Advances in Artificial Intelligence and Machine Learning; Research 4 (3) 2746-2763 Received 5-07-2024; Accepted 23-09-2024; Published 30-09-2024

Evaluation of Machine Learning Techniques for Motivational Quotes Classification and Categorization

Adhiveer Kapuria adhiveer.kapuria@pathways.in
Pathways School
Gurugram, HR 98195-4322, INDIA

Parth Bhavsar parth19diat@gmail.com
Defence Institute of Advanced Technology
Pune, MH 411025, INDIA

Nishant Kejriwal kejriwalnishant1990@gmail.com
Indian Institute of Technology
Jodhpur, RJ 342030, INDIA

Corresponding Author: Adhiveer Kapuria

Abstract

The effective classification and categorization of motivational quotes are critical for various applications in content management and recommendation systems. This paper presents a comprehensive methodology for consolidating and categorizing motivational quotes using machine learning techniques. The dataset, sourced from Kaggle's Quotes-500K, contains over half a million quotes categorized into various tags such as ”Love,” ”Life,” ”Motivation,” and more. We address category redundancies by consolidating similar tags and selecting appropriate embedding models to capture semantic relationships between quotes. The embedding models are evaluated using t-SNE for finding the most distinctive model for representing motivational quotes. The final categorization focuses on three key categories: ”Love,” ”Inspirational,” and ”Humor”. We then evaluate the performance of several AI models, including Neural Networks, K-Nearest Neighbors (KNN), Euclidean distance-based classifiers, and Cosine Similarity methods. The proposed improved neural network model outperformed other models with a test accuracy of approximately 98%. These results suggest that our approach offers a robust solution for scalable and accurate categorization of motivational content. Future work will explore deeper NLP techniques, fine-tuning strategies for not just increased number of categories but further enhancing the categorization accuracy along with ranking them within their category.

Keywords: Motivational Quotes, Machine Learning, Quote Categorization, Embedding Models, K-NN nearest neighbour, Neural Networks

Citation: Adhiveer Kapuria, et al. Evaluation of Machine Learning Techniques for Motivational Quotes Classification and Categorization. Advances in Artificial Intelligence and Machine Learning. 2024;4(3):160.

1. INTRODUCTION

Motivational quotes have a profound impact on personal growth, emotional well-being, and social interactions, often shared widely across digital platforms. The sheer volume and diversity of such quotes pose challenges for effective classification and categorization, which are crucial for content organization and recommendation systems. Despite the prevalence of motivational quotes, there is limited research on systematically categorizing them using advanced machine learning techniques. Previous studies have primarily focused on sentiment analysis or generic text classification but have not specifically addressed the categorization of motivational quotes into distinct, meaningful categories. This research seeks to bridge this gap by exploring the following research question.

1.1 Research Question

How can motivational quotes be effectively categorized using machine learning techniques to improve classification accuracy and category distinctiveness?

1.2 Objectives

The primary objectives of this study are:

1. To consolidate and reduce redundancy in categories by merging semantically similar tags.
2. To evaluate and compare the performance of various embedding models for categorizing quotes.
3. To develop and optimize machine learning models that accurately classify quotes into distinct categories
4. To assess the effectiveness of a neural network model for this classification task and determine its performance relative to other approaches

2. PREVIOUS WORK

Research specifically and directly focused on the multi-class categorization of motivational quotes into distinct and granular categories is very limited. However, sentence classification is a fundamental task in natural language processing (NLP) that involves assigning predefined categories to sentences. It has applications in sentiment analysis, spam detection, topic categorization, and intent classification in dialogue systems. With advancements in machine learning, particularly deep learning, there have been significant improvements in sentence classification models in recent years.

Lexicon-based approaches were among the earliest methods used for sentence classification. These approaches rely on sentiment lexicons like SentiWordNet [1], or AFINN [2], where words have associated sentiment scores. Lexicon-based methods often struggle with ambiguity and context, but they are still used in systems where interpretability is critical.

Machine learning approaches, particularly Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression, have also been used for sentence classification tasks. These methods treat sentences as feature vectors, typically using bag-of-words or TF-IDF [3], representations. Joachims [4], demonstrated the effectiveness of SVMs for text classification, including sentences, by transforming text into high-dimensional feature spaces. Deep learning techniques have revolutionized sentence classification by enabling models to capture semantic and syntactic information more effectively. Models like Recurrent Neural Networks (RNNs) [5], and Long Short-Term Memory (LSTM) [6], networks excel in capturing sequential dependencies in sentences, making them suitable for complex sentence-level tasks. Kim et al. [7] demonstrated the use of Convolutional Neural Networks (CNNs) for sentence classification, where convolutional filters capture local dependencies between words. CNNs were found to be highly effective for short-text classification due to their ability to learn hierarchical features.

