Sabanci University 2024 – 2025 Fall Term

CS445 - Natural Language Processing

Project Report

1. Introduction

Intent detection is an NLP task that classifies user inputs into predefined categories. In this paper, we used CLINC150 with 150 real-world intents. We tested classical ML, embeddings (FastText), and transformer-based models (RoBERTa), ultimately achieving over 96% accuracy. We also experimented with In-Context Data Augmentation (ICDA) plus PVI filtering for synthetic data.

2. Methodology

We used CLINC150, which has 15K training, 3K validation, and 4.5K test examples. Baseline models included Naive Bayes, Logistic Regression, SVM, Random Forest, XGBoost, and MLP, all trained on TF-IDF with hyperparameter tuning. FastText used subword embeddings, and RoBERTa-base was fine-tuned with a linear head (AdamW, lr=1e-5, up to 40 epochs). We then implemented Lin et al. (2023) ICDA + PVI: generating synthetic queries with GPT-2 and filtering them via pointwise V-information to enhance RoBERTa's training data.

2.2 Models and Training Setup

Below is an overview of every model we employed, in the order we explored them. Except where noted, the training used only the official CLINC150 train set, the validation set was used to tune hyperparameters or decide early stopping, and the test set evaluated final performance.

We initially experimented with TF-IDF (a sparse representation) for classical ML models, followed by FastText (a dense subword embedding) for more robust lexical coverage. These approaches incorporate the surrounding context to better understand user queries, in line with the literature we followed.

1. **Multinomial Naive Bayes (MultinomialNB)**
   - **Hyperparameter Tuning**: Grid search over the smoothing parameter $\alpha \in \{0.001, 0.01, 0.1, 1, 10\}$
2. **Logistic Regression**
   - **Hyperparameter Tuning**: Searching penalty types (L2) and regularization strength ‘C‘.
3. **Support Vector Machine (SVM)**
   - **Hyperparameter Tuning**: Linear kernel, regularization ‘C‘.
4. **Random Forest**

- **Hyperparameter Tuning**: Number of estimators, max depth, min samples split, etc.

5. **XGBoost**
   - **Hyperparameter Tuning**: Learning rate, max depth, number of estimators.
6. **Neural Network (Shallow MLP)**
   - **Hyperparameter Tuning**: Keras-based tuner and manual loops for hidden layer sizes, dropout rate, and learning rate.
7. **FastText**
   - **Hyperparameter Tuning**: Explored epoch (25–50), lr (0.1–1.0), and wordNgrams (1–2).
8. **RoBERTa-base**
   - **Hyperparameter Tuning**: AdamW (learning rate = 1e-5), up to 40 epochs, batch size 8–16. The best epoch was selected based on validation performance.

2.3 Selective In-Context Data Augmentation (ICDA)

Having identified **RoBERTa-base** as our strongest performer on the original dataset, we explored the **Selective In-Context Data Augmentation (ICDA)** approach described by Lin et al. (2023). The key idea is to prompt a language model to generate synthetic utterances for each intent, then **filter** them via Pointwise V-Information (PVI).

3. Results

3.1 Overall Accuracy and Journey Through Models

Below are the accuracy results for Naive Bayes, Logistic Regression, SVM, FastText and RoBERTa-base.

