


PROJECT ON NATURAL LANGUAGE PROCESSING		
Group number		Deadline
Group 7		May 3, 2025
5 mai 2025		2024-2025
Lecturer: Dr Cheikh M. Bamba Dione		

Project : Text classification on Bank Data

Group members :

- Chimezie Anthony Odinakachukwu
- Bayiha Jean
- Magatte Fall
- Tidiga Aly

1 Introduction

1.1 Problem

Text classification is a fundamental task in Natural Language Processing (NLP), involving the assignment of predefined categories to textual data. In the banking sector, accurately classifying customer intents is crucial for enhancing user experience and operational efficiency. However, challenges such as limited labeled data, high costs of model training, and the need for real-time processing persist.

1.2 Objectives and Motivation

The primary objective of this project is to explore efficient and cost-effective methods for intent classification in banking applications. Motivated by the increasing demand for intelligent customer service solutions and the limitations of traditional approaches, this study aims to leverage advanced NLP techniques to address these challenges.

1.3 Research Questions

This project seeks to answer the following research questions :

- How can few-shot learning techniques be applied to intent classification in the banking domain ?
- What are the trade-offs between performance and resource utilization when using large language models (LLMs) for this task ?
- How do label inaccuracies in datasets like BANKING77 affect model performance ?

1.4 Related Works

Text classification has evolved significantly from traditional models like logistic regression and support vector machines using TF-IDF, to more advanced deep learning approaches. Pretrained word embeddings such as Word2Vec and GloVe have been widely used, and transformer-based models like BERT and RoBERTa have set new benchmarks in performance.

In the banking domain, Loukas et al. proposed few-shot intent classification using GPT-3.5 and GPT-4 with the Banking77 dataset, demonstrating that in-context learning can compete with or even surpass fine-tuned models [1]. They further introduced low-cost LLM techniques using retrieval-augmented generation and GPT-4 based on data augmentation [2].

Tunstall et al. introduced SETFIT, a prompt-free, sample-efficient method for few-shot learning using Sentence Transformers [3]. SETFIT avoids handcrafted prompts and works well even with minimal labeled data. It outperforms many large prompt-based models, while being faster and requiring fewer resources.

Zhang et al. proposed Gen-PINT, a generative intent detection model that uses instruction tuning to generalize across domains in low-resource settings [4]. Gen-PINT reformulates intent detection as a generation task and achieves state-of-the-art results in zero- and few-shot benchmarks such as Banking77, without needing to update model parameters.

Label noise has also been studied in this context. Ying and Thomas highlighted that nearly 14% of Banking77 samples may be mislabeled, which can distort model evaluation results [5].

Finally, industrial applications like Amazon Comprehend show how these advances are applied in large-scale real-world systems [6].

1.5 Plan

The report is organized as follows :

- **Section 2 : Methods** - Details the methodologies employed for intent classification, including model selection and training procedures.
- **Section 3 : Evaluation** - Presents the evaluation metrics and results obtained from testing the models.
- **Section 4 : Conclusion** - Summarizes the findings and discusses potential future work.
- **Section 5 : References** - Lists all academic works cited throughout the report.

2 Methods

2.1 Approach and Theoretical Background

The main objective of this project is to develop a system capable of classifying user banking queries based on their underlying intent (e.g., checking balance, making a transfer, login issues, etc.). To achieve this, we adopted an approach that combines **Natural Language Processing (NLP)** techniques with **supervised machine learning** algorithms.

Classification Models

Logistic Regression Logistic regression is a linear probabilistic classifier that models the probability of class membership using a logistic (sigmoid) function. It is particularly effective for multiclass classification when combined with numerical representations of text such as TF-IDF. Its main strengths include fast training, interpretability, and robustness to high-dimensional data.

Multi-Layer Perceptron (MLPClassifier) The MLPClassifier is a feedforward neural network trained with backpropagation. Unlike logistic regression, it can capture non-linear relationships thanks to its hidden layers and non-linear activation functions. It is suited for modeling more complex decision boundaries in text classification tasks.