With the introduction of Transformer models, particularly BERT (Bidirectional Encoder Representations from Transformers), sentence classification tasks have seen a dramatic improvement. Devlin et al. introduced BERT [8], which uses bidirectional training to capture context from both directions in a sentence. BERT has been fine-tuned for various sentence classification tasks with state-of-the-art performance. More recent transformer models like RoBERTa [9], and ALBERT [10], further optimized transformer architectures, achieving faster training times and higher accuracy. Research by Sun et al. [11], examined fine-tuning strategies for BERT on text classification tasks, showing the importance of task-specific adaptation. While state-of-the-art models like BERT and RoBERTa have set new benchmarks in sentence classification, research is ongoing to improve efficiency and accuracy, particularly for low-resource languages and specific tasks with limited labeled data.

3. METHODOLOGY

Classifying small text messages, like quotes, using machine learning involves a structured process, a combination of multiple methods to optimize accuracy and efficiency. The initial step is data preprocessing, including text cleaning (removal of stop words and punctuation) and normalization (lowercasing, stemming) [12], to ensure uniformity across the dataset. This is critical for reducing noise and enhancing the quality of features. Next, feature extraction techniques, such as TF-IDF [3], bag-of-words, or embeddings (Word2Vec [13], BERT [8]), are used to convert the text into numerical vectors. Various algorithms can be applied, including Naive Bayes [14], for simplicity, SVM for high-dimensional spaces [15], K-Nearest Neighbors (KNN) for proximity-based classification [16], and neural networks (LSTM, CNN) [7] for deeper contextual understanding. The system also emphasizes model evaluation, employing metrics like accuracy and precision to assess model performance. These metrics provide a holistic view of how well the model generalizes to unseen data. To further enhance model performance, hyperparameter tuning for optimization [17] are implemented. Tuning helps the model learn more effectively, leading to better performance on unseen data, which is crucial when working with limited datasets like motivational quotes.

This combination of preprocessing, feature extraction, and algorithm selection forms a robust, multimethod system that maximizes performance and adaptability to different classification tasks. Each method is chosen based on its strengths and relevance to the dataset, resulting in an efficient and accurate text classification system.

3.1 Dataset Preparation

The dataset, sourced from Kaggle's Quotes-500k [18], contains quotes, their authors, and category tags. Initially, the data required preprocessing, including resolving category redundancies. Categories like "Humor" and "Humorous" were merged to ensure consistency. After consolidation, the categories "Love," "Inspirational," and "Humor" were selected for focused analysis. The final cleaned dataset was structured into distinct CSV files for each category.

3.2 Embedding Models and Feature Extraction

To convert text into meaningful numerical representations, several embedding models were evaluated. Feature extraction is essential for converting raw text into numerical representations that machine learning models can process. In this research, several state-of-the-art embedding models were employed to capture the semantic richness of motivational quotes. These models include $all-MiniLM-L6-v2$[3], $all-mpnet-base-v2$[19], $facebook-dpr-ctx-encoder-multiset-base$[20], and $paraphrase-MiniLM-L6-v2$. Each model provides unique strengths in encoding the semantic relationships within the text, crucial for accurate classification. By representing each quote as a dense numerical vector, these models enable downstream machine learning algorithms to capture semantic similarities, differences, and contextual relationships between quotes.

3.2.1 *$all-MiniLM-L6-v2$*

The *$all-MiniLM-L6-v2$* model is a distilled version of Microsoft's MiniLM, designed to offer high-quality sentence embeddings with minimal computational resources. It uses 6 transformer layers and outputs a 384-dimensional embedding. Despite its compact size, it retains strong performance for tasks requiring semantic similarity and sentence classification, making it ideal for scenarios with limited computational power but demanding real-time applications.

3.2.2 *$all-mpnet-base-v2$*

The *$all-mpnet-base-v2$* model is a part of the MPNet family, which incorporates both masked language modeling and permuted language modeling. It improves upon BERT by learning better dependency structures between tokens. This results in better contextual embeddings, making it highly effective for capturing subtle nuances in motivational quotes and other short texts.

