CSE440: Natural Language Processing II Multi-Class Text Classification: A Comparison of Word Representations and ML/NN Models

Fahmida Afrin

Faria Akter Tonny

Kazi Shahid Ahmed Galib

Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh

***Abstract*—Recent years have witnessed rapid advances in natural language processing (NLP), yet multi–class question–answer classification remains challenging due to the richness of human language and the large number of semantic categories. This work explores a standard ten–class question-answer dataset (85-15 training/validation split) and systematically compares classical machine learning (ML) models with neural network (NN) architectures under four word-representation schemes: Bag-of-Words (BoW), term frequency–inverse document frequency (TF-IDF), pre-trained GloVe embeddings and Skip-gram Word2Vec. After extensive exploratory data analysis and text pre-processing (tokenization, stop-word removal, lemmatization and handling class imbalance), we train tuned logistic regression, multinomial naive Bayes and random forest classifiers on BoW and TF-IDF features. Subsequently we implement deep neural networks, bidirectional recurrent networks (RNNs), long short-term memory (LSTM) and gated recurrent unit (GRU) models on dense GloVe and Skip-gram representations. We evaluate models using accuracy, macro-averaged F1 and the area under the receiver operating characteristic curve (AUC). Our best classical baseline, *logistic regression + TF-IDF*, attains a test accuracy of 70.3% and macro-F1 of 0.701, while the best neural model, a deep neural network trained on Skip-gram embeddings, achieves a slightly higher test accuracy of 71.2% and macro-F1 of 0.707. Bidirectional RNNs and LSTMs provide competitive performance ($\approx$ 0.71–0.72 accuracy), whereas GRU models exhibit greater over-fitting and do not surpass the DNN. We conclude with a discussion of model behaviour, observed class-wise confusion patterns, and suggest avenues for further improvement.**

***Index Terms*—text classification, word embeddings, machine learning, neural networks, GloVe, Skip-gram**

Author Contributions

Fahmida Afrin designed and performed the exploratory data analysis and implemented text preprocessing, the Naive Bayes and Random Forest Models and Deep Neural Network. Kazi Shahid Ahmed Galib developed the neural network models and conducted hyper-parameter tuning. Faria Akter Tonny worked with Logistic Regression, evaluated the models, produced visualizations and wrote the report. All authors contributed to discussion and final editing.

This project was completed as part of CSE440 at Brac University.

I. Introduction

Multi-class question–answer classification aims to assign incoming questions to one of several intent categories so that appropriate answers can be returned. In educational and customer-service domains, automatic routing reduces human workload and improves response time. However, the task is challenging because questions span diverse topics and exhibit varying lengths, vocabulary and grammatical styles. Simple keyword matching fails when different classes share similar terms or when semantically meaningful phrases are rare in the training data.

The CSE440 lab project requires students to compare a suite of text representation techniques (BoW, TF-IDF, GloVe and Skip-gram) with both classical ML and modern NN models [1]. Previous studies show that word embeddings capture semantic similarities beyond raw term frequencies, and that sequence models such as LSTMs excel at tasks where word order matters [2]. Nevertheless, simpler linear models often provide strong baselines and are computationally efficient [3].

This report follows the IEEE conference format and summarizes our methodology, experimental results and key insights from the project. All experiments strictly use the provided training set for model development and reserve the test set for final evaluation.

II. Methodology

Our workflow comprises four main stages: data exploration and pre-processing, word representation, model development, and evaluation.

A. Dataset and Pre-processing

The dataset consists of approximately *280,003* question–answer pairs annotated into *ten* intent categories (numbered 0–9) provided by the course instructors. We split the data

into an 85 % validation set and 15 % held-out test set. Each question was lower-cased, tokenized using NLTK, stripped of punctuation and stop words, and lemmatized to reduce inflectional variance. Because some classes (e.g., category 1) dominate the corpus while others are under-represented (e.g., category 8), we computed class weights inversely proportional to class frequencies to mitigate bias during training. Empty or extremely short questions were removed.

