finally	0.065	0.206	–	–	–

4.2 Multinomial Naive Bayes

The performance of the multinomial Naive Bayes classifiers is summarized in Table 4.3. The unigram model achieved an accuracy of 88.75%, with precision, recall, and F1-score all equal to 0.8875. Adding bigram features slightly decreased overall performance, resulting in an accuracy of 88.12% and an F1-score of 0.8809. This suggests that for this dataset, the unigram model is slightly more effective than the combined unigram-bigram model.

Table 4.3. Naive Bayes Performance

Model	Accuracy	Precision	Recall	F1-score
Unigram	0.8875	0.8875	0.8875	0.8875
Unigram + Bigram	0.8812	0.8861	0.8812	0.8809

The most indicative terms for distinguishing fake and genuine reviews are shown in Table 4.4. Words such as relax, originally, and luxury are strong indicators of fake reviews, while terms like priceline, sofa, and fridge are associated with genuine reviews. These feature importances are consistent with the expectation that deceptive reviews often use subjective and emotionally positive language, whereas genuine reviews tend to reference concrete and factual details.

Table 4.4. Top Indicative Terms (Naive Bayes)

Fake Terms	Importance	Corr.	Genuine Terms	Importance	Corr.
relax	2.73	0.15	priceline	-3.03	-0.15
originally	2.58	0.13	sofa	-2.69	-0.13
luxury	2.56	0.22	25	-2.69	-0.14
settled	2.42	0.13	fridge	-2.62	-0.12
egg	2.32	0.11	stated	-2.55	-0.12

4.3 Logistic Regression with Lasso Penalty

The performance of the logistic regression classifiers with Lasso regularization is summarized in Table 4.5. Using unigram features, the model achieved an accuracy of 85.00%, with precision, recall, and F1-score all equal to 0.8500. Adding bigram features slightly reduced performance, resulting in an accuracy of 83.77% and an F1-score of 0.8375. This indicates that, for logistic regression, unigram features are more effective than the combined unigram-bigram features on this dataset.

Table 4.5. Logistic Regression Performance

Model	Accuracy	Precision	Recall	F1-score
Unigram	0.8500	0.8500	0.8500	0.8500
Unigram + Bigram	0.8377	0.8375	0.8375	0.8375

The top indicative terms for fake and genuine reviews are shown in Table 4.6 Words such as smelled, turned, and finally are strong indicators of fake reviews, while terms like star, world, and concierge point to genuine reviews. Similar to Naive Bayes, fake reviews tend to use subjective or narrative language, whereas genuine reviews include more concrete and factual references.z

Table 4.6. Top Indicative Terms (Logistic Regression)

Fake Terms	Importance	Corr.	Genuine Terms	Importance	Corr.
smelled	1.56	0.17	star	-1.67	-0.13
turned	1.36	0.16	world	-1.50	-0.09
finally	1.33	0.21	concierge	-1.30	-0.07
millennium	1.25	0.19	street	-1.09	-0.11
chicago	1.23	0.42	elevator	-1.07	-0.20

4.4 Random Forest

The performance of the **Random Forest classifier** was evaluated under two feature configurations: Unigram and Unigram+Bigram. Both models were trained using Folds 1–4 and tested on Fold 5, following the same fold-based evaluation framework described previously. The Out-of-Bag (OOB) estimate was used during training to approximate generalization without requiring further data partitioning.

Model Performance **Unigram Model.** The Random Forest model trained with uni-gram features achieved a test accuracy of **0.8562**, demonstrating robust discrimination between truthful and deceptive reviews. Precision, recall, and F_1-score values indicated balanced performance across both classes, with particularly strong recall for truthful instances. The detailed results are shown in Table 4.7.

Model	**Accuracy**	**Precision**	**Recall**	**F_1-score**
RF (Unigrams)	0.8562	0.8202	0.9125	0.8639

Table 4.7. Performance metrics for the Random Forest classifier using unigram features. Out-of-Bag estimate: **0.8469**.

The classification report indicated a precision of **0.8202** and recall of **0.9125** for truthful reviews, suggesting that the model captured authentic linguistic signals with high sensitivity. Conversely, deceptive reviews yielded a slightly higher precision (**0.9014**) but lower recall (**0.8000**), reflecting a trade off between false positives and false negatives that aligns with known challenges in deception detection.

Unigram+Bigram Model. Extending the vocabulary to include bigram features resulted in a test accuracy of **0.8250**. As shown in Table 4.8, this configuration achieved slightly lower performance across most metrics, indicating that the inclusion of local contextual patterns did not yield further improvements for this dataset.