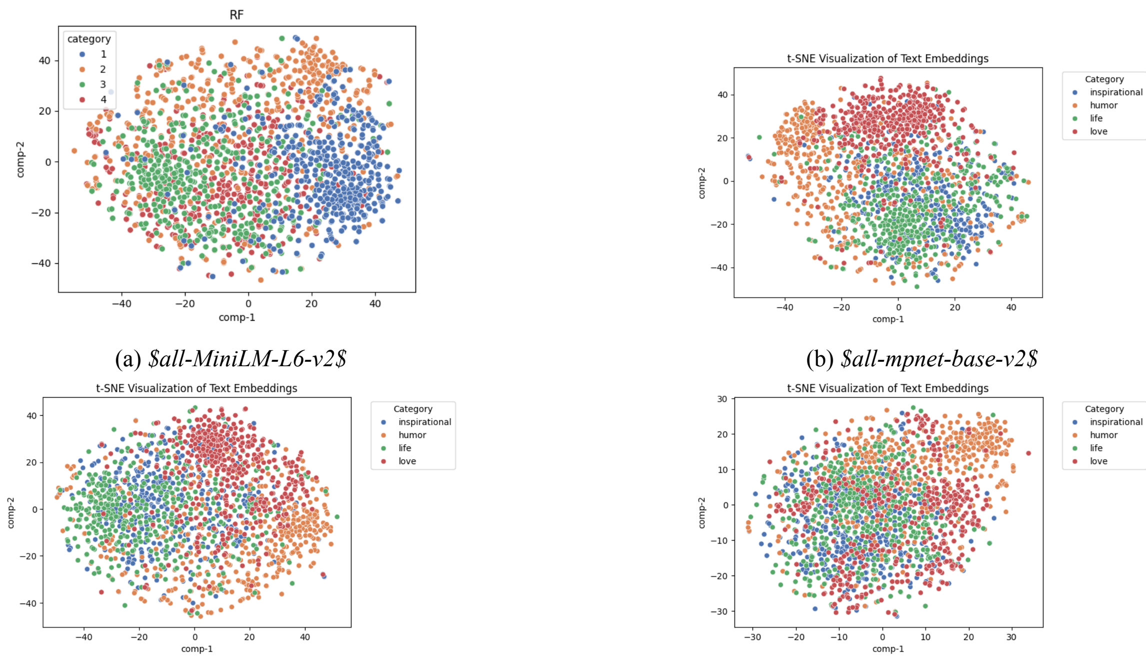
3.2.3 *$facebook-dpr-ctx-encoder-multiset-base$*

The *$facebook-dpr-ctx-encoder-multiset-base$*is part of Facebook's Dense Passage Retrieval (DPR) system, which is optimized for retrieval tasks. It encodes context passages (e.g., sentences) into dense embeddings that can be efficiently searched for semantic similarity. This model is especially useful for tasks requiring fast and accurate retrieval of semantically similar quotes.

3.2.4 *$paraphrase-MiniLM-L6-v2$*

The *$paraphrase-MiniLM-L6-v2$* model, like its sibling *$all-MiniLM-L6-v2$*, is designed for efficiency while preserving semantic meaning. It excels at paraphrase detection, making it particularly suitable for categorizing quotes with similar meanings but different wording. This allows the classification system to group motivational quotes that express the same underlying message.

These models were tested using t-SNE, [21] a dimensionality reduction technique that effectively visualizes high-dimensional data, as shown in FIGURE 1, for all of the embeddings.



(a) *$all-MiniLM-L6-v2$* (b) *$all-mpnet-base-v2$*

(c) *$paraphrase-MiniLM-L6-v2$* (d) *$facebook-dpr-ctx-encoder-multiset-base$*

Figure 1: t-SNE based visualizations for various embeddings

The t-SNE analysis revealed that categories like ”Life” and ”Inspirational” were inseparable and therefore required merging or exclusion from further analysis. The t-SNE visualization helped us in selecting the final representation model for Motivational quotes based on separability of the different classes.

3.3 Category Consolidation and Model Selection

The remaining categories, “Love,” “Inspirational,” and “Humor,” were used for model training. Various machine learning models were then employed for classification, including:

1. **Neural Network (NN):** A multi-layer feedforward network with ReLU activations and dropout layers.

2. **K-Nearest Neighbor (KNN):** Evaluated with the optimal value of k, with a voting scheme based on maximum votes in the k nearest neighbors for each query.

3. **Centroid based classification:** Nearest neighbour using centroid of clusters with euclidean distance.

4. **Cosine Similarity based classification:** Nearest neighbour using the angular similarity between vectors.

3.4 Model Training and Evaluation

The dataset was split into 80% training and 20% testing subsets. For the neural network, the architecture included multiple hidden layers with dropout [22] and batch normalization [23]. The model was optimized using the Adam optimizer [24] and CrossEntropyLoss [25], with a ReduceLROnPlateau scheduler [26] for dynamic learning rate adjustment. The training loop was set for 100 epochs, with performance tracked using accuracy, precision, and recall.