Text Representation

We used two complementary methods to convert textual queries into numerical vectors :

- **TF-IDF (Term Frequency - Inverse Document Frequency)** : emphasizes informative words while reducing the influence of frequent but uninformative ones. It produces sparse vectors that work well with linear models.
- **Word Embeddings (spaCy's `en_core_web_md`, Sentence-Transformer's `all-mpnet-base-v2`)** : generate dense vector representations of words based on their contextual meaning. These embeddings help capture semantic relationships between words.

SETFIT : Sentence Transformer Fine-Tuning for Few-Shot Text Classification SETFIT (Sentence Transformer Fine-Tuning) is an effective method for text classification in *few-shot* scenarios, i.e., when only a small number of labeled examples are available. Unlike prompt-based approaches or large language models, SETFIT is lightweight, fast to train, and does not require prompts.

The approach consists of three main steps :

1. Fine-tuning the Sentence Transformer :

A pre-trained Sentence Transformer (ST) model is adapted using contrastive training in a *Siamese* setup. For each class, positive triplets (sentences from the same class) and negative triplets (sentences from different classes) are generated. The goal is to learn to bring similar sentences closer and push dissimilar ones apart in the embedding space.

2. Training the classification head :

The fine-tuned ST model encodes each training example into a vector. These embeddings are then used to train a classifier (typically logistic regression) that predicts the class labels from these vectors.

3. Prediction (Inference) :

A new sentence is encoded by the ST model and then classified using the trained classification head. This allows fast and efficient predictions, even with very limited training data.

Advantages of SETFIT :

- No need for prompts or complex architectures.
- Effective with very few examples per class.
- Lightweight and fast to train, suitable for resource-constrained environments.

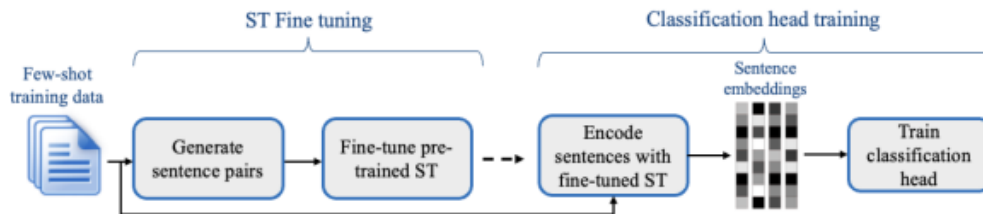


FIGURE 1 – The steps of SetFit model

2.2 Aspects Studied

We investigated several aspects of the methods :

2.3 Experimental Pipeline

Each model was trained through the following steps :

1. **Text Preprocessing** : lowercasing, punctuation removal, stopwords filtering, and lemmatization.
2. **Vectorization** : conversion of text to numerical form using TF-IDF or word embeddings (MPNet-v2).

Aspect	Objective
Model performance comparison	Evaluate the relative performance of logistic regression vs. MLP
Effect of vectorization methods	Compare results using TF-IDF vs. word embeddings
Class imbalance handling	Assess the effect of using <code>class_weight='balanced'</code>
Prediction quality	Use metrics such as accuracy, precision, recall, and F1-score

3. **Label Encoding** : class labels were converted to integers using `LabelEncoder` to be compatible with `MLPClassifier`.
4. **Training and Evaluation** : data was split into a training set (80%) and a test set (20%). Models were evaluated using classification reports and confusion matrices.

2.4 Data Description

Source and Format

The dataset used in this project is the **Banking77** dataset, publicly available on Hugging Face : <https://huggingface.co/datasets/PolyAI/banking77>.

It consists of real-world banking-related queries, each labeled with one of 77 fine-grained intent categories. The dataset is designed to simulate customer interactions in digital banking environments, making it a highly relevant benchmark for intent classification tasks.