Model	**Accuracy**	**Precision**	**Recall**	**F_1-score**
RF (Unigrams+Bigrams)	0.8250	0.7889	0.8875	0.8353

Table 4.8. Performance metrics for the Random Forest classifier using combined unigram and bigram features.

Hyperparameter Summary For both variants, model selection relied on the Out-of-Bag (OOB) score as the principal indicator of generalization. The optimized parameter grid is summarized below:

$$
\begin{aligned}
\texttt{n\_estimators} &\in \{300, 600\},\\
\texttt{max\_features} &\in \{\texttt{"sqrt"}, 0.1\},\\
\texttt{max\_depth} &\in \{\texttt{None}, 40\},\\
\texttt{min\_samples\_leaf} &\in \{1, 2\},\\
\texttt{bootstrap} &= \texttt{True}.
\end{aligned}
$$

This parameterization balances model complexity and ensemble diversity, mitigating overfitting by constraining feature selection at each split while maintaining strong representational capacity through a large number of trees.

Interpretation of Feature Importance Feature importance analysis has revealed distinctive lexical indicators associated with deceptive and truthful reviews (Table 4.9). Deceptive texts tended to include experiential and self-referential terms (for example, "*chicago*," "*my*," "*decided*"), while truthful reviews were characterized by situational and descriptive terms (for example, "*location*," "*great*," "*floor*"). These findings align with psycholinguistic evidence suggesting that deceptive writing often relies on personal narrative framing and affective wording, whereas truthful writing is more externally descriptive and concrete.

Type	Term	Importance	Direction (P1–P0)
Deceptive	chicago	0.0402	−0.416
Deceptive	my	0.0122	−0.194
Deceptive	decided	0.0081	−0.134
Truthful	location	0.0136	+0.200
Truthful	great	0.0069	+0.147
Truthful	floor	0.0039	+0.106

Table 4.9. Representative top terms contributing to deceptive vs. truthful classification in the Random Forest model (unigram configuration).

4.5 Gradient Boosting

Gradient Boosting was evaluated using stratified 5-fold cross-validation for hyperparameter tuning and an independent test fold (Fold 5) for final validation. This procedure ensured consistent, non-overlapping data partitions between training and testing, therefore allowing for a fair estimation of generalization performance. The model evaluation was based on

accuracy, precision, recall, and F_1 score.

After tuning, both unigram and unigram+bigram feature representations we obtained competitive results, with the latter achieving slightly higher overall performance.

Table 4.10 summarizes the evaluation metrics on the test fold for both configurations.

Table 4.10. Gradient Boosting Classifier Performance

Model	Accuracy	Precision	Recall	F1-score	CV_ACC
Unigram	0.8656	0.8545	0.8812	0.8677	0.8711
Unigram + Bigram	0.8719	0.8696	0.8750	0.8723	0.8812

The unigram+bigram model slightly outperformed the unigram-only configuration across all metrics, achieving an accuracy of 87.2% and an F_1 score of 0.8723, compared to 86.6% accuracy and 0.8677 F_1 for the unigram model. However, a McNemar test comparing the two models yielded b=9, c=11, and p=0.8238, indicating that the performance difference was not statistically significant. Both models therefore perform comparably well, with the unigram+bigram configuration showing a modest, but non-significant, edge.

Feature directionality was analyzed to identify lexical cues that most strongly distinguish between deceptive and truthful reviews. The top features for each model are reported in Table 4.11, ranked by the difference in feature presence between truthful (P1) and deceptive (P0) texts. Positive values indicate truthful-leaning terms, while negative values indicate deceptive-leaning ones.

Table 4.11. Top Indicative Terms (Gradient Boosting Classifier)

Unigram Model		Unigram + Bigram Model	
Fake Terms	Direction(P1–P0)	Fake Terms	Direction(P1–P0)
chicago	-0.334	chicago	-0.334
my	-0.234	my	-0.234
experience	-0.127	at the	-0.134
at	-0.103	experience	-0.127
luxury	-0.098	my room	-0.111
Genuine Terms	Direction_g	Genuine Terms	Direction_g
location	0.248	location	0.248
we	0.181	we	0.181
great	0.147	great	0.147
floor	0.144	floor	0.144
michigan	0.122	michigan	0.122

Across both models, deceptive reviews tended to use more personal and experiential language (for example : *my*, *experience*, *luxury*), whereas truthful reviews were more likely to contain specific spatial or descriptive terms (for example: *location*, *floor*, *michigan*). This pattern aligns with prior findings that deceptive writing often emphasizes self-reference and affect, while truthful reviews rely on concrete, verifiable details.