4. ORIGINAL NEURAL NETWORK EVALUATION AND RESULTS

The original neural network (NeuralNet) was designed as a fully connected feedforward model with three hidden layers. The input consisted of 768-dimensional feature vectors, followed by an expansion in the first hidden layer to 1536 dimensions and further to 2304 dimensions in the second layer, with a reduction back to 1536 dimensions in the third hidden layer. The detailed architecture of the original neural network is shown in FIGURE 2. ReLU activation functions [27], and dropout layers (30%) were applied to prevent overfitting. During training, the network was optimized using CrossEntropyLoss and the Adam optimizer with a learning rate of 0.001 and weight decay of 0.01. A learning rate scheduler (StepLR) was employed, reducing the rate every 10 epochs.

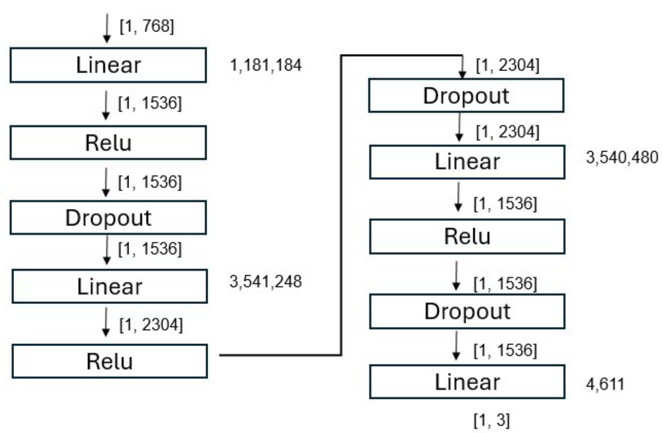


Figure 2: Original Neural Network Architecture

4.1 Training Performance

The training accuracy rapidly increased from 0.50 to over 0.80 within the first 10 epochs, stabilizing around 0.85 to 0.87 after 20 epochs. The validation accuracy followed a similar trend, reaching 0.80 by the 10th epoch and stabilizing just below 0.85. Although there was a slight gap between training and validation accuracy [28], as shown in FIGURE 3, it was minimal, indicating that overfitting was controlled.

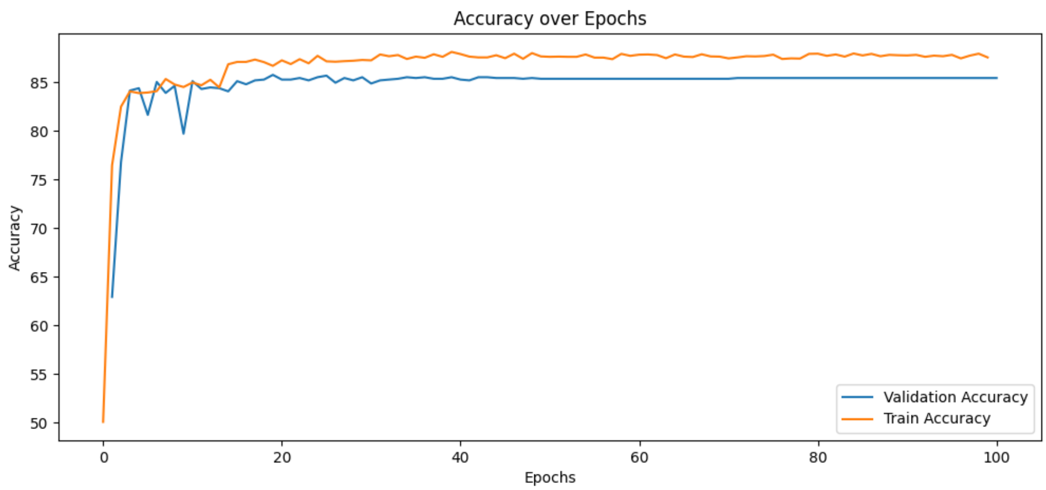


Figure 3: Training and validation accuracy

4.2 Testing Results

Testing was conducted on a hold-out dataset comprising 600 samples, with 200 samples from each category. The model achieved a test accuracy of 88.87%, with precision and recall also around 88.87%. The confusion matrix showed a balanced distribution of correctly classified quotes, highlighting the model's overall strong performance as shown in FIGURE 4. The results indicate balanced performance across all categories, with the "Love" class achieving the highest recall and F1-score, as shown in TABLE 1. Despite some minor misclassifications, the results suggest that the model generalized well to unseen data.