B. Word Representations

BoW and TF-IDF. We first transformed tokens into sparse vectors using the Bag-of-Words representation with unigrams and bigrams and a minimum document frequency of 5. TF-IDF weighting down-weights common words and produced higher discriminative power. To reduce dimensionality and training time for neural models, we applied truncated singular value decomposition (SVD) to project TF-IDF vectors into 300–768 dimensions.

Pre-trained Embeddings. For dense representations we used 100-dimensional GloVe embeddings trained on Wikipedia and Gigaword. Tokens not present in the GloVe vocabulary were initialized randomly. We also trained a Skip-gram Word2Vec model on the training corpus with window size 5, negative sampling and 300 dimensions. Embeddings were fine-tuned during neural training.

C. Models and Hyper-parameter Tuning

a) Classical ML models.: We compared three algorithms on sparse features: logistic regression (LR) with L2 regularization, multinomial naive Bayes (MNB) and a random forest (RF) classifier. Hyper-parameters such as regularization strength, smoothing parameter α and number of trees were tuned on the validation split via grid search.

b) Deep Neural Networks.: On dense TF-IDF or embedded vectors, we implemented fully connected networks (DNNs) with two hidden layers (384–768 units), gelu/ ReLU activations, batch normalization, dropout and weight decay. The Adam or AdamW optimizer with an initial learning rate between 3×10^{-4} and 5×10^{-4} and clipnorm of 1.0 was used.

c) Recurrent Models.: We experimented with simple RNNs, bidirectional RNNs (BiRNN), LSTMs and GRUs. To aggregate information across time steps we either used the final hidden state or applied GlobalAveragePooling1D over all time steps. Model sizes ranged from 256–512 recurrent units. Recurrent dropout (0.1–0.2) and L2 weight regularization mitigated over-fitting. Early stopping and reduce-on-plateau callbacks controlled training epochs.

D. Evaluation Metrics

Models were evaluated on the validation and test splits using four metrics: accuracy, macro-averaged F1, weighted F1 and the macro-averaged area under the ROC curve (AUC). Confusion matrices and one-versus-rest ROC curves provide fine-grained insight into per-class performance.

III. Results and Discussion

A. Classical Baselines

Table I summarizes the performance of ML models on BoW and TF-IDF representations. Logistic regression with TF-IDF clearly outperformed BoW and random forest. MNB offered competitive accuracy and F1 but was marginally worse than LR. The random forest suffered from high-dimensional sparsity despite SVD compression.

TABLE I
Performance of classical ML models on validation (val) and test sets. TF-IDF consistently outperforms BoW; best results in bold.

Model & Representation	Val		Test	
	Acc	Macro-F1	Acc	Macro-F1
LR + BoW	0.665	0.665	–	–
LR + TF-IDF	**0.702**	**0.699**	**0.703**	**0.701**
MNB + BoW	0.686	0.682	–	–
MNB + TF-IDF	0.693	0.688	0.692	0.688
RF + TF-IDF (SVD-300)	0.592	0.587	0.594	0.590

To illustrate misclassification patterns, Fig. 1 shows the validation confusion matrix for the Logistic Regression + TF-IDF model. The darker diagonal indicates correct predictions, while off-diagonal blocks (e.g., confusion between classes 0 and 8) highlight overlap in vocabulary across some categories.

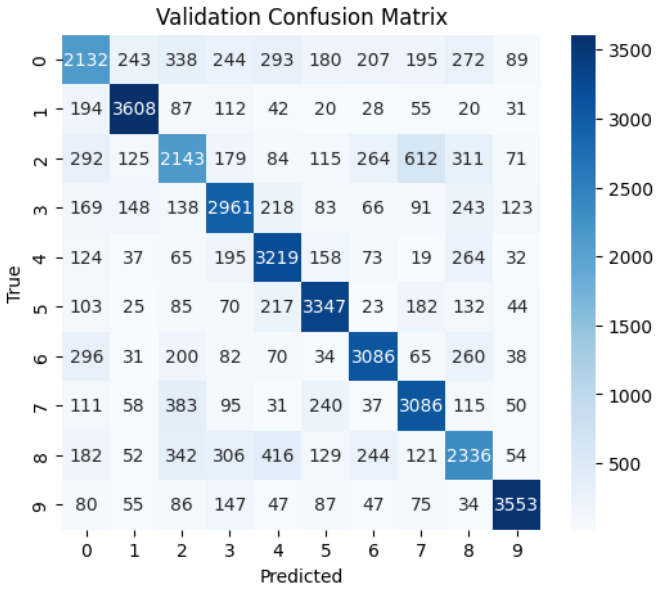

Fig. 1. Validation confusion matrix for Logistic Regression + TF-IDF. Most errors occur between neighbouring classes that share similar lexical patterns.