In summary, Gradient Boosting achieved a robust and stable performance across both feature representations. While unigram+bigram features provided a small gain in all evaluation metrics, the improvement was not statistically significant. The model's interpretable term associations further highlight linguistic tendencies that differentiate deceptive from truthful opinion texts.

5 Discussion

5.1 Simple Classification Tree

The Decision Tree classifier achieved a moderate accuracy of around 68%. Although the model has a limited capacity to capture linguistic patterns, it offers an interpretable decision-making structure that allows direct examination of which words contribute most to the classification. This interpretability makes it a useful baseline before applying more complex ensemble or linear models.

The Decision Tree was trained once using unigram features and again with bigram features added to the same unigram representation. However, adding bigram features did not improve model performance. The accuracy decreased slightly from 68.1% for the unigram model to 65% for the unigram and bigram model. This suggests that for sparse textual data, increasing feature dimensionality can hurt performance in simple models like Decision Trees. As the tree partitions feature space based on discrete word presence, the introduction of bigrams expanded the vocabulary without necessarily adding discriminative power. In this case, the unigram-only model generalized better, while the unigram+bigram model may have introduced slight overfitting and reduced interpretability.

The confusion matrices 8.1 and 8.2 further illustrate this effect: the unigram model correctly classified more deceptive reviews (53 vs. 47) and fewer truthful ones as deceptive, resulting in a more balanced performance. In contrast, the unigram+bigram model showed increased confusion between the two classes, reinforcing that adding bigrams did not improve discriminative power.

Feature and correlation analysis revealed that words such as *chicago*, *decided*, *turned*, *east*, and *finally* have strong positive correlations with fake or deceptive reviews. This indicates that deceptive writers often include vivid descriptions and location-related storytelling to sound authentic. In contrast, the word *called* is negatively correlated with fake and occurs more often in truthful reviews, reflecting concrete interactions typical of genuine customer experiences. Interestingly, the feature analysis revealed that only one term (*called*) showed a strong negative correlation with fake reviews. This may suggest that the Decision Tree primarily relied on features associated with deceptive writing, while fewer features were indicative of truthful reviews. It may also imply that deceptive reviews share

more consistent linguistic cues, whereas truthful reviews exhibit greater lexical variety.

5.2 Multinomial Naive Bayes

The experimental results offer several key insights into the linguistic characteristics of deceptive versus truthful reviews and the effectiveness of different feature sets for this classification task.

The simpler unigram-only model gives slightly superior performance on the test set when compared to the more complex unigram+bigram model. This suggests that for the Multinomial Naive Bayes classifier, increasing feature complexity by including bigrams does not necessarily lead to better generalization. A possible explanation for this is the risk of overfitting, where the model learns patterns specific to the training data, such as idiosyncratic two-word phrases, that do not generalize well to unseen reviews. Furthermore, the inclusion of bigrams significantly increases feature sparsity. Many bigrams appear infrequently, which can introduce noise and make it difficult for a probabilistic model like Naive Bayes to estimate their true predictive value reliably.

The feature analysis illuminates the linguistic differences between the two classes of reviews. Fake-indicative terms consistently point towards a more narrative and descriptive style. The unigram model identified words like relax, luxury, and originally, which aim to create a sense of atmosphere rather than state facts. The unigram+bigram model reinforced this by highlighting the frequent use of proper nouns and location names, such as chicago millennium and sheraton chicago. This suggests that deceptive reviews may be constructed by writers attempting to sound authentic through the deliberate inclusion of specific-sounding but generic details. In contrast, genuine-indicative terms are characterized by their concreteness and focus on functional details. Words like priceline, sofa, fridge, and the number 25 consistently emerged as strong predictors of truthful reviews. These terms reflect a focus on the practical aspects of a hotel stay—the booking process, room amenities, and specific costs. This linguistic pattern suggests that genuine reviewers are recounting an actual experience by recalling tangible details, rather than constructing a story.

5.3 Logistic Regression with Lasso Penalty

The experimental results from the Logistic Regression classifier, which utilized an L1 (Lasso) penalty, offer several key insights into the linguistic characteristics of deceptive

versus truthful reviews and the effectiveness of different feature sets for this classification task.

The simpler unigram-only model demonstrated slightly superior performance on the independent test set when compared to the more complex unigram+bigram model, achieving an accuracy of 85.0% versus 83.8%. This outcome suggests that for a linear model like Logistic Regression, even with the feature-selecting properties of L1 regularization, increasing feature complexity does not necessarily lead to better generalization. A possible explanation is that the inclusion of numerous, often sparse, bigram features increases the model's dimensionality to a point where it may overfit to noise within the training data, despite the regularization. The strong performance of the unigram model indicates that individual keywords already provide a robust signal for classification, and the marginal contextual information from bigrams is outweighed by the added complexity and noise.