Class	Precision	Recall	F1-Score
Humor	0.88	0.90	0.89
Inspirational	0.89	0.83	0.86
Love	0.89	0.94	0.91

Table 1: Class-wise Performance Metrics for the Original Neural Network

Figure 4: Testing Confusion matrix

5. KNN EVALUATION AND RESULTS

The K-Nearest Neighbors (KNN) algorithm [16] was used to classify the quotes based on their feature embeddings. KNN is a simple yet powerful non-parametric algorithm that classifies data points based on their proximity to the 'k' nearest neighbors in feature space. For this classification task, each quote was represented as a feature vector in a high-dimensional space using embeddings. The Euclidean distance was computed between the feature vectors to measure their proximity, and the majority class among the nearest neighbors determined the classification. To improve model performance, data standardization was applied, which ensures that all features have the same scale, allowing the distance calculations to be more meaningful. Standardization is particularly important in KNN, as it ensures that features with larger ranges do not disproportionately affect the distance computation.

Hyperparameter tuning was conducted to identify the optimal value of 'k'—the number of nearest neighbors used for classification. A grid search method was employed to explore different values of 'k', and cross-validation (cv=5) was used to assess the performance of each configuration across different subsets of the data. This ensures that the model generalizes well to unseen data. The results showed that the best accuracy was achieved at k=8, beyond which performance started to plateau. At k=8, the algorithm strikes a balance between overfitting (lower values of 'k') and underfitting (higher values of 'k'), allowing for robust and stable classification, as shown in FIGURE 5.

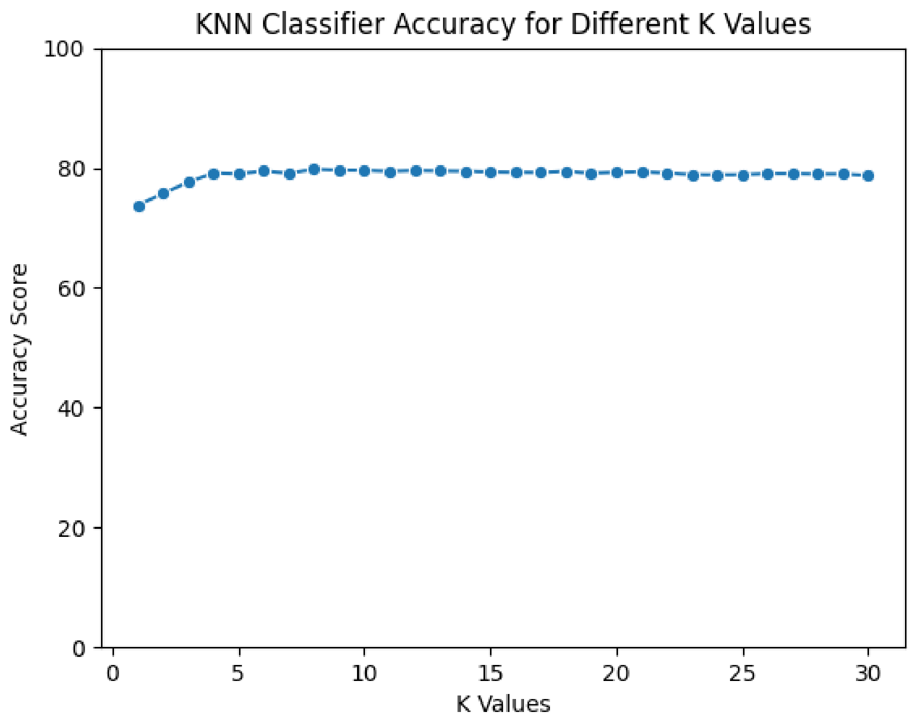


Figure 5: Training and validation accuracy

The final class of the quote is decided based on the class which receives the maximum votes amongst the K nearest neighbors.

5.1 Testing Results

Testing was conducted on a dataset comprising 600 samples, with 200 samples from each category. The model achieved a test accuracy of 84.4%, with precision of 85.0% and recall of 84.4%. The confusion matrix showed a balanced distribution of correctly classified quotes, highlighting the model's overall strong performance, as shown in FIGURE 6. These class-wise metrics suggest that while "Humor" had lower recall, the other classes performed consistently well across all metrics, as shown in TABLE 2. While some misclassifications occurred, they were minimal and did not significantly affect overall performance.