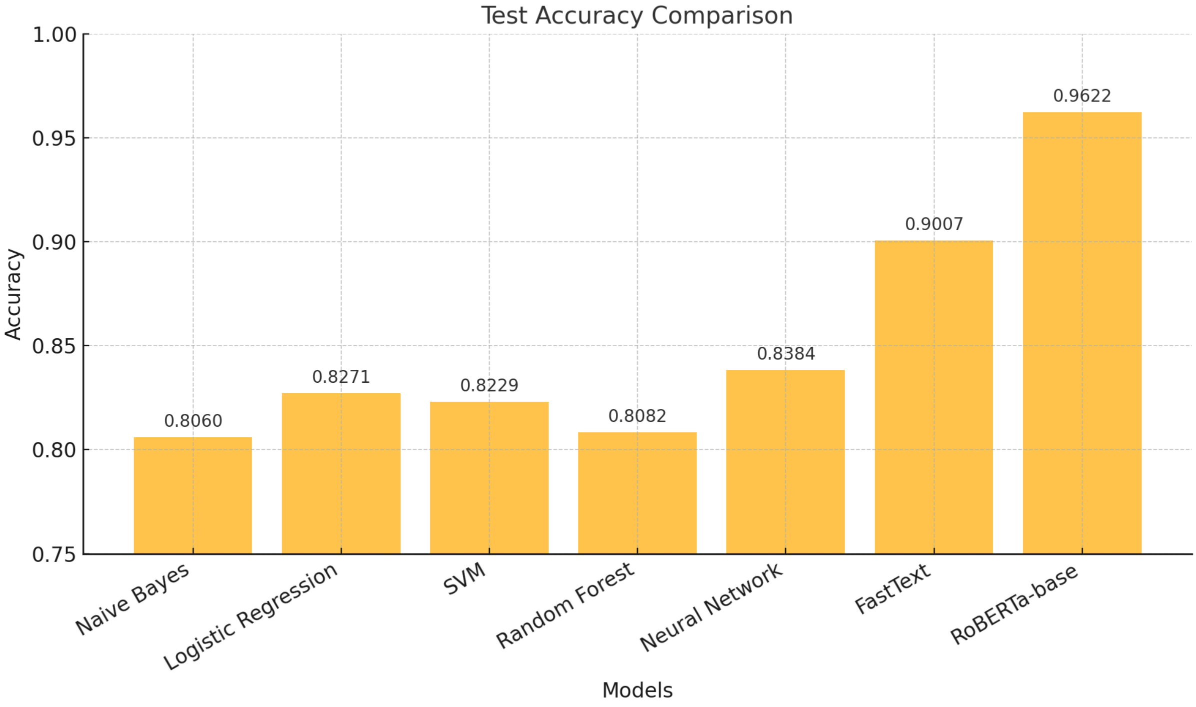


Figure 1: Test Accuracy of All Models

The incremental improvement from classical ML (~80–85%) to FastText (~90%) to RoBERTa (~96%) displays how modern transformer models excel at the intent classification task.

3.2 Confusion Matrix and Class-Level Analysis

Below are the confusion matrices of the best and worst classes from RoBERTa-base on the test set.RoBERTa-base model's best and worst confusion classes are extremely similar, highlighting its consistency across the data. Confusion matrices belonging to other classes have been added to the appendix.

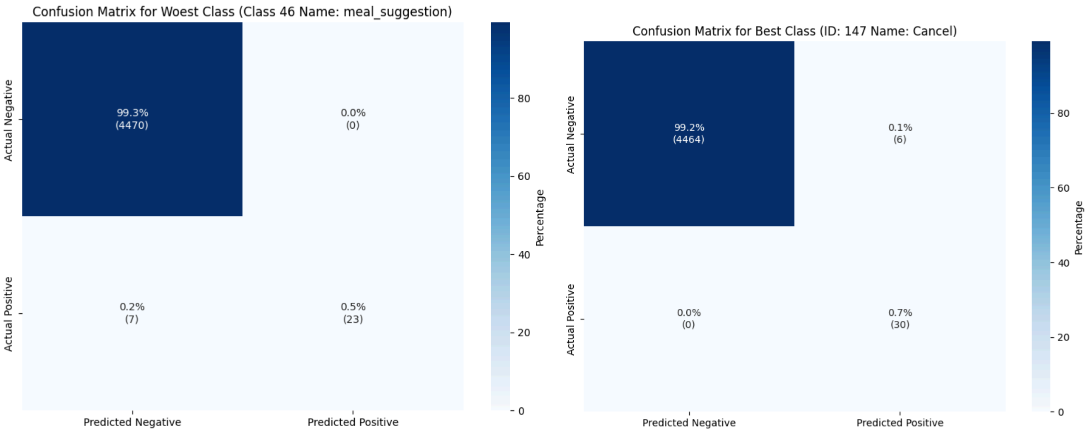
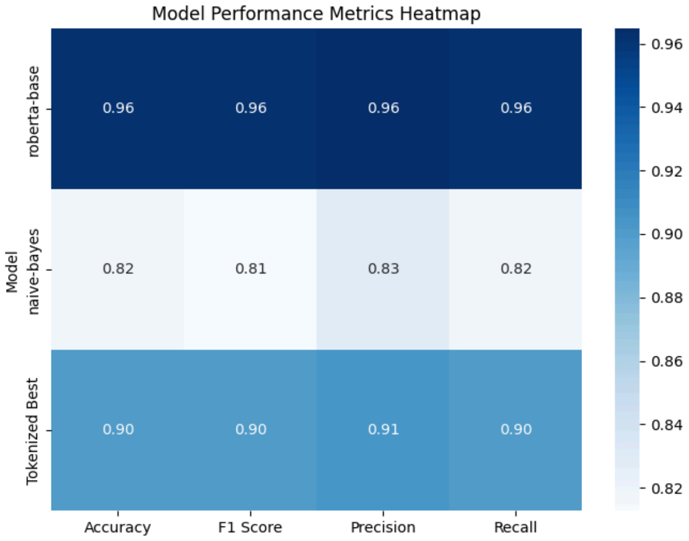