The feature analysis illuminates the linguistic differences between the two classes of reviews. Fake-indicative terms consistently point towards a more narrative and descriptive style. The unigram model identified words like smelled, turned, and finally, which suggest a focus on recounting a specific story arc. The unigram+bigram model reinforced this by highlighting the bigram hotel chicago, suggesting that deceptive reviews may be constructed by writers attempting to sound authentic through the deliberate inclusion of specific-sounding but generic details. In contrast, genuine-indicative terms are characterized by their concreteness and focus on functional details. Words like star (likely referring to hotel ratings), concierge, and elevator consistently emerged as strong predictors of truthful reviews. The unigram+bigram model added the highly specific phrase booked hotel, which points directly to the process of making a reservation. This linguistic pattern suggests that genuine reviewers are recounting an actual experience by recalling tangible details and processes, rather than constructing a story.

5.4 Random Forest

Random Forest has achieved a high and stable performance across both feature representations. Using unigram features, it reached an accuracy of **85.6%**, with precision of **0.8202**, recall of **0.9125**, and an F_1-score of **0.8639**. When the bigram features were added, the performance slightly decreased, with a final accuracy of **82.5%**, precision of **0.7889**, recall of **0.8875**, and F_1-score of **0.8353**. Overall, the Random Forest demonstrated strong generalization, performing consistently well across both variants.

The Random Forest model combines multiple decision trees trained on random subsets of the data and features. This structure helps reduce variance while maintaining predictive strength, allowing the model to capture both frequent and subtle lexical patterns that separate truthful from deceptive writing. The confusion matrices show that both versions achieved balanced performance, with the unigram model correctly identifying most truthful reviews and maintaining relatively low false positive rates for deceptive ones.

The unigram-only model outperformed the unigram+bigram variant. Although bigrams can capture a short contextual expressions, such as "*very good*" or "*my stay*," they also increase the feature space and may introduce redundancy. This likely led to mild overfitting and reduced performance in the more complex model. In this dataset, single-word features already provided enough linguistic variation for accurate classification, suggesting that adding local context did not yield substantial benefits.

Feature importance analysis supports this interpretation. The most influential terms for deceptive reviews included *chicago*, *my*, and *decided*, which tend to appear in more personal or narrative writing. These words do suggest that deceptive reviewers often focus too much on personal experiences or emotional storytelling to appear genuine. In contrast, truthful reviews were usually characterized by concrete and descriptive terms such as *location*, *great*, and *floor*, which describe tangible aspects. This difference indicates that truthful reviews emphasize verifiable details, while deceptive ones rely more on subjective or affective descriptions.

5.5 Gradient Boosting

The Gradient Boosting shows strong and consistent results, performing above the single Decision Tree and close to the Random Forest model. With unigram features, it reached an accuracy of **86.6%**, precision of **0.8545**, and recall of **0.8812**. When bigram features were added, the accuracy increased slightly to **87.2%**, with precision and recall values of **0.8696** and **0.8750**. Cross-validated accuracy values around **0.87–0.88** confirmed that the model has generalized well across folds.

Gradient Boosting builds its predictions step by step, where each new tree focuses on correcting the errors made by the previous ones. This iterative process allows the model to adjust its boundaries gradually, capturing smaller lexical differences that may signal deceptive writing. The confusion matrices in Figure 8.4 show that both versions of the

model achieved balanced performance, with only a few misclassifications in each class.

The comparison between unigram and unigram+bigram variants shows only a small difference. While the inclusion of bigrams helped the model capture short patterns such as "*my room*" or "*at the*," most of the discriminative information was already present in single-word features. The McNemar test ($b = 9, c = 11, n = 20, p = 0.824$) confirmed that the difference was not statistically significant, suggesting that adding more features did not lead to a meaningful improvement and slightly increased model complexity.

The feature importance results offer a clearer view of what the model learned. Across both configurations, deceptive reviews often included more narrative or self-focused terms such as *chicago*, *my*, and *experience*, which may reflect an attempt to sound more vivid or convincing. Truthful reviews, on the other hand, contained more specific and descriptive words such as *location*, *floor*, and *michigan*, referring to the observable details rather than subjective impressions. These patterns suggest that the model effectively picked up on linguistic tendencies known to differentiate deceptive from truthful writing.