Class	Precision	Recall	F1-Score
Humor	0.92	0.76	0.83
Inspirational	0.81	0.87	0.84
Love	0.82	0.91	0.86

Table 2: Class-wise Performance Metrics for KNN

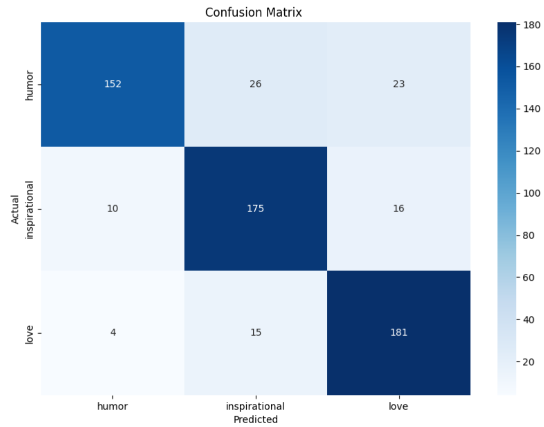


Figure 6: Testing Confusion matrix

6. CENTROID BASED CLASSIFICATION EVALUATION AND RESULTS

The centroid-based classification method [29] was used to categorize quotes by computing the Euclidean distance between test instances and predefined centroids [30]. Each centroid represents the mean feature vector for a given category. During classification, the test instances are assigned to the category whose centroid is nearest based on the smallest Euclidean distance. The distance is computed using the formula:

$$Distance(p, q) = \sqrt{\sum_{i}^{n} (p_i^2 - q_i^2)}$$

In centroid-based classification, the performance heavily depends on the representation of centroids, which are calculated as the mean vectors of all training instances for each class.While there are no traditional hyperparameters to tune (such as learning rates or layers), the primary concern is selecting the right embedding model and ensuring that centroids accurately represent each category. Care was taken to avoid category overlap, as highly similar categories could lead to misclassification due to proximity in the vector space.

6.1 Testing Results

The model was evaluated on the same test dataset used in previous methods. The classification performance was Test Accuracy, Precision and Recall were all 85%. These metrics indicate that the centroid-based classification is effective, achieving a balance between simplicity and accuracy, as shown in FIGURE 7. The use of Euclidean distance provides a straightforward approach for categorizing the quotes based on their embeddings. These metrics reflect strong performance, with especially balanced results for the “Love” class. While some minor misclassifications occurred, particularly in the "Inspirational" class, the model generalized well across all categories, as shown in TABLE 3.

Class	Precision	Recall	F1-Score
Humor	0.79	0.89	0.84
Inspirational	0.85	0.75	0.80
Love	0.90	0.90	0.90

Table 3: Class-wise Performance Metrics for Centroid-Based Classification

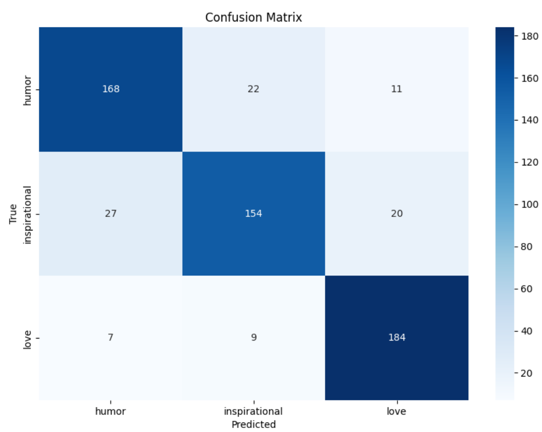


Figure 7: Centroid based testing confusion matrix

7. COSINE SIMILARITY BASED CLASSIFICATION EVALUATION AND RESULTS

Cosine similarity measures the cosine of the angle between two vectors in a multidimensional space [31], effectively evaluating how similar two embeddings are, irrespective of their magnitude. For each test instance, cosine similarity is computed between the instance's feature vector and each centroid vector. The formula is:

$$CosineSimilarity = \frac{A.B}{||A||||B||}$$

Where, A and B are the two vectors being compared. A.B is the dot product of vectors A and B. ||A|| and ||B|| are the magnitudes (or norms) of vectors A and B.

Here, the dot product evaluates similarity based on directionality, making it suitable for high-dimensional data. The category with the highest cosine similarity is selected. This approach achieved a test accuracy, precision, and recall of 84%, showing consistent performance across metrics, as shown in FIGURE 8. The Love class has the strongest performance, with both precision and recall, making it the most well-classified category. The Humor and Inspirational classes lower performance across all metrics, suggesting that the model may occasionally miss classify, as shown in TABLE 4.