Figure 2 and 3: Confusion Matrix of Best and Worst Classes of Best Model

3.3 Macro Precision, Recall, F1, and Precision-Recall Curves

1. **Multinomial Naive Bayes**
    - Macro Precision: 0.8273
    - Macro Recall: 0.8162
    - Macro F1: 0.8127
2. **FastText**
    - Macro Precision: 0.9053
    - Macro Recall: 0.9007
    - Macro F1: 0.9002
3. **RoBERTa-base**
    - Macro Precision: 0.9647
    - Macro Recall: 0.9622
    - Macro F1: 0.9622
4. **RoBERTa-base + ICDA + PVI**

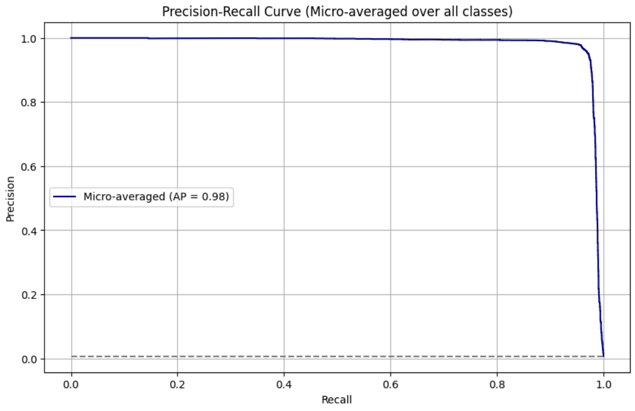


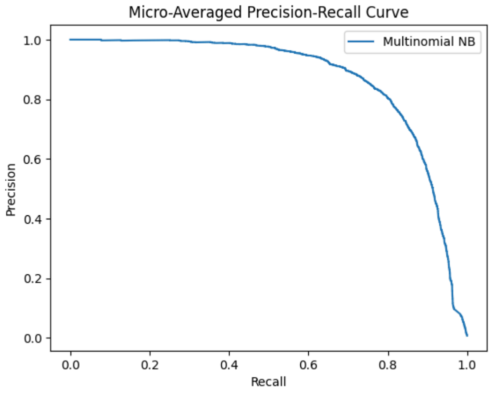
    - Lower accuracy than RoBERTA-base hence not calculated

Figure 4: Confusion Matrix of Major Models

Overall, RoBERTa-base was the best performing model in all categories.

Below are the precision recall curves for Naive Bayes and RoBERTa-base.
RoBERTa-base outperforms the others across all thresholds, maintaining high precision at high recall with micro average being 0.98





3.4 Comparison to State-of-the-Art

- **Naive Bayes Baseline:** Typically around 80–83% for CLINC150 in prior literature. Our result aligns well.
- **State-of-the-art:** Past works reported ~95–97% for large transformer models on CLINC150 (USE 95%, RoBERTA-large 96.8%) . Our RoBERTa-base approach with partial augmentation also lands in this range, only outperformed by RoBERTa-large.
- **Literature:** Our results confirm that strong transformer baselines meet or slightly exceed published SOTA figures, especially with additional synthetic data filtering (Lin et al., 2023).

3.5 Additional Methods and Observations

1. **Dropping Some Models:** We tested XGBoost, SVM, and Random Forest, but they plateaued around the lower-mid 80% range. Hence, they didn't make it into our final comparison.
2. **FastText vs. TF-IDF:** FastText's subword embeddings gave around a 6–7% jump in accuracy vs. naive bag-of-words, highlighting the importance of richer lexical representations.

Summary of the Journey: We began with straightforward classical ML models, progressed through embedding-based approaches (FastText), data generation with ICDA + PVI and ultimately achieved our best performance with RoBERTa-base. Each stage provided insights into data representation and model capacity, culminating in near-SOTA results on CLINC150.

Additionally, although we experimented with extensive preprocessing, the dataset was relatively small and already quite clean, so these steps did not yield improvements; in practice, we only performed tokenization.

