Project 2

Mohammed Musthafa Rafi
mohd7@iastate.edu
Kaggle username: Mohammed Musthafa Rafi

1 Method

I have explored three main approaches for document classification. The first two involve Logistic Regression models with different feature representations—GloVe embeddings and TF-IDF vectors. The third approach integrates contextualized embeddings from BERT with static GloVe vectors to capture both global lexical semantics and sentence-level context. While the final classifier remains Logistic Regression in all three cases, the nature of the input embeddings greatly impacts performance, especially for identifying minority class samples more accurately.

Method A: GloVe Embeddings + Logistic Regression.

1. **Preprocessing**: Concatenate title and abstract, lowercase, and tokenize.
2. **Embedding**: Use GloVe 42B (300d) vectors (1). Each document is the average of its token embeddings.
3. **Oversampling**: Apply *RandomOverSampler* to mitigate class imbalance.
4. **Model**: Train a Logistic Regression model (`max_iter=1000`) on the oversampled data.
5. **Threshold Tuning**: Evaluate at default 0.5 and a custom threshold 0.23.

Method B: TF-IDF + Logistic Regression.

1. **Preprocessing**: Identical text concatenation and lowercasing.
2. **TF-IDF Features**: Use `TfidfVectorizer` with up to 10,000 features, stop words removed, and $(1, 2)$-grams.
3. **Model**: Train Logistic Regression with `class_weight=balanced`, `max_iter=1000`.
4. **Threshold Tuning**: Evaluate at default 0.5 and a custom threshold 0.17.

Method C: BERT + GloVe Embeddings + Logistic Regression.

1. **Preprocessing**: Same text lowercasing and concatenation.
2. **Embedding Fusion**: Combine a BERT-based embedding (e.g., from a pretrained model) with the GloVe 42B (300d) vector, typically through concatenation or averaging.
3. **Model**: Train a Logistic Regression classifier on the fused embeddings.
4. **Threshold Tuning**: Evaluate at a custom threshold (e.g., 0.22) and optionally others.

2 Discussion

GloVe + Logistic Regression (Method A). Table 1 shows the Dev set results at thresholds 0.5 and 0.23. Lowering the threshold yields higher recall for class 1 but at the expense of overall precision and accuracy.

Threshold	Prec (C1)	Rec (C1)	F1 (C1)
0.5	0.58	0.94	0.71
0.23	0.43	1.00	0.60

Table 1: GloVe-based Logistic Regression on Dev (Class 1). Overall accuracy: 93% (0.5) vs. 88% (0.23).

TF-IDF + Logistic Regression (Method B). In Table 2, we see that TF-IDF features allow strong performance at both threshold 0.5 and a custom threshold 0.17. Precision and recall vary as expected with threshold changes.

Threshold	Prec (C1)	Rec (C1)	F1 (C1)
0.5	0.65	0.92	0.76
0.17	0.44	1.00	0.61

Table 2: TF-IDF-based Logistic Regression on Dev (Class 1). Overall accuracy: 95% (0.5) vs. 88% (0.17).

BERT + GloVe + Logistic Regression (Method C). Integrating contextualized embeddings (e.g., BERT) with static GloVe vectors can capture both semantic nuances and broader lexical information. At a threshold of 0.22, we achieved 96% overall accuracy on the Dev set, with precision of 0.80 and recall of 0.78 for class 1 (Table 3). Lowering the threshold further (e.g., 0.20) can increase the number of predicted positives, but generally decreases precision.

Threshold	Prec (C1)	Rec (C1)	F1 (C1)
0.22	0.80	0.78	0.79

Table 3: BERT + GloVe on Dev (Class 1). Accuracy: 96%.

All three methods show strong performance, but BERT+GloVe appears to balance accuracy and class 1 performance, yielding an F1 of 0.79 at a 96% overall accuracy. As before, threshold adjustments let us trade off precision and recall for the minority class.

References

[1] Jeffrey Pennington, Richard Socher, and Christopher Manning.
2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

[2] Nils Reimers and Iryna Gurevych.
2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1910.06386*.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

A Appendix: Resources

Below are relevant resources and references:

- https://github.com/itsMustafamr/qtl_text_analysis_project_579 (GitHub repository with code from project 1 with basic preprocessing)
- https://nlp.stanford.edu/projects/glove/ (GloVe embeddings)
- BERT vs. Simple Logistic Regression for NLP (Medium article)
- https://scikit-learn.org/stable/ (scikit-learn for Logistic Regression, TF-IDF, and other utilities)
- https://github.com/google-research/bert (BERT)

All Python scripts, threshold-tuning, and classification reports are committed to the GitHub repo. Github - https://github.com/itsMustafamr/Project2_COMS_579