Class	Precision	Recall	F1-Score
Humor	0.83	0.84	0.83
Inspirational	0.83	0.77	0.80
Love	0.86	0.92	0.89

Table 4: Class-wise Performance Metrics for Cosine Similarity-Based Classification

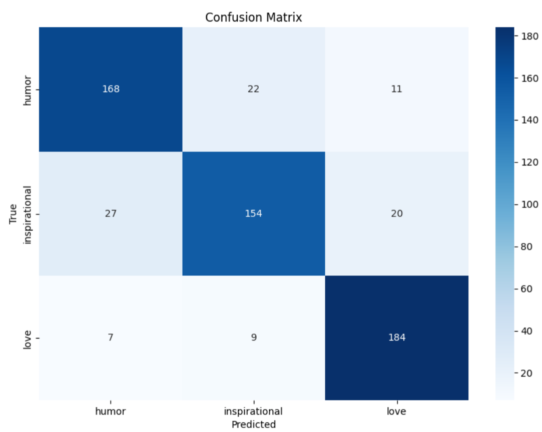


Figure 8: Cosine Similarity testing confusion matrix

8. SUMMARY OF RESULTS

The comparison of model performances reveals that the neural network achieved the highest accuracy at 88%, making it the most effective among the tested models, as shown in TABLE 5. The Euclidean distance-based model followed with 85% accuracy, while both the KNN and cosine similarity models achieved 84%. These findings emphasize that while simpler models like KNN and cosine similarity perform well, the neural network's more complex architecture offers better overall accuracy for classifying motivational quotes. Based on deployment platform, we can select either of the above classifier keeping a balance between accuracy and speed.

Technique	Accuracy
Original Neural Network	88%
KNN with k neighbors voting	84%
Centroid based euclidean classification	85%
Cosine similarity based classification	84%

Table 5: Summary of results

9. IMPROVED NEURAL NETWORK ARCHITECTURE AND EVALUATION

In order to optimize the performance of the neural network we changed the following in the previous model. The updated architecture is as shown in FIGURE 9.

1. **Architecture Complexity**: More sophisticated, with four hidden layers instead of three that progressively increase dimensionality from 768 to 1536, then reduce it back to 512. Each hidden layer includes batch normalization, ReLU activation, and higher dropout rates (up to 30%) for better regularization [22].
2. **Learning Rate Adjustment**: Replaced the basic StepLR scheduler with a more dynamic ReduceLROnPlateau scheduler instead of the that adjusts the learning rate based on validation accuracy, enabling finer control over optimization [26].
3. **Dropout and Regularization**: Utilized variable dropout rates (25%-30%) tailored to each layer's complexity, improving model generalization, instead of fixed 30% dropout.

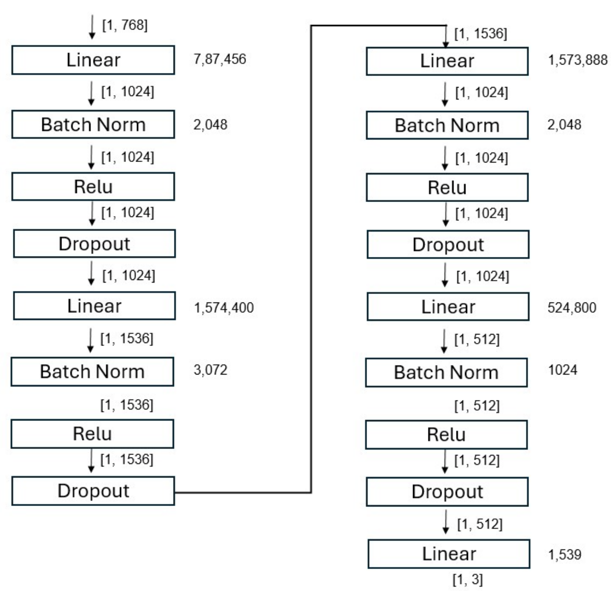


Figure 9: Improved Neural Network Architecture

9.1 Training and Validation Performance

The improved neural network model underwent hyperparameter tuning, particularly with respect to the learning rate. The initial learning rate was set to 0.0001, ensuring stable learning during the early stages of training. After 50 epochs, the learning rate remained constant, as the validation metrics (such as validation accuracy or loss) continued to improve, preventing an early reduction. However, after 80 epochs, the learning rate was reduced to 0.00001 as shown in FIGURE 10. This decrease occurred due to the validation metric plateauing, triggering the learning rate scheduler to lower the rate, allowing for finer adjustments to the model weights in the later stages of training. This gradual reduction in learning rate helped the model avoid overshooting and improved its convergence during training.

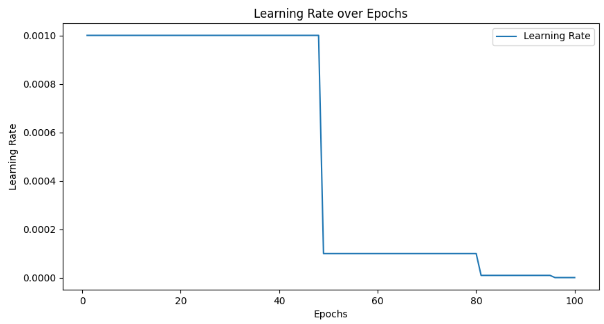


Figure 10: Improved Neural Network Learning Rate Tuning

The training accuracy rapidly increased from 0.50 to over 0.80 within the first 10 epochs, stabilizing around 0.85 to 0.87 after 20 epochs. Validation accuracy showed a consistent and gradual increase, stabilizing just below 0.85 after the initial surge, as shown in FIGURE 11. The minimal gap between training and validation accuracy indicates a well-regularized model with controlled overfitting. The steady performance during the later epochs shows that the model generalized well to unseen data.