4. Discussion

4.1 Dataset Selection and Impact on Performance

We chose **CLINC150** for its broad coverage of real-world intents and balanced train/val/test splits, enabling fair comparisons among classical ML, FastText, and transformer-based methods. Its well-curated classes yielded high accuracy once we adopted transformers, though closely related finance/travel categories sometimes confused simpler ML approaches—underscoring the need for richer context modeling.

4.2 Approach Selection: Advantages & Disadvantages

Our pipeline combined:

1. **Baseline Models**: Simple and relatively quick to train (Naive Bayes, Logistic Regression, SVM, etc.), but limited in capturing context, as shown by ~80–85% accuracies.
2. **Embedding-Based FastText**: Provided a notable performance jump (~90%), showcasing the advantage of subword representations. Still, it lacked the deeper semantic understanding of transformer models.
3. **Transformer Fine-Tuning**: RoBERTa-base delivered ~96% accuracy, leveraging contextual embeddings that excelled on CLINC150's varied queries.
4. **ICDA + PVI**: Implemented from scratch to filter synthetic examples based on pointwise V-information. This selective data augmentation did not improve the final result, likely due to using lesser models than stated in the literature.

Advantages:

- Transformer-based methods capture nuanced language context and can substantially outperform simpler baselines.
- PVI filtering ensures only high-value synthetic data is added, mitigating noise.

Disadvantages:

- Training large transformers (or generating/filtering synthetic data) is computationally heavy. We recorded 10+ hours for RoBERTa-base training and ~7 hours for the data-generation step.
- Complexity rises with prompt engineering, threshold tuning, and multiple training phases.

4.3 Comparison to Existing Systems

Literature on CLINC150 often reports **95–97%** accuracy with powerful models such as **RoBERTa-large** or fine-tuned BERT variants. The original PVI paper (Lin et al., 2023) employed **OPT-66B** to generate synthetic utterances and **RoBERTa-large** as the final classifier, achieving top-tier performance in that range.

- **Our Approach**: We generated synthetic examples using **GPT-2**, a smaller language model than OPT-66B, and fine-tuned **RoBERTa-base** instead of RoBERTa-large. Despite these more modest resources, we still reached ~96% accuracy, placing our system within the established SOTA band.
- **Differences**: Using a smaller generator meant faster sampling but potentially less linguistic variety. Meanwhile, RoBERTa-base—while strong—may lack the final accuracy ceiling that RoBERTa-large can provide. Even so, our results confirm that the PVI methodology scales down effectively and delivers competitive outcomes.

4.4 Limitations

- **Computational Overheads**: RoBERTa-base training took 10+ GPU hours, with ~7 hours per ICDA + PVI run.
- **Model Size Constraints**: GPT-2 offers less diverse and inaccurate generation compared to OPT-66B.
- **Threshold Sensitivity**: We set PVI thresholds manually; finer hyperparameter searches might improve outcomes.
- **Overfitting Risks**: Large transformers can overfit if not carefully tuned, especially when synthetic data distribution differs from real-world queries.

Despite these challenges, our smaller-scale pipeline still achieved near-SOTA accuracy (~96%), confirming the efficacy of PVI-based augmentation on CLINC150.