5.6 How does the performance of the generative linear model (multinomial naive Bayes) compare to the discriminative linear model (regularized logistic regression)?

Based on the test set results, the generative Multinomial Naive Bayes model (F1 score 0.8875) performed better than the discriminative regularized Logistic Regression model (F1 score v). This difference arises may because Naive Bayes, as a generative model, estimates class-conditional probabilities and handles high-dimensional sparse text features effectively. In contrast, Logistic Regression can be more sensitive to noise and tend to overfit when the data are sparse.

5.7 Are Random Forests and Gradient Boosting able to improve on the performance of the linear classifiers?

For Both Random Forest and Gradient Boosting performed better than the linear models, especially Logistic Regression. While the linear classifier captured some general word-level patterns, it was limited by its single decision boundary and struggled to model non-linear or interactive relationships between features.

The tree-based ensembles handled these patterns more effectively. Random Forest reduced noise by averaging many smaller trees, and Gradient Boosting refined its predictions step by step, learning from the remaining errors of each stage. This allowed both models to recognize more complex linguistic cues that linear models tend to overlook.

Overall, the models achieved higher accuracy and F_1-scores (around **85–87 (percent)**) compared to the linear classifier, showing that non-linear ensemble methods can better capture the subtle variations present in deceptive and truthful writing.

5.8 Does performance improve by adding bigram features instead of using just unigrams?

In the case of the Decision Tree, adding bigram features did not improve performance over the unigram model, as described in Section 5.1. Although bigrams can capture short contextual cues such as negations or common phrases, the limited dataset size may have made these features too sparse, potentially leading to overfitting. Consequently, the unigram representation provided a better trade-off between feature richness and generalization.

Based on the results discussed in Sections 5.2 and 5.3, adding bigram features does not significantly improve model perfromance. Both Multinomial Naive Bayes and Logistic Regression achieved slightly better accuracy with unigram-only features. This suggests that the additional complexity introduced by bigrams may increase sparsity and noise, leading to marginal or no improvement in generalization, while unigrams already capture the most informative cues for distinguishing deceptive and truthful reviews.

For Random Forest Adding bigram features did not improve performance. The unigram model reached an accuracy of 0.8562 with an F_1-score of 0.8639, while the unigram+bigram model dropped to 0.8250 accuracy and an F_1 of 0.8353. This decrease suggests that expanding the feature space introduced additional sparsity and more redundancy, making the model more prone to a mild overfitting. In this dataset, single-word features already captured the most informative lexical patterns, and the added contextual information from bigrams did not contribute in any meaningful way to the classification.

For Gradient Boosting The addition of bigrams produced only a small improvement, which was not statistically significant, however the unigram model achieved an accuracy of 0.8656 and an F_1 of 0.8677, while the unigram+bigram model slightly increased to 0.8719 accuracy and an F_1 of 0.8723 (CV_ACC $\approx$ 0.8812). A McNemar test (b=9, c=11, p=0.824) confirmed that the difference between both models was not meaningful. This shows that while Gradient Boosting can incorporate short contextual patterns, most of the useful signal is already captured by unigrams.

5.9 What are the five most important terms (features) pointing towards a fake review, and what are the five most important terms pointing towards a genuine review?

1. **Simple Classification Tree:**
   - Top 5 Fake-indicative terms: chicago, decided, turned, east, finally.
   - Top 5 Genuine-indicative terms: called.
2. **Multinomial Naive Bayes:**
   - Top 5 Fake-indicative terms: relax, originally, luxury, settled, egg.
   - Top 5 Genuine-indicative terms: priceline, sofa, 25, fridge, stated.
3. **Logistic Regression (L1 penalty):**
   - Top 5 Fake-indicative terms: smelled, turned, finally, millennium, chicago.
   - Top 5 Genuine-indicative terms: star, world, concierge, street, elevator.
4. **Random Forest:**

- Top 5 Fake-indicative terms: chicago, my, decided, finally, experience.
- Top 5 Genuine-indicative terms: location, great, floor, is, are.

5. **Gradient Boosting:**

- Top 5 Fake-indicative terms: chicago, my, experience, at the, my room.
- Top 5 Genuine-indicative terms: location, we, great, floor, michigan.

5.10 Statistical Test

To compare the performance of the five models, a statistical significance test was conducted specifically, the **pairwise McNemar test**since all models were evaluated on the same test set (Fold 5). This test was chosen to determine whether the observed differences in performance between models were statistically significant. The McNemar test is a *non-parametric* test that evaluates whether two classifiers differ significantly in their proportions of correct and incorrect predictions on the same dataset. A similar approach was adopted by Rainio *et al.* [5] for comparing their CNN-based models.