- **Source** : Hugging Face Datasets - PolyAI/banking77
- **Number of samples** : 13,083 queries
- **Number of intent classes** : 77 distinct intents
- **Format** : Structured JSON-like format (can be accessed via Python as a Hugging Face dataset object)

Repartition of the 77 categories in the training dataset

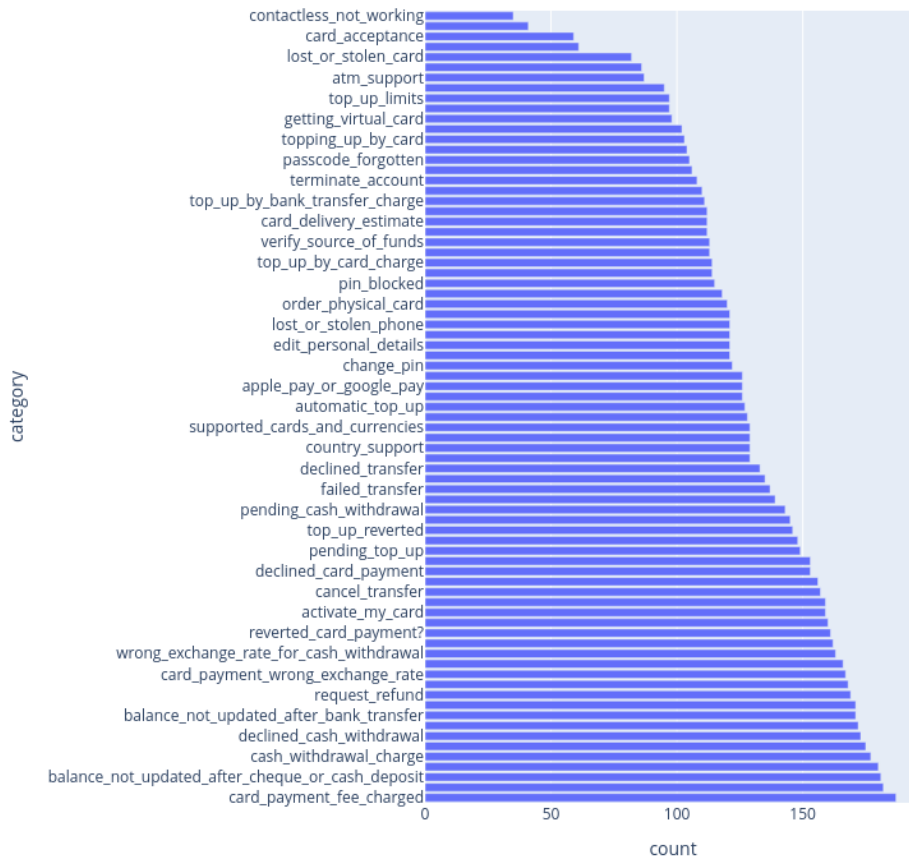


FIGURE 2 – Repartition of the 77 categories in the training dataset

Data Cleaning and Formatting

The dataset was accessed using the `datasets` library from Hugging Face and converted to a pandas DataFrame for ease of use. Minimal cleaning was required, as the dataset is already well-structured. However, the text was standardized by :

- Lowercasing all characters,
- Removing punctuation and special characters,
- Removing stopwords,
- Lemmatizing each word to its root form.

Labels were encoded using `LabelEncoder` to transform the string-based intent labels into numerical class indices suitable for machine learning classifiers.

Train/Test Splitting

The dataset was randomly split into two sets :

- **Training set** : 80% of the data
- **Test set** : 20% of the data

This ensures that model performance is evaluated on previously unseen examples, simulating real-world prediction scenarios.

2.5 Motivation for Choosing This Dataset

Banking queries are ideal for NLP-based classification because :

- They are short, user-driven, and vary in form but not in intent.
- They are often ambiguous and context-dependent, which challenges models to generalize well.
- This use case reflects real-world needs for automating virtual assistants and improving customer support in the financial sector.

