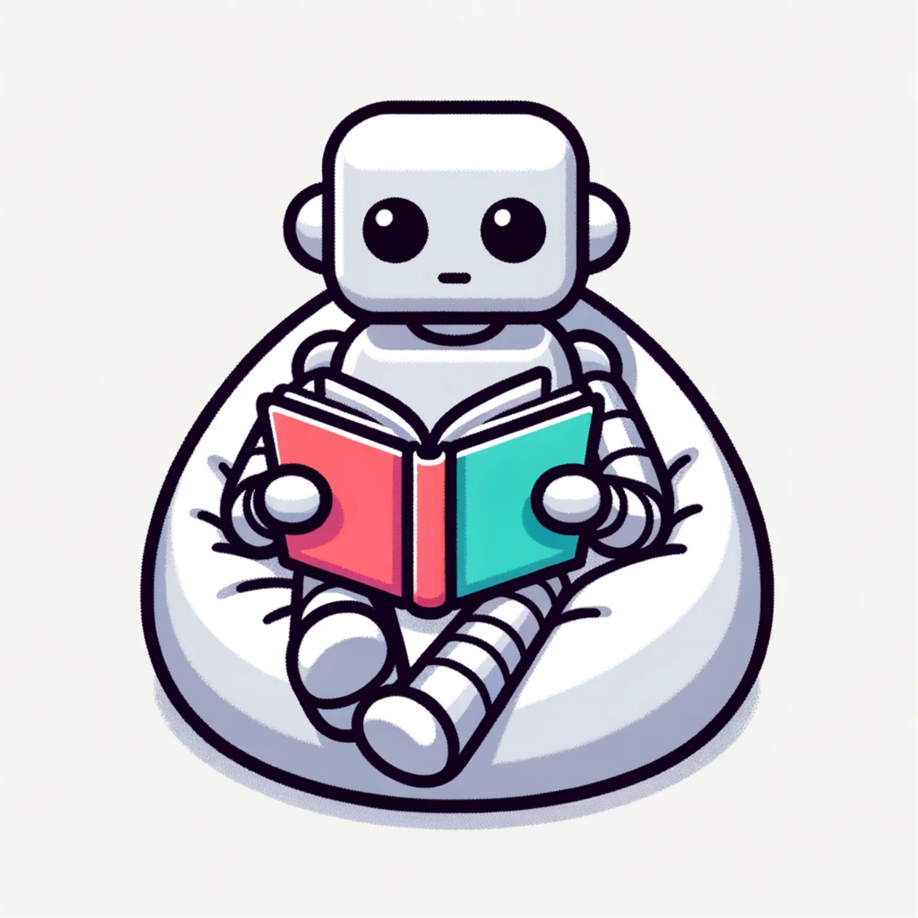
Text classification analysis

NPR MC 1

rami tarabishi

October 2023



Contents

1 Introduction

This short report covers and discusses the dataset and text classification models chosen, my methodology and results found during NPRs first mini challenge.

1.1 Objectives and problem description

Banks receive a large volume of inquiries from customers on a daily basis, ranging from account-related questions to transaction disputes and loan inquiries. Accurately categorizing and understanding the intent behind these inquiries is crucial for providing efficient and effective customer support. In this context, the problem of intent detection in bank inquiries can be addressed using text classification methods.

The objectives of this mini challenge are to earn knowledge about different natural language processing methods through the evaluation of 2 different text classification models. More specifically to my chosen dataset is to evaluate these 2 text classification models on their ability to correctly and automatically categorize bank inquiries into their specific intent categories. In the real world this would be used to streamline customer support by redirecting the inquires automatically to their intended/responsible divisions.

2 Model selection

2.1 TF-IDF with Support Vector Machines

Term Frequency-Inverse Document Frequency with Support Vector Machines is a technique used for information retrieval and text classification.

Term Frequency measures the frequency of a term or word within a document. It is calculated as the number of times a term appears in a document divided by the total number of terms in the document. It reflects how important a term is in a document.

Inverse Document Frequency is a measure of how unique or important a term is a cross a collection of documents or corpus. It is calculated as the logarithm o f the total number of documents divided by the number of documents containing the term. Higher values indicate that a term is rare and provides more discriminative power.

TF-IDF Score the combination of the two through multiplication of TF by its IDF returns a representation of how important a term is within a document relecvant to the entire corpus. Finally TF-IDF transforms the documents into a numerical vector, where each element in the vector corresponds to the TF-IDF score of a term in the document. Then a Vector space model is constructed where each document is a point in this space which represents its importance.

Support Vector Machines or SVMs are powerful classifiers, they seek to find the best linear hyperplane to separate data into distinct classes while maximizing the margin between these classes, where the margin is the distance between the hyperplane and the nearest data points from each class.

To achieve text classification, SVMs compute the hyperplanes that separate the space where each class sits in the TF-IDF vector space, where then test data can be inserted in and the class predicted from its place in the space.