The results of the pairwise McNemar tests (Table 5.3) revealed that the **Single Classification Tree** performed **significantly worse than all other models** ($p < 0.001$ after Bonferroni correction) summarized in Table 5.1 and Interactive McNemar comparison matrix (HTML version) . This indicates that its lower accuracy is not due to random variation but reflects a genuine difference in predictive performance.

Among the remaining four models**Logistic Regression**, **Multinomial Naïve Bayes**, **Random Forest**, and **Gradient Boosting**—no statistically significant differences were observed (all $p > 0.05$).

These findings are consistent with the accuracy values summarized in Table 8.3. The Decision Tree exhibits a substantially lower accuracy (0.681), representing a statistically significant drop in performance compared to the other four models, which achieved accuracies between 0.85 and 0.8875. This suggests that, while the Multinomial Naïve Bayes model achieved the highest numerical accuracy, its advantage over the other high-performing models is *not statistically significant*, indicating that the four top models perform comparably well on this dataset.

Table 5.1. Pairwise McNemar comparison (✓= row better, ✗= worse, · = no significant difference; $\alpha = 0.001$).

	Naive Bayes	**Logistic Regression**	**Gradient Boosting**	**Random Forest**	**Decision Tree**
Naive Bayes	–	·	·	·	✓
Logistic Regression	·	–	·	·	✓
Gradient Boosting	·	·	–	·	✓
Random Forest	·	·	·	–	✓
Decision Tree	✗	✗	✗	✗	–

Table 5.2. Pairwise McNemar Test p-values between models (Bonferroni-corrected $\alpha = 0.001$).

Table 5.3. Pairwise McNemar Test p-values between models (Bonferroni-corrected $\alpha = 0.001$).

Model	**Naive Bayes**	**Logistic Regression**	**Gradient Boosting**	**Random Forest**	**Decision Tree**
Naive Bayes	0	0.3268	0.8597	0.7353	$\mathbf{1.6 \times 10^{-5}}$
Logistic Regression	0.3268	0	0.6171	0.7604	**0.000 27**
Gradient Boosting	0.8597	0.6171	0	1.0000	$\mathbf{2.66 \times 10^{-5}}$
Random Forest	0.7353	0.7604	1.0000	0	**0.000 29**
Decision Tree	$\mathbf{1.6 \times 10^{-5}}$	**0.000 27**	$\mathbf{2.66 \times 10^{-5}}$	**0.000 29**	0

6 Conclusion

The Multinomial Naive Bayes achieved the strongest single-model performance on the held-out test fold ($F_1 = 0.8875$; Acc. $= 0.8875$), slightly outperforming the discriminative linear baseline with L1-regularized Logistic Regression ($F_1 \approx 0.85$; Acc. $= 0.85$). Among the tree ensembles, Gradient Boosting was competitive (best Acc. $= 0.8719$; $F_1 = 0.8723$, CV_ACC ≈ 0.88), and Random Forest achieved high accuracy as well (Acc. $= 0.8562$; $F_1 = 0.8639$ with unigrams). In short, Naive Bayes led, Gradient Boosting was a close second, and Random Forest exceeded the Decision Tree and Logistic Regression baselines.

Adding bigrams did not consistently improve results. For Random Forest, unigram features outperformed the unigram+bigram variant. For Gradient Boosting, bigrams offered only a small gain that was *not* statistically significant. For Naive Bayes and Logistic Regression, unigram-only models remained slightly stronger. Taken together, single-word features captured most of the discriminative signal in this dataset; expanding the n-gram window increased sparsity and complexity without reliable gains.

From the Pairwise McNemar test, it was confirmed that the Single Classification Tree performed significantly worse than all other models ($p < 0.001$, Bonferroni-corrected), while differences among the top four classifiers were not statistically significant ($p > 0.05$).

Across methods, the most informative terms were consistent. Deception-leaning features included *chicago*, *my*, *decided*/*experience*, and short phrasal variants (e.g., *at the*, *my room*), suggesting narrative, self-focused, or atmosphere-building language. Genuine-leaning features emphasized concrete, verifiable details such as *location*, *floor*, *michigan*, *we*, and quality descriptors like *great*. This split—subjective narrative vs. externally verifiable detail—was stable across Naive Bayes, Logistic Regression, Random Forest, and Gradient Boosting.

7 References

1. M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, *"Finding deceptive opinion spam by any stretch of the imagination,"* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 309–319, 2011.

2. M. Ott, C. Cardie, and J. T. Hancock, *"Negative deceptive opinion spam,"* Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 497–501, 2013.

3. Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy, "Towards a general rule for identifying deceptive opinion spam," ACL 2014, pp. 1566–1576.

4. A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance, *"What Yelp fake review filter might be doing?"* Proceedings of the International Conference on Web and Social Media (ICWSM), pp. 409–418, 2013.

5. Rainio, O., Teuho, J., and Klén, R., *"Evaluation metrics and statistical tests for machine learning."* Scientific Reports, vol. 14, p. 6086, 2024. `https://doi.org/10.1038/s41598-024-56706-x`

8 Appendix

Table 8.1. Best Hyperparameters (Naive Bayes)

Model	α	Max Features	F1
Unigram	1	3000	0.8601
Unigram + Bigram	0.5	3000	0.8724

Table 8.2. Best Hyperparameters (Logistic Regression)

Model	C	Max Features	F1
Unigram	1	1000	0.8165
Unigram + Bigram	1	2000	0.8204

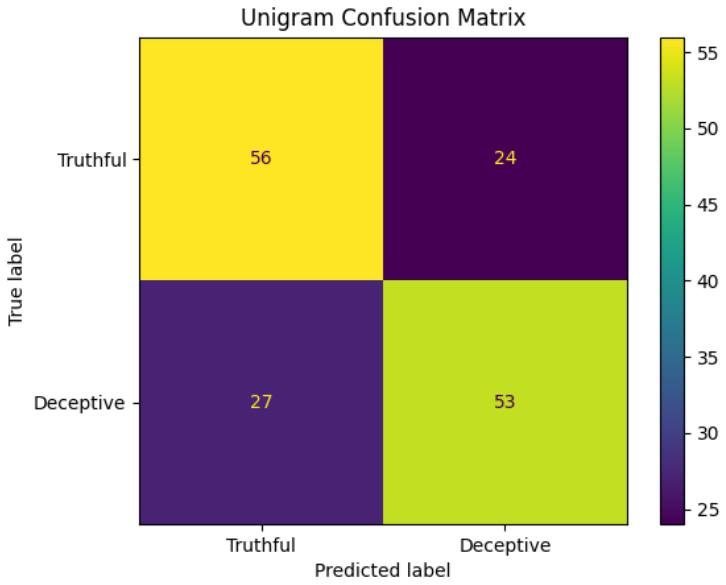

Figure 8.1. Confusion matrix for Unigram Classification Tree.

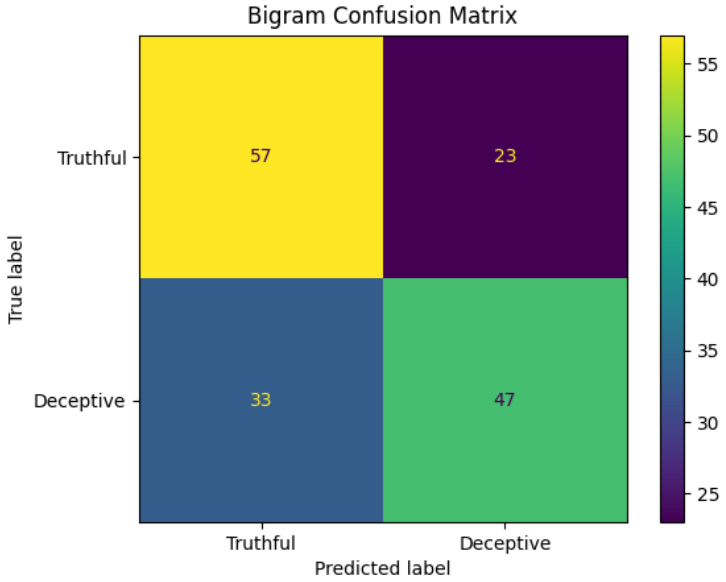

Figure 8.2. Confusion matrix for Unigram and Bigram Classification Tree.