3 Evaluation

Technical Limitations

- **Untested Complex Models** : Evaluation of advanced language models (BERT, DistilGPT2) could not be conducted due to hardware constraints (insufficient GPU capabilities and RAM requirements). These architectures demand computational resources exceeding our local infrastructure.
- **Alternative Solutions** : Our methodology focused on lightweight architectures (TF-IDF + Logistic Regression, simple neural networks) providing an optimal balance between performance and efficiency under resource-constrained conditions.
- **Future Improvements** : Potential implementations could leverage :
 - Cloud-based solutions (Google Colab Pro, AWS SageMaker)
 - Compact model variants (DistilBERT, TinyBERT)
 - Model optimization techniques (quantization, pruning)

3.1 Performance Metrics

TABLE 1 – Comparison of NLP Model Performances

Metric	Models					
	TF-IDF + Log. Reg.	MPNet + Log. Reg.	MPNet + MLP	Simple Neural Network	20-shot SetFit	3-shot SetFit
Accuracy	0.861 727	0.880 192	0.989 542	0.57	0.86	0.61
Recall	0.853 946	0.878 000	0.954 461	0.68	0.86	0.60
F1-score	0.854 574	0.874 744	0.954 763	0.61	0.86	0.63

Interpretation : We notice that the MLPClassifier has the highest macro scores (precision, recall, f1-score) on the test set, but SetFit performed better on unseen user queries. This is due to a distribution mismatch : the test data closely resembles the training set and this helps the direct vector-to-class mapping of MLP. However, MLP lacks semantic flexibility and struggles with sentence variations whereas SetFit’s contrastive fine-tuning enables it to recognize intent across diverse formulations, making it more robust for real-world prediction.

3.2 Analysis and Interpretation

TF-IDF based logistic regression is efficient but less accurate due to lack of semantic context. Using SpaCy embeddings improved performance. MLPClassifier performed best but with higher training cost.

Misclassifications were often observed between semantically close intents such as `activate_card` and `card_delivery_estimate`. This highlights the challenges in intent disambiguation with short text inputs.

3.3 Chatbot using DistilGPT-2

CPU-Optimized NLP Banking Assistant

This CPU-optimized NLP chatbot handles common banking requests such as mobile wallet integration. It combines :

- Text embedding-based intent recognition
- Guided dialog flow (confirmation prompts, step-by-step instructions)
- Robust misunderstanding handling

Despite current limitations (simple queries, basic banking domain), its lightweight architecture enables progressive upgrades while maintaining efficient standard hardware performance.

3.4 Limitations and Future Work

- Incorporating contextual embeddings like BERT or DistilBERT.
- Exploring ensemble methods.
- Using retrieval-augmented generation or few-shot prompting with LLMs.

4 Conclusion

In conclusion, through this project we have learned a lot, especially how misclassified observations can influence negatively the performance of a model and the usefulness of models using transformers and few-shot learning to understand semantic

meaning and relationships between the words of a dataset and enhance a model robustness and understanding. We also addressed all of our research questions :

- Few-shot learning techniques : We applied them on banking dataset and evaluated their performance which was really good and enabled the model to capture the meaning of unseen sentences.
- Trade-offs between performance and resource utilization with large language models (LLMs), we observed that these models deliver strong performance but they required great computational resources.
- Impact of label inaccuracies on model performance : We evaluated many models and showed how incorrect labels have degraded the predictive accuracy and generalization.

References

Références

- [1] Lefteris Loukas, Ilias Stogiannidis, Prodromos Malakasiotis, and Stavros Vassos. *Breaking the Bank with ChatGPT : Few-Shot Text Classification for Finance*. FinNLP-Muffin Joint Workshop, Macao, August 2023.
- [2] Lefteris Loukas et al. *Making LLMs Worth Every Penny : Resource-Limited Text Classification in Banking*. 4th ACM Conference on AI in Finance (ICAIF), Brooklyn, NY, November 2023.
- [3] Lewis Tunstall et al. *Efficient Few-Shot Learning Without Prompts*. arXiv preprint arXiv :2209.11055, 2022.
- [4] Feng Zhang et al. *From Discrimination to Generation : Low-Resource Intent Detection with Language Model Instruction Tuning*. Findings of ACL 2024, pp. 10167-10183.
- [5] Ying and Thomas. *Label Errors in BANKING77*. Preprint, 2022.
- [6] Amazon Web Services. *Amazon Comprehend - NLP Service*. <https://aws.amazon.com/comprehend/>, Accessed 2025.