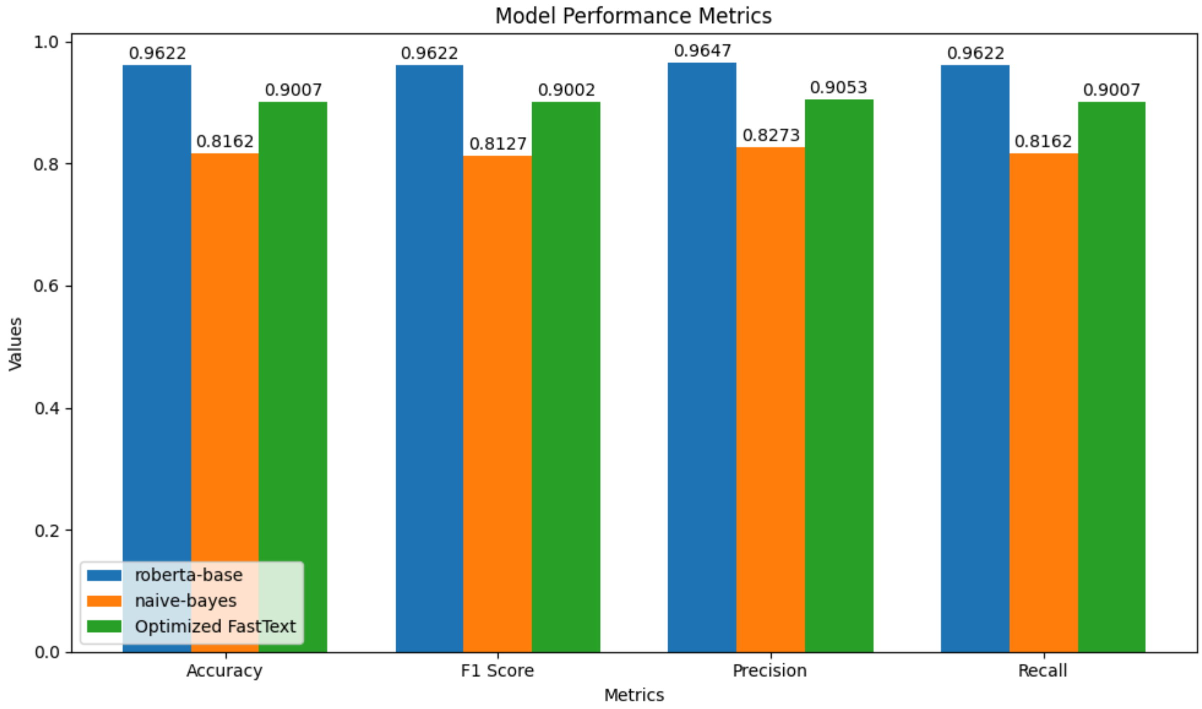
5. Conclusion

This project tackled **intent classification** on the **CLINC150** dataset (150 diverse intents). We tested various methods:

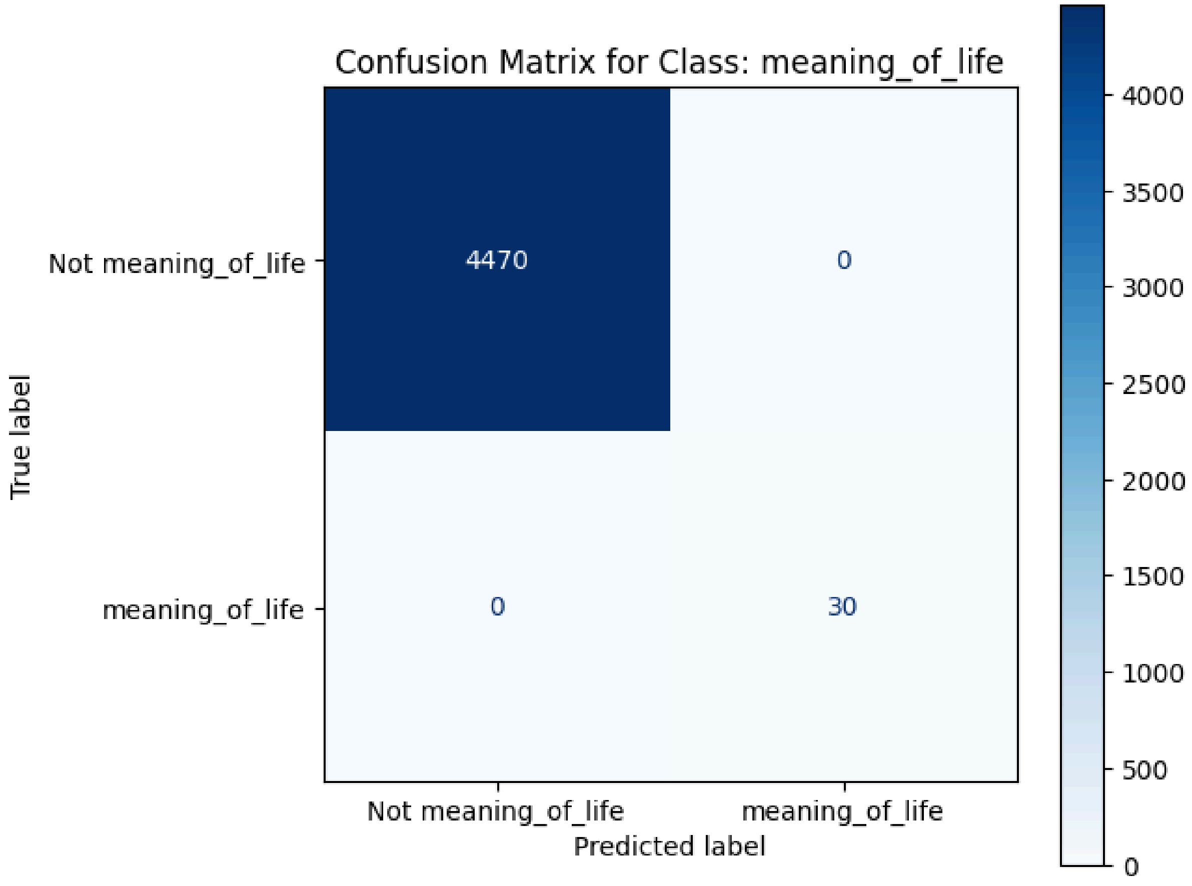
- **Classical ML (TF-IDF)**: Despite hyperparameter tuning (Naive Bayes, Logistic Regression, SVM, etc.), results plateaued around **80–85%** accuracy.
- **FastText**: Subword embeddings lifted accuracy to **~90%**, highlighting the benefits of denser lexical representations.
- **RoBERTa-base**: Fine-tuning a transformer model reached **~96%**, aligning with state-of-the-art for large-scale intent detection.
- **ICDA + PVI**: Generating synthetic queries (via GPT-2) and filtering with pointwise V-information further aided minority classes, though at a computational cost.

