IDS703 - Final Project Report - Customer Satisfaction Classification

Performance Evaluation Using Real and Synthetic Data

Overview

This report presents the development and evaluation of a customer satisfaction classification model using BERT, a pre-trained NLP model. The project evaluates the model's performance on both synthetic and real datasets to understand its ability to classify customer feedback sentiment accurately. Through quantitative and qualitative analyses, the report highlights the strengths and limitations of real-world data, focusing on the challenges of capturing nuanced patterns and variability inherent in customer reviews. The findings demonstrate the practical significance of real data for robust applications and emphasize the limitations of synthetic data as a baseline for testing.

Introduction

Customer satisfaction classification is critical for understanding user feedback and improving business decisions. By analyzing customer reviews, businesses can identify areas of improvement and enhance user experiences. This project focuses on:

- Fine-tune a BERT model for binary classification (recommend or not recommend).
- Evaluating the model's performance on real-world customer reviews and a baseline synthetic dataset.
- Using quantitative metrics and qualitative examples to gather insights of the model's strengths, weaknesses, and potential applications.

Model Overview

- **Model Used:** BERT (Bidirectional Encoder Representations from Transformers); pre-trained on a large corpus of text and fine-tuned for binary classification.
- **Synthetic Data Generation:** Used as a controlled baseline for comparison, generated with predefined positive and negative seed words combined with random vocabulary sampling.
- **Real Data:** Sourced from an e-commerce review dataset with customer reviews labeled as "Recommend" or "Not Recommend," reflecting real-world sentiment patterns.
- **Training Process:** Fine-tuning BERT with separate train-test splits for real and synthetic datasets. Metrics such as accuracy, precision, recall, and F1-score used for evaluation.

Synthetic Data Generation

The synthetic dataset was generated to provide a controlled baseline for model evaluation. By using synthetic data, the model's ability to classify sentiment in a controlled environment was assessed before applying it to real-world reviews. The process involved the following steps:

- **Template Design:** Four distinct templates were designed to simulate common review structures, each containing placeholders for sentiment-driving words. For example, "I `[user_verb]` this product because it `[product_verb]` my expectations."

- **Sentiment Mapping:** Sentiment-driving words were selected from predefined lists of adjectives and verbs:
    - Positive adjectives: `["amazing", "great", "perfect", "excellent", "loved"]`
    - Negative adjectives: `["terrible", "bad", "poor", "horrible", "disappointing"]`
    - Positive verbs: `["love", "like", "appreciate", "enjoy"], ["exceeded", "met", "fulfilled", "impressed"]`
    - Negative verbs: `["dislike", "hate", "avoid", "regret"], ["fell short of", "failed to", "disappointed", "lacked"]`
- **Generation Process:**
    - A sentiment (`positive` or `negative`) was randomly selected for each review.
    - A random template was populated with sentiment-driving words corresponding to the selected sentiment.
    - Reviews were labeled as `1` for positive sentiment and `0` for negative sentiment.
- **Dataset Composition:** A total of 2,000 reviews were generated, equally split between positive and negative sentiments. This controlled balance ensured a neutral baseline for model evaluation.
- **Assumptions and Limitations:** The synthetic dataset reflects idealized sentiment patterns, which may not fully capture the ambiguity and variability of real-world reviews. For example, neutral or mixed-sentiment cases are absent, simplifying the task for the model.
- **Example Reviews:**
    - Positive: "The quality was amazing, and I would love to recommend it."
    - Negative: "I found this product to be disappointing, and it fell short of my needs."

Training and Validation

1. Training Setup

- **Loss Function**: For a binary classification task in this case, we employ cross entropy loss as our loss function. The model outputs logits, and the loss compares these logits with the true label.
- **Optimizer and Learning Rate**: We used AdamW as the optimizer for fine-tuning BERT, which helps apply proper weight decay and gradient updates suitable for transformers. The learning rate was set to 5×10^{-5}.
- **Epochs**: Because BERT is already trained, fewer epochs are needed. Fine-tuning often happens in 2-5 epochs. Training longer can lead to overfitting, as BERT is already highly capable of representing language. In this project, we arbitrarily chose epochs of 3.
- **Batch Size**: Common batch sizes for fine-tuning range from 8 to 32. Based on our GPU memory, we chose a batch size of 16.

2. Fine-tuning Process

We fine-tuned the entire pre-trained BERT model for this task. This means all layers of BERT, including its pre-trained weights, were updated during training. A fully connected classification layer was added on top

of BERT using the `BertForSequenceClassification` class to perform binary classification (recommend vs. not recommend). The steps for fine-tuning were as follows:

1. **Forward Pass**: Input token IDs and attention masks are fed into BERT.
2. **Compute Loss**: The model's final layer outputs logits, and you compute the cross-entropy loss against the true labels.
3. **Backward Pass**: Compute gradients of the loss with respect to all model parameters, including BERT weights and the classification head.
4. **Optimizer Step**: Update the model parameters using AdamW optimizer.
5. **Scheduler Step**: Adjust the learning rate according to a linear decay schedule.

3. Validation Process

After each epoch or at certain intervals, we run the model on a validation set to monitor performance (accuracy, F1-score, etc.). This helps detect overfitting and guides when to stop training or adjust hyperparameters.

Quantitative Evaluation

Metric	Synthetic Data	Real Data
Accuracy	1.00	0.91
Precision (Not Recommend)	1.00	0.79
Precision (Recommend)	1.00	0.94
Recall (Not Recommend)	1.00	0.73
Recall (Recommend)	1.00	0.96
F1-Score (Not Recommend)	1.00	0.76
F1-Score (Recommend)	1.00	0.95
Macro-Average F1 Score	1.00	0.85
Weighted-Average F1 Score	1.00	0.91

1. Accuracy Observations

- Synthetic data achieves perfect accuracy (1.00) due to its simplified and controlled nature.
- Real data achieves a high accuracy of 0.91 as well, demonstrating its ability to capture nuanced sentiment patterns in customer reviews.

