Team ID: Team Name:

Intent Classification in Banking Queries: A Comparative Study of Classical, Embedding-based, and Transformer Models

Abstract

- We tackle intent classification for short banking queries using the BANKING77 dataset[1] (77 intents).
- We compare three model families: TF/TF-IDF + Multinomial NB, Word2Vec sentence embeddings + MLP, and fine-tuned transformer encoders (BERT / DistilBERT), all trained on the same split and evaluated with macro-F1.
- Transformer models achieve the best performance: BERT with CLS pooling reaches macro-F1 ≈ 0.95, while TF-based and Word2Vec baselines remain competitive (macro-F1 ≈ 0.86 and 0.88) at substantially lower computational cost.

Goal of the Project

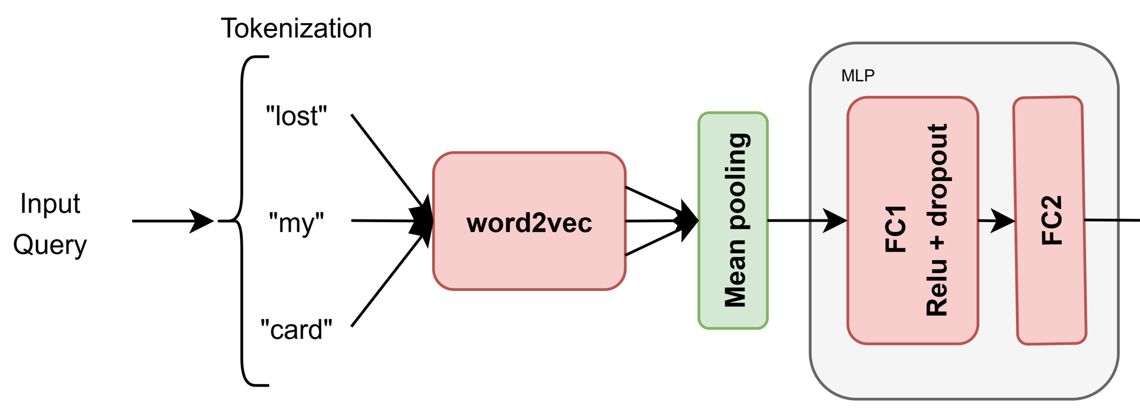
- Build an intent classifier that maps short banking queries to 77 predefined BANKING77 intent classes.
- Compare classical, embedding-based, and transformer models on BANKING77 and analyze their results to understand performance trade-offs.

Methodology

- **Dataset**
  - **BANKING77:** 13k short English banking queries labeled into 77 intent classes.
  - Clean, curated text: no noise → only light preprocessing needed.
  - Imbalanced dataset → we report macro-F1, macro-precision, and macro-recall.
  - **Preprocessing (per method)**:
    - **Method 1:** default preprocessing from CountVectorizer / TfidfVectorizer (lowercasing, tokenization, n-grams).
    - **Method 2:** Lowercasing, quote normalization, and regex-based word tokenization; sentence = mean of Word2Vec embeddings (OOV → UNK).
    - **Method 3:** Raw text passed to the pretrained tokenizer for subword tokenization and truncation (max length 64), with dynamic padding at the batch level.
  - **Data split:** 70% train / 15% validation / 15% test, using a fixed random seed and the same split.