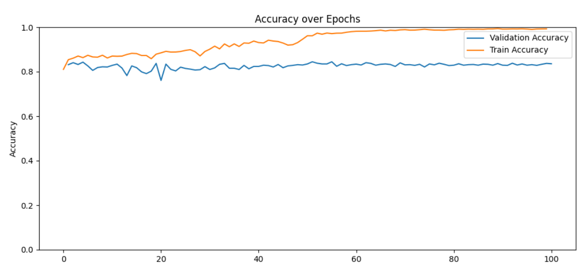


Figure 11: Improved Neural Network Training

9.2 Testing Results

Testing was performed on a hold-out dataset identical to that used in other evaluations. The model achieved an increased test accuracy of 97.84%, as shown in FIGURE 12. The confusion matrix indicated balanced classification across categories, with high precision and recall values, confirming the model's capability to handle the complexities of the motivational quote classification task. The model demonstrated highly effective performance across all classes, with consistently high precision, recall, and F1-scores as shown in TABLE 6.

Class	Precision	Recall	F1-Score
Humor	0.94	0.96	0.95
Inspirational	0.97	0.94	0.95
Love	0.97	0.98	0.98

Table 6: Class-wise Performance Metrics for the Improved Neural Network

These results indicate that the model is well-optimized for the task, particularly excelling in identifying "Love" quotes. The overall balance between precision and recall across categories demonstrates that the model generalizes well and maintains high accuracy in its classifications.

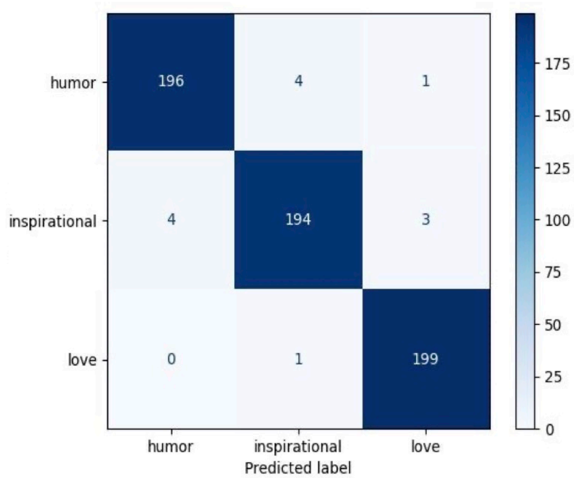


Figure 12: Improved Neural Network Confusion Matrix

This enhanced architecture and learning strategy emphasize the importance of adaptive optimization and deep regularization in achieving state-of-the-art performance.

10. CONCLUSION AND FUTURE WORK

The study successfully developed an improved neural network architecture for classifying motivational quotes, achieving a test accuracy of 97.84%. The enhancements, including dynamic learning rate adjustment and more advanced regularization techniques, were instrumental in outperforming simpler models like KNN and centroid-based classifiers. Despite these successes, there remain areas for further exploration.

1. **Evaluating Other Machine Learning Techniques**: Future studies could explore integrating more advanced machine learning models to improve classification performance. Transformer-based models LLMs [32] like GPT-3 could enhance the contextual understanding of quotes. Ensemble learning methods, such as Random Forests or XGBoost, could combine the strengths

of different algorithms for better results. Hybrid architectures that merge CNNs, RNNs [33] with LSTMs could also improve classification of short texts and sequential dependencies.

2. **Expanding the Dataset**: Incorporating a larger and more diverse dataset could enhance the model's ability to generalize across different types of motivational content. Additionally, exploring multi-lingual datasets could make the model more robust in various languages.
3. **Fine-Grained Category Classification and Ranking**: Introducing more granular categories could improve the model's precision. For instance, subcategories under broader themes like ”Love” or ”Inspirational” could be identified, allowing for more nuanced classification. Ranking of motivation quotes within their category can also be explored in future.
4. **Integration with Real-World Applications**: Deploying this model in real-world scenarios such as recommendation systems or content management tools could validate its practical effectiveness. Performance could be monitored in dynamic environments, where user interactions provide continuous feedback for model refinement. Further this could be part of daily affirmations and mental well-being apps.
5. **Exploration of Explainability and Interpretability**: As the model