Overall, our findings show that **transformer-based** approaches, especially when selectively augmented with synthetic data, can achieve almost-SOTA performance on CLINC150. Future work could investigate larger language models, more advanced prompt engineering, or refined thresholding for PVI to push accuracy even higher.

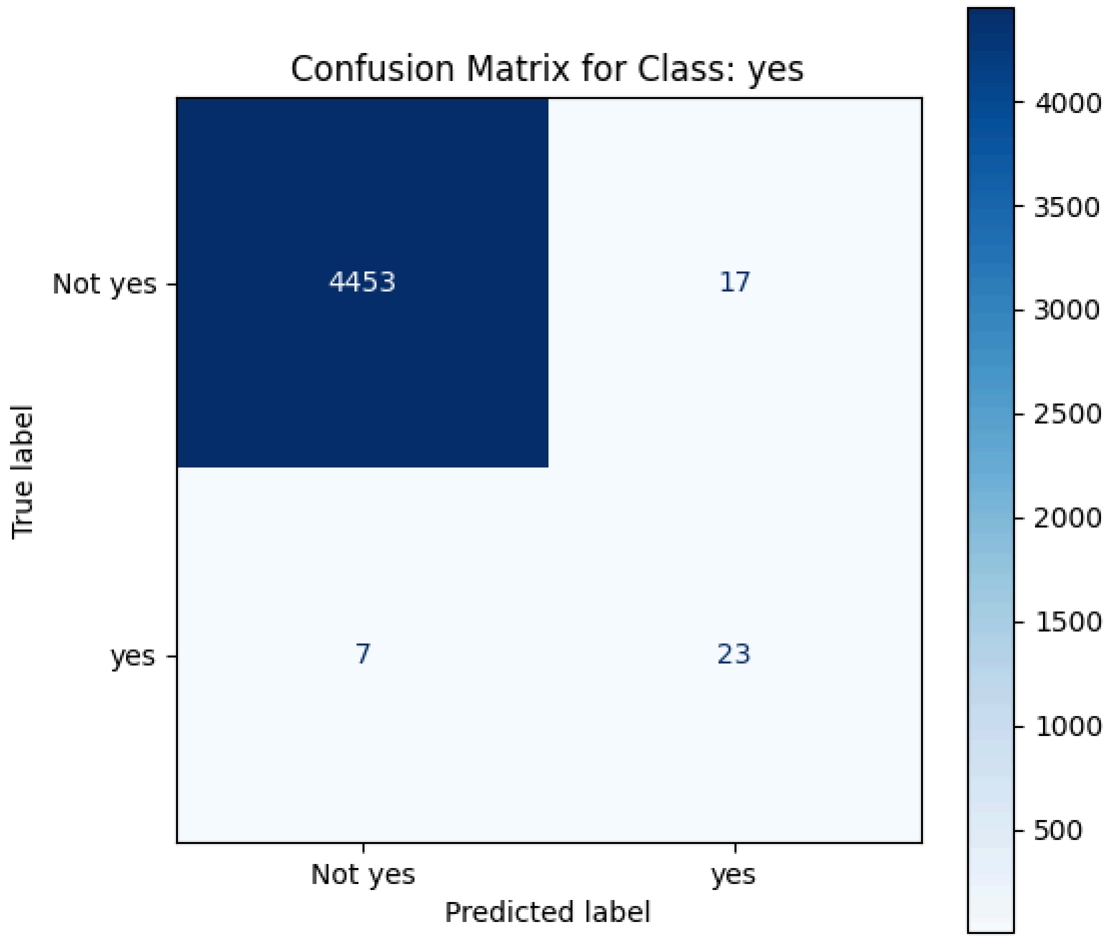
6. Appendix



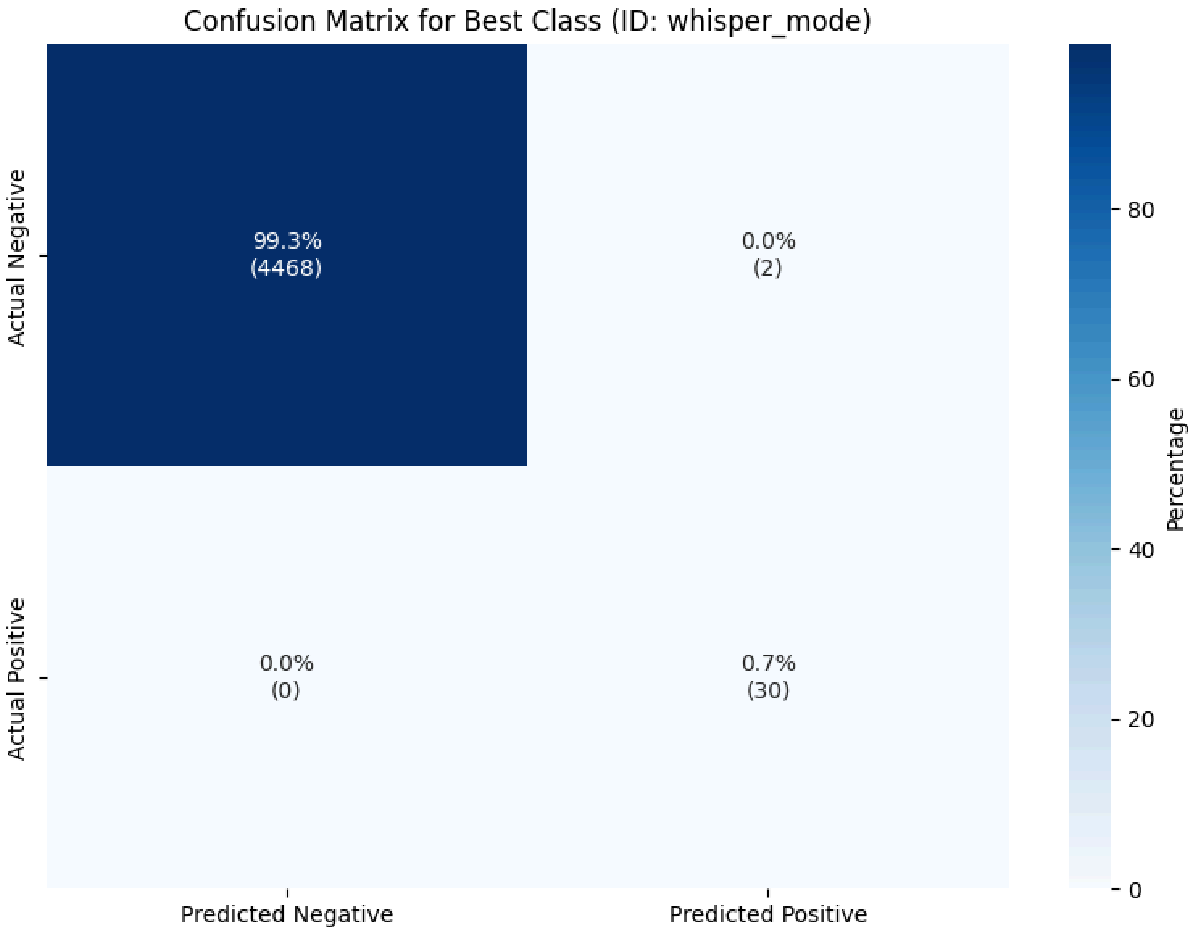
Appendix 1: RoBERTa-base, FastText and Naive Bayes comparison