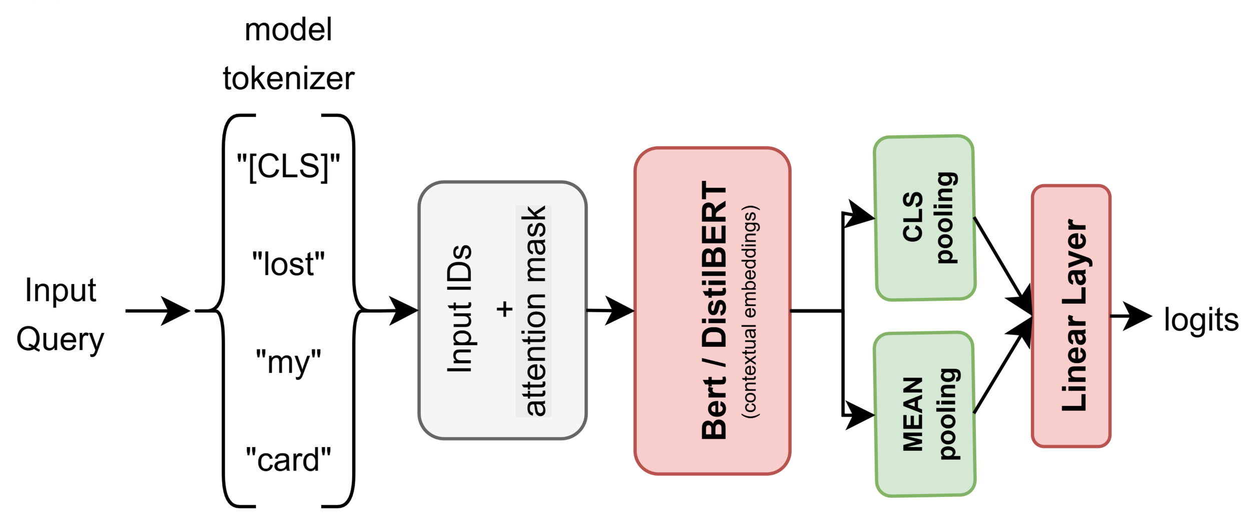
- **Method 1 TF / TF-IDF + Multinomial Naive Bayes**
  - **Approach**
    - Convert queries to sparse bag-of-words vectors (TF / TF-IDF).
    - Train a Multinomial Naive Bayes classifier using grid search.
    - Evaluate the best two models on the test set.
  - **Tuning**
    - 5-fold stratified grid search.
    - Searched *n*-grams, min_df, max_df, sublinear TF (TF-IDF), and NB smoothing α.
    - Objective: macro-F1 (class-imbalance aware).
    - **Best Configurations**
      1. TF: unigrams + bigrams, min_df = 2, $\alpha = 0.1$.
      2. TF-IDF: unigrams, min_df = 3, sublinear TF = True, $\alpha = 0.1$.

- **Method 2 Word2Vec Sentence Embeddings + MLP**
  - **Approach**
    - Pretrained word2vec-google-news-300.
    - Sentence embedding = mean of word vectors.
    - Train a small MLP classifier (ReLU + dropout).



  - **Tuning**
    - Hyperparameter tuning on the validation macro-F1.
    - **Best configuration :** Hidden=512, Dropout=0.1, Batch=64, lr= 0.001, weight_decay=0, label_smoothing=0.1
  - **OOV Handling**
    - Variant 1: Include OOV tokens.
    - Variant 2: Ignore OOV tokens.

- **Method 3 Fine-Tuned Transformer Encoders (BERT / DistilBERT)**
  - **Approach**
    - Fine-tune two pretrained encoders: BERT-base-uncased and DistilBERT-base-uncased.
    - Sentence representation (pooling strategy):
      1. CLS token.
      2. Mean pooling.
    - Apply a Linear layer as classifier head.



  - **Tuning**
    - Jointly fine-tune encoder + classification head using AdamW and cross-entropy.
    - Hyperparameter tuning on validation macro-F1.
    - **Best configuration:**
      1. BERT (CLS / AVG): lr = 3e-5, weight_decay = 0.0, warmup_ratio = 0.06, batch_size = 64, epochs = 15.
      2. DistilBERT (CLS): lr = 3e-5, weight_decay = 0.01, warmup_ratio = 0.06, batch_size = 64, epochs = 15.

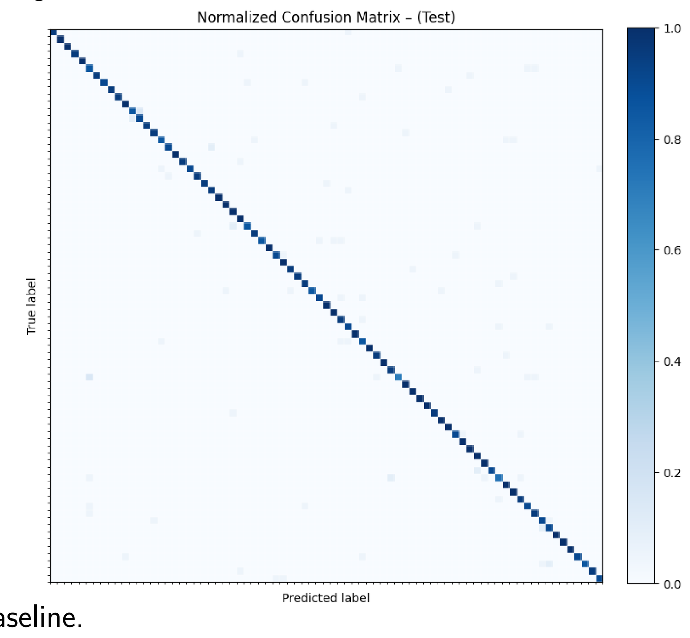
Results

- **Metrics**
  - Metrics reported on the test split: macro-recall, macro-precision, macro-F1 (averaged over 77 intents).
- **Overall performance**