2.2 DistilBERT

Bidirectional Encoder Representations from Transformers better known as BERT is a pre-trained transformer based NLP model that captures contextual information in text. As BERT is "bidirectional" the model can use the context left and right of a word allowing it to learn the meaning of words from its surroundings. This makes BERT extremely good at understanding context in text.

DistilBERT is a so called distilled version of BERT, providing most of regular BERTs performance while being much smaller and faster, I chose DistilBERT due to its performance and speed with understanding context and questions making it suitable for my chosen dataset.

2.3 Additional models

Some additional models that I looked at but decided not to use in this mini challenge are:

1. Multinomal Naive Bayes
2. Logistic Regression
3. Recurrent Neural Networks

Both Multinomial Bayes and Logistic Regression would build upob the TF-IDF vector space as the classifier just like SVMs do.

3 Methodology

3.1 Data set

For my dataset I am choosing the Banking77 dataset (provided by PolyAI through huggingface) which is about inquiries to banks, it is comprised of around 13000 inquiries, by default it is split into 10000 to train and 3000 to test. The dataset has 77 intent classes.

The distribution of intent classes is as follows for both the train and test set:

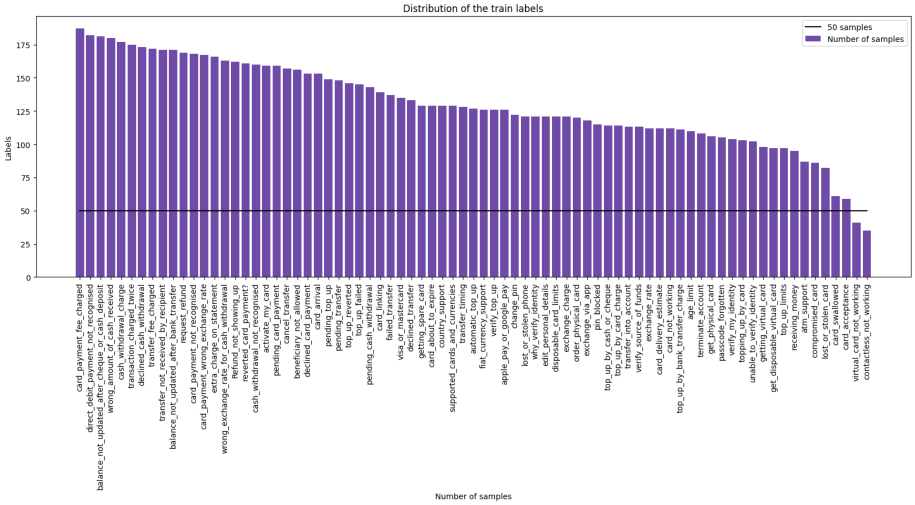


Figure 1: Distribution of the training set

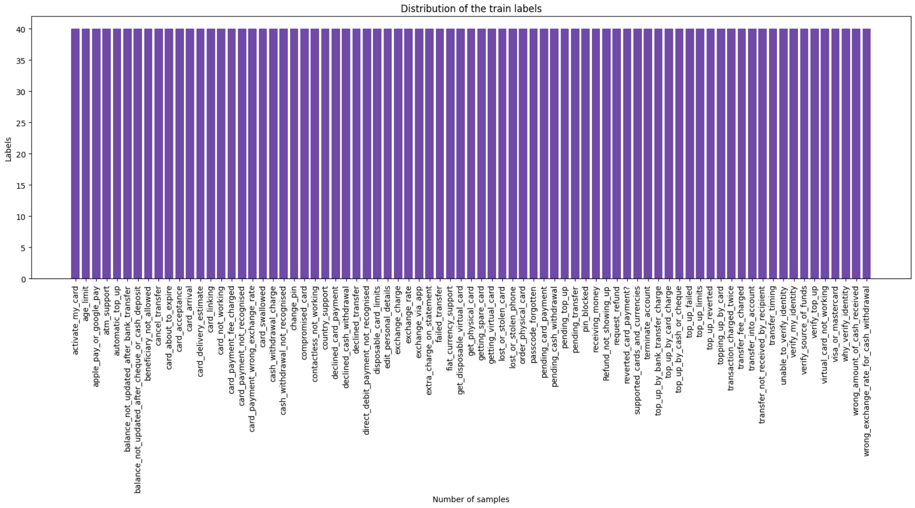


Figure 2: Distribution of the testing set

As one can see the training dataset is not very balanced, which can lead to misrepresentation of performance when using cetrain metrics like accuracy, therefore I will have to take that into account and use appropriate methods. On the other hand, the publishers of the dataset made the pre-selected training set perfectly balanced containing 40 samples of each class.