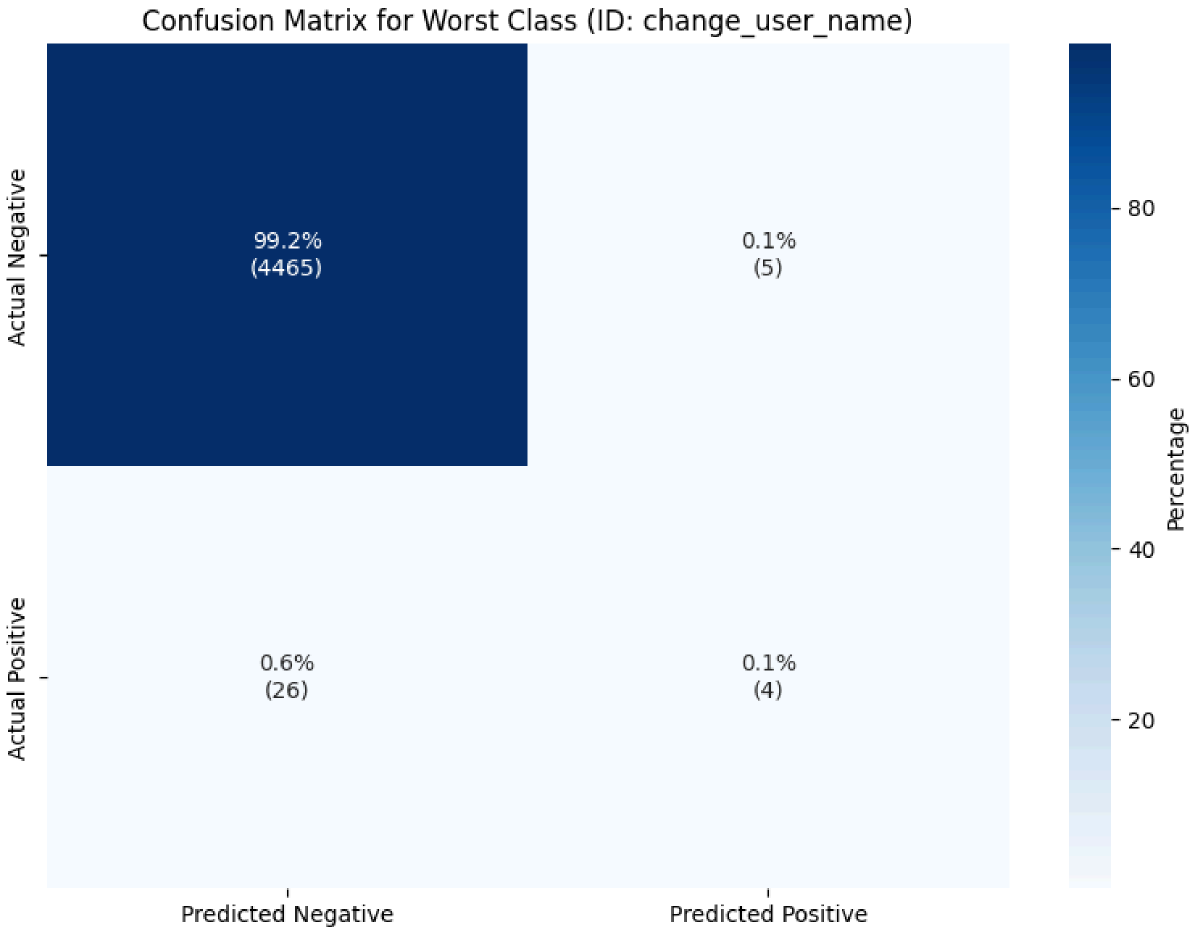
Appendix 2: FastText Best Class Confusion Matrix



Appendix 3: FastText Worst Class Confusion Matrix



Appendix 4: Naive Bayes Best Class Confusion Matrix



Appendix 5: Naive Bayes Worst Class Confusion Matrix

7. References

Hugging Face. (2022). *Fine-tune a pretrained model*. Retrieved January 5, 2025, from https://huggingface.co/docs/transformers/en/training

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). *Bag of tricks for efficient text classification*. arXiv. https://arxiv.org/abs/1607.01759

Lin, Y.-T., Papangelis, A., Kim, S., Lee, S., Hazarika, D., Namazifar, M., Jin, D., Liu, Y., & Hakkani-Tur, D. (2023). *Selective in-context data augmentation for intent detection using pointwise V-information*. arXiv. https://arxiv.org/pdf/2302.05096v1

Mishra, P. (2022). *Fine-tune BERT for text classification*. Kaggle. Retrieved January 5, 2025, from https://www.kaggle.com/code/pritishmishra/fine-tune-bert-for-text-classification?scriptVersionId=116951029