Method	Model	Recall	Precision	F1-score
Baselines	BERT [2]	0.926	0.931	0.926
	distilbert [3]	-	-	0.924
	RoBERTa [4]	0.935	0.937	0.935
1	TF + Multinomial NB	0.865	0.871	0.865
	TF-IDF + Multinomial NB	0.855	0.864	0.855
2	Word2Vec + MLP (with oov)	0.875	0.886	0.875
	Word2Vec + MLP (ignoring oov)	0.879	0.889	0.879
3	DistilBERT (cls)	0.930	0.933	0.930
	BERT (cls)	**0.945**	**0.947**	**0.945**
	BERT (avg)	0.940	0.943	0.940

- **Confusion Matrix**
  - Strong diagonal dominance indicates highly accurate class separation.
  - Very low off-diagonal density → misclassifications are rare and scattered.
  - No major overlapping classes, reflecting stable generalization across all 77 intents.

Fig: Normalized confusion matrix for the best model.



- **Key Observations**
  1. **First Method:**
     - TF performs slightly better than TF-IDF on BANKING77.
     - Best TF model reaches macro-F1 0.865, a strong classical baseline.
  2. **Second Method:**
     - Word2Vec sentence embeddings + MLP improve performance over bag-of-words.
     - Ignoring OOV tokens yields a small but consistent gain (0.875 → 0.879 macro-F1).
  3. **Third Method:**
     - All transformer variants clearly outperform Methods 1 and 2.
     - DistilBERT (CLS) reaches macro-F1 = 0.930, BERT (avg) reaches 0.940, and BERT (CLS) reaches 0.945, with BERT clearly surpassing DistilBERT and CLS pooling slightly ahead of mean pooling.
  4. **Overall:**
     - Performance improves from sparse (TF) → dense static embeddings (Word2Vec) → contextual transformers.
     - Method 1 gives a strong, efficient baseline; Method 2 adds semantic structure at moderate cost; Method 3 achieves the best performance, matching and exceeding published transformer baselines.

Analysis

- **First Method:**
  - TF performs slightly better than TF-IDF, which may be due to short, repetitive queries where down-weighting frequent terms removes useful signal.
  - Best TF setup (unigrams + bigrams, min_df = 2, = 0.1) may be due to bigrams capturing common patterns and min_df removing noisy rare n-grams.
  - Overall performance (≈0.86 macro-F1) is strong for a classical model, showing it handles this dataset surprisingly well.
- **Second Method:**
  - Word2Vec improves over TF/TF-IDF, which may be due to dense semantic embeddings reducing sparsity and giving similar embeddings to related words, unlike TF/TF-IDF's sparse counts with no semantic structure.
  - Ignoring OOV tokens slightly outperforms using a UNK embedding, which may be due to avoiding distortion in the averaged sentence representation.
  - Both variants outperform Method 1, and the best result (≈0.879 macro-F1) is solid for such a lightweight model, which is also expected given the added semantic information.
- **Third Method:**
  - Transformers outperform Methods 1 and 2, explained by the fact that they learn contextual embeddings using multi-head attention, unlike TF/TF-IDF's sparse counts or Word2Vec's static vectors.
  - BERT surpasses DistilBERT, explained by its larger depth and representational capacity, leading to stronger feature extraction.
  - Within BERT, CLS pooling slightly beats mean pooling: the CLS token is explicitly trained as a sequence summary, while averaging can dilute important tokens.
  - We outperform the published Banking77 baselines, which may be due to more careful fine-tuning (hyperparameter tuning, longer training, and using the best pooling choice, CLS).
- **Compute trade-off:** TF/TF-IDF (fastest) → Word2Vec → DistilBERT → BERT (slowest, most accurate), as expected from increasing model complexity.

Limitations

- Limited compute and time: hyperparameter search spaces were kept small and only a few configurations were explored per model family.
- several variants were not tested (e.g., deeper MLPs, alternative pooling strategies, different pretrained encoders).
- Limited number of training runs and random seeds, so results may be sensitive to initialization and specific splits.

References

1 PolyAI, *BANKING77 Dataset*. Hugging Face Datasets, 2024. Available at: `https://huggingface.co/datasets/PolyAI/banking77`

2 P. Schmid, *BERT-Banking77*. Hugging Face, 2021. Available at: `https://huggingface.co/philschmid/BERT-Banking77`

3 N. Procopio, *DistilBERT-base-uncased-Banking77*. Hugging Face, 2023. Available at: `https://huggingface.co/nickprock/distilbert-base-uncased-banking77-classification`

4 N. Procopio, *XLM-RoBERTa-base-Banking77*. Hugging Face, 2023. Available at: `https://huggingface.co/nickprock/xlm-roberta-base-banking77-classification`