B. Deep Neural Networks

Table II compares neural models trained on dense representations. A two-layer DNN with GloVe achieved a validation accuracy of 0.710 and macro-F1 of 0.703, while the same architecture on Skip-gram embeddings slightly improved test performance. Bidirectional RNNs with Skip-gram performed

nearly as well but lagged behind DNNs, likely because the task benefits more from global word presence than strict order. The LSTM model with trainable GloVe embeddings produced the highest accuracy (0.721) and macro-F1 (0.716) among recurrent models. GRU models achieved similar training accuracy but exhibited larger gaps between training and validation accuracy, indicating over-fitting (see Fig. 2).

TABLE II
NEURAL NETWORK RESULTS ON VALIDATION AND TEST SETS. ACCURACIES AND MACRO-F1 SCORES ARE REPORTED. THE BEST VALUE WITHIN EACH COLUMN IS BOLDED.

Model & Representation	Val		Test	
	Acc	Macro-F1	Acc	Macro-F1
DNN + TF-IDF (SVD-768)	0.673	0.666	–	–
DNN + GloVe (100d)	0.710	0.703	0.711	0.705
DNN + Skip-gram (300d)	–	–	**0.712**	**0.707**
BiRNN + Skip-gram	0.709	0.705	0.710	0.705
BiRNN + GloVe	0.702	0.701	0.704	0.701
SimpleRNN + Skip-gram	0.700	0.699	0.701	0.699
SimpleRNN + GloVe	0.693	0.687	0.693	0.687
LSTM + GloVe	**0.719**	**0.714**	**0.721**	**0.716**

C. GRU Analysis

GRU networks were trained with both Skip-gram and GloVe embeddings. Although training accuracy exceeded 0.84, the validation accuracy stagnated around 0.69–0.72, and macro-F1 did not surpass the DNN baseline. Figure 2 plots training and validation accuracy for a representative GRU run. The widening gap after the third epoch indicates over-fitting despite the use of dropout and weight decay. Incorporating global pooling (e.g., average pooling across time steps) or using bidirectional architectures may help mitigate this issue.

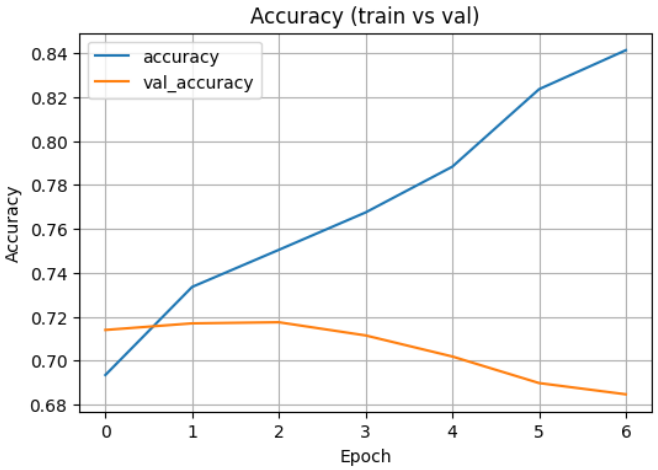

Fig. 2. Training and validation accuracy over epochs for a GRU model (Skip-gram, 512 units). While training accuracy climbs steadily to 0.84, validation accuracy peaks around 0.72 then declines, showing over-fitting.

Confusion matrices and ROC curves provide further insight. The test confusion matrix for a GRU model (Fig. 3) shows a strong diagonal but noticeable confusion between classes 2 and 8. The macro- average ROC curve (Fig. 4) yields an AUC of 0.948, comparable to the DNN.