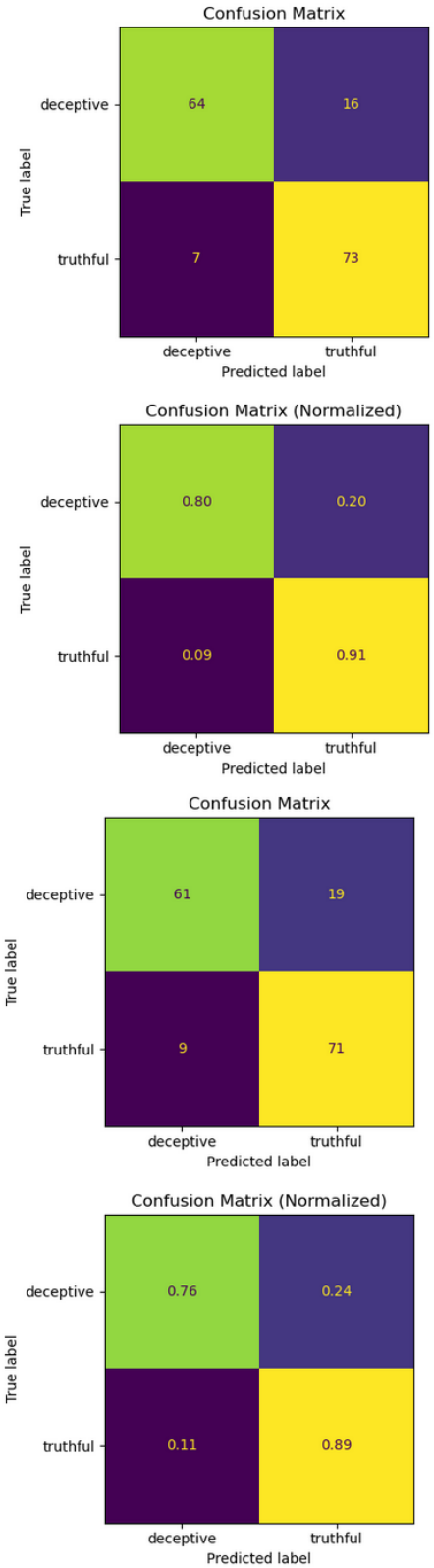

Figure 8.3. Confusion matrices for the **Random Forest classifier** using unigram (top) and unigram+bigram (bottom) representations. The model shows high recall for truthful reviews and balanced accuracy across both classes, with slightly reduced performance when bigrams are included.

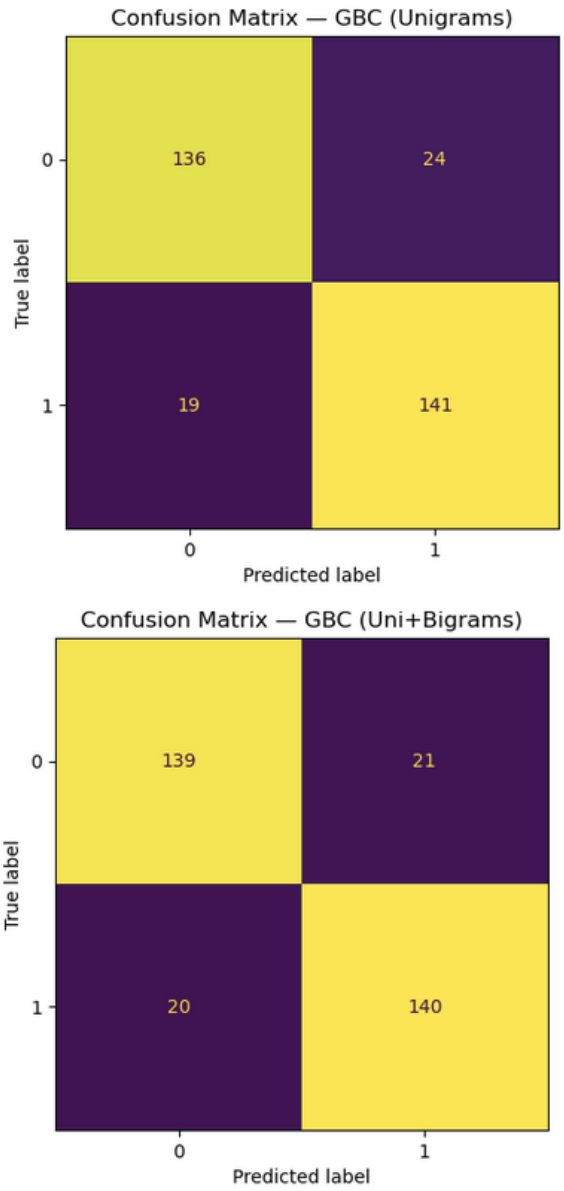

Figure 8.4. Confusion matrices for the **Gradient Boosting classifier** using unigram (top) and unigram+bigram (bottom) representations. Both models demonstrate balanced classification with very few misclassifications, confirming strong generalization and stability across feature configurations.

Table 8.3. Accuracy of the Best Performing Models

Model	**Accuracy**
Logistic Regression	0.8500
Multinomial Naïve Bayes	**0.8875**
Single Classification Tree	0.6810
Random Forest	0.8562
Gradient Boosting	0.8719