Some examples of data are:

Text	Label
I am still waiting on my card?	card arrival
Can I track my card while it is in the process of delivery?	card arrival
Am I able to cancel a transfer I just made	cancel transfer
Isn't my top-up supposed to mean its complete?	pending top up
I would like to know how much I can top-up.	top up limits
How do I add my new card?	card linking

3.2 Data preparation and exploration

Usually the first steps in prepping data is to clean the data of noise, this involves removing special characters, HTML tags, "stop words" and or anything that is not raw text. Next as machine learning models cannot process raw text, Tokenization is used to break the text down into individual words or sub units and after Tokenization stemming or lemmatization is done to reduce words to their root forms. Finally the clean data is then structured into (numerical vectors/embeddings) and split into training, validation and testing sets.

As for data exploration there is not much to be done due to the data being unstructured, other than analysing the classes which was displayed above in the last section.

3.3 The training process

The process I used for training the models made very easy and simple by huggingfaces libraries that I discovered while searching for dataets to use. They provide both the models and methods used to setup/configure, train and evaluate them. The libraries I used were:

1. Datasets — To load and structure data
2. Transformers — Provided pretrained models and config/trainer functions
3. Evaluate — Provided evaluation metrics

As TF-IDF and SVMs are implemented in the SKLearn library the training process was different than DistilBERT.

TF-IDF with SVMs was trained by first using sklearns TF vectoriser to create the TF-IDF vector space of both the training and test dataset. Once the vector space was created a SVM (Also provided by sklearn) computed the hyperplanes separating the classes in the space.

DistiliBERT was trained through the aforementioned huggingface libraries using the "trainer" method which takes in a model config that declares all the training arguments, hyper parameters, evaluation metrics and device to train the model on, (I used my local 2080ti to train all the models).

3.4 Evaluation metrics

For the evaluation metrics I decided to keep it simple with accuracy and a weighted F1 Score.

Accuracy is the ratio of true predictions to false ones, I chose due to it being a simple and easy to understand metric, although as mentioned before it can be less accurate of a depiction of model performance as the classes are not balanced.

F1 Score is the harmonic mean between precision (Measures the accuracy of positive predictions) and recall (Or sensitivity, it is the ratio of true positives to the total number of actual positive instances), I chose it as a back-up metric to accuracy due to the imbalance of classes and it being a strong indicator of classification model performance.

4 Results and discussion

4.1 TF-IDF with SVM

The TF-IDF with Support Vector Machines (SVMs) model yielded promising results in the context of intent detection in bank inquiries, achieving an accuracy of 88.57 percent and an F1 score of 88.56. These results highlight the model's ability to correctly categorize customer inquiries into specific intent categories. Moreover, one of the noteworthy advantages of the TF-IDF with SVMs approach is its exceptional speed in both training and inference, making it a highly efficient choice for real-time intent detection in a customer support system.

Some pros and cons of the model are:

1. Speed — in context to my dataset, the TF-IDF model was extremely fast to train and run inference on.
2. Interpretability — SVMs provide a transparent decision boundary, making it easier to understand and interpret the model's predictions

And a big con of TF-IDF models is relative Performance — In comparison to the DistilBERT model, it performed measurably worse by about 5 percent in both Accuracy and F1 scoring. This can be undesirable especially in the customer support sector.

4.2 DistilBERT

The DistilBERT model, a modern transformer-based architecture, demonstrated exceptional performance in the realm of intent detection in bank inquiries. It achieved an impressive accuracy of 93.31 percent and an F1 score of 93.3. These results underscore the model's capability to accurately categorize customer inquiries, outperforming the TF-IDF approach by a notable margin.

Pros of DistilBERT and BERT models are:

1. Performance — BERT models excel in understanding the contextual nuances of language, leading to overall better performance.
2. Multilingual Capability — Many BERT variants, including DistilBERT, support multiple languages, making them versatile for tasks involving diverse linguistic contexts.

Cons of BERT models:

1. Computationally expensive — Transformer models like BERT can be computationally expensive in both compute and memory, requiring substantial resources for training and inference.
2. Inability to Handle Out-of-Distribution Data — BERT models, including DistilBERT, can struggle when faced with out-of-distribution data or queries that significantly differ from the training data.

Out of curiosity I trained a regular BERT model with the same hyper parameters as my DistilBERT and achieved practically the same results with an accuracy of 93.28 percent and an F1 score of 93.28.

4.3 Conclusion

In the realm of customer support, where the efficient and accurate categorization of inquiries is paramount, the choice of text classification model plays a pivotal role. My evaluation of two distinct models, TF-IDF with SVMs and the BERT-based DistilBERT, underscores the critical importance of model performance in this context. The superior accuracy and F1 score achieved by the DistilBERT model, over 93 percent, exemplify the pivotal role that advanced, context-aware models play in automating intent detection. It is noteworthy that even the full-fledged BERT model, yielded practically the exact same performance as the smaller DistilBERT, highlighting the efficiency and resource optimization inherent to the latter. These findings underscore the transformative potential of state-of-the-art NLP models in revolutionizing customer support, enabling institutions to provide prompt, precise, and personalized assistance while emphasizing the impact of model selection on the overall quality of customer service.