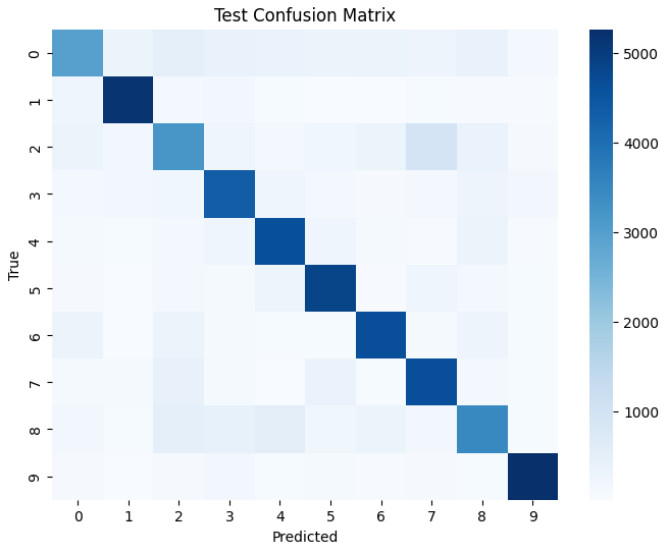

Fig. 3. Test confusion matrix for a GRU model (Skip-gram). Most mis-classifications occur between semantically similar categories.

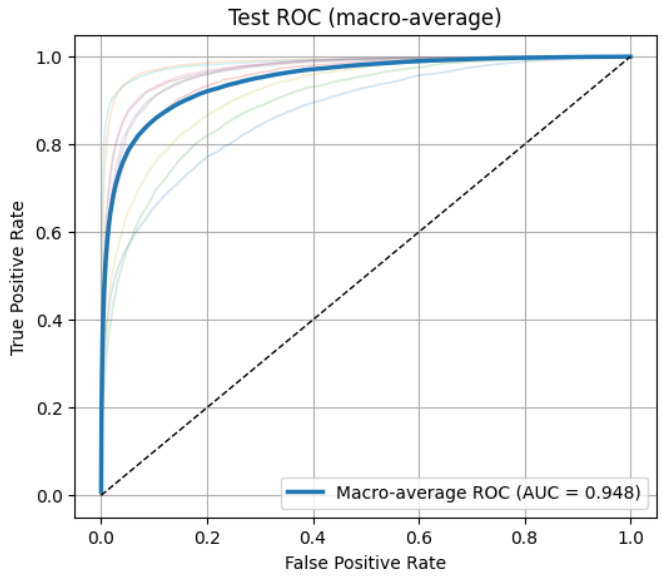

Fig. 4. Macro-averaged ROC curve for the GRU on the test set. The AUC of 0.948 indicates strong ranking performance across classes.

IV. CONCLUSION AND FUTURE WORK

This project compared classical and neural models for multi-class question–answer classification under four word-representation schemes. Consistent with prior literature, TF-IDF surpassed BoW for linear models, and logistic regression with TF-IDF served as a strong baseline ($\approx 70\%$ accuracy). Dense word embeddings enabled more expressive networks: a two-layer DNN with Skip-gram embeddings

achieved the highest test accuracy and macro-F1 ($\approx 71\%$). Bidirectional RNNs performed competitively but did not meaningfully surpass the DNN, suggesting that word order carries limited additional information for this task. LSTM had the best performance overall securing 0.72 accuracy. GRU models tended to over-fit, highlighting the importance of regularization and architectural choices.

Future work could explore transformer-based models such as BERT or RoBERTa, which have delivered state-of-the-art results on many NLP tasks. Data augmentation and semi-supervised learning might alleviate class imbalance and improve performance on rare categories. Finally, a more fine-grained error analysis could reveal dataset biases and guide curriculum improvements.

ACKNOWLEDGMENT

We thank the CSE440 course faculties for providing the dataset and helpful feedback.

REFERENCES

[1] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.

[3] C. D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[4] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] T. Chollet *et al.*, "Keras," GitHub repository, 2015. [Online]. Available: https://keras.io

[6] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org

[7] R. Rehurek and P. Sojka, "Gensim – Statistical semantics in Python," *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, 2010.

[8] W. McKinney, "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, pp. 51–56, 2010. (pandas library)

[9] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[10] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.