2. Precision Observations

- Synthetic data achieves perfect precision (1.00) for both classes. This indicates that the synthetic reviews are straightforward, with clear sentiment patterns driving predictions.
- Real data exhibits lower precision for the "Not Recommend" class (0.79) due to the nuanced language in customer reviews.

3. Recall Observations

- Synthetic data's perfect recall (1.00) across classes reflects its lack of ambiguous or nuanced cases.
- Real data's recall for the "Not Recommend" class is 0.73, showing challenges in identifying all negative reviews.

4. F1-Score Observations

- Synthetic data scores perfectly (1.00) for both classes points to its controlled nature.
- Real data’s F1-scores demonstrate its struggle to balance precision and recall due to linguistic complexities.

5. Overall F1-Score Observations

- The Macro-Average F1 Score for synthetic data is 1.00, as all classes are perfectly balanced and correctly classified.
- For real data, the Macro-Average F1 Score is 0.85, reflecting the model's challenge in balancing performance across both classes.
- The Weighted-Average F1 Score for real data is 0.91, which accounts for the larger number of "Recommend" samples, showing strong overall performance on the imbalanced dataset.

6. General Observations

- Synthetic data provides an ideal testing baseline, with simplified patterns leading to perfect performance.
- Real data provides a robust and realistic evaluation of the model, exposing its limitations in handling ambiguous or subtle sentiments.

Qualitative Evaluation

1. Synthetic Data Examples

Good Performance Examples:

Review Text	True Label	Predicted Label
The quality was great, and I would like recommend it.	1	1
I found this product to be terrible, and it fell short of my needs.	0	0
I like this product because it exceeded my expectations.	1	1

- Reason for Success: The model excels due to the controlled nature of the synthetic dataset, which uses clear sentiment-driving words.

Poor Performance Examples:

- None Identified: The synthetic dataset’s controlled design and simplified patterns resulted in perfect performance.

2. Real Data Examples

Good Performance Examples:

Review Text	True Label	Predicted Label
This shirt is super comfortable and super chic it looks great and feels great.	1	1
I returned the dress because the color was not as pictured in the online photo I did not like the color.	0	0
This romper is adorable I love the long sleeves.	1	1

- Reason for Success: These reviews contained clear sentiment-driving phrases ("love," "great") and helped the model to make correct predictions.

Poor Performance Examples:

Review Text	True Label	Predicted Label
The material is awesome and its a super pretty dress it just didn't work on my body.	0	1
The shirt was cute im very petite and it ran very wide so didn't fit me well.	0	1

- Reason for Failure: Neutral and mixed-sentiment phrases confused the model, leading to misclassifications.
- Hypothesis: The model may have been biased toward positive sentiment in ambiguous cases due to training data imbalances.

Pros and Cons of the Model

1. Quality and Correctness

The model demonstrates strong performance on real-world data, achieving an accuracy of 91%. This reflects its ability to capture the nuanced patterns present in realistic customer reviews. However, slight performance differences across metrics reveal challenges inherent in handling real-world complexities.

- Strengths: High recall for the "Recommend" class (0.96) indicates the model's ability to generalize well for positive sentiment. This is particularly important for scenarios where missing a positive recommendation might be more costly.
- Limitations: Lower recall (0.73) and precision (0.79) for the "Not Recommend" class highlight challenges in recognizing nuanced or mixed-negative feedback. Reviews with ambiguous language (e.g., "not bad" or "okay") remain a key difficulty.
- Reason: Real-world data often includes noise, ambiguous language, and variability in expression that make it inherently harder to classify accurately. This leads to slightly lower overall performance compared to controlled synthetic data.

2. Data, Time, and Computational Requirements

- Real Data Complexity: Training on real data required significantly more preprocessing and computational resources. The larger dataset size, combined with variability in sentence structures and expressions, demanded careful tokenization and more epochs for convergence.

- Synthetic Data Efficiency: In contrast, synthetic data provided a computationally efficient baseline due to its smaller size and structured patterns. However, while faster to process, synthetic data cannot replicate the challenges and richness of real-world data.

3. Interpretability

- Real Data: Predictions on real data were more meaningful and aligned with human intuition. Correct classifications often corresponded to reviews containing clear sentiment-driving words (e.g., "amazing" or "terrible"). However, misclassifications were more difficult to debug due to real-world variability in tone, mixed sentiment, and rare edge cases.
- Synthetic Data: While not the focus, synthetic data served as a simplified baseline for evaluation. It allowed for easy debugging and initial model validation but lacked the complexity needed to simulate realistic reviews.

4. Additional Observations

- Bias and Noise in Real Data: Real-world datasets often contain biases (e.g., over-representation of positive reviews) and noise (e.g., spelling errors, colloquial expressions). These issues contribute to performance limitations and necessitate additional preprocessing steps or fine-tuning strategies.
- Generalization: The model's ability to generalize to unseen real-world examples shows its robustness. However, the slightly lower performance on "Not Recommend" reviews suggests that further work is needed to handle subtle or mixed feedback better.

Conclusion

This project emphasizes the critical role of real-world data in developing effective machine learning models. While synthetic data served as a useful baseline for testing, real data reflected the complexities and variability of actual customer reviews, which slightly impacted accuracy. The BERT model performed well on real data, providing meaningful insights for sentiment analysis. Moving forward, efforts should focus on refining preprocessing methods and enhancing the model’s ability to handle ambiguous and nuanced cases to better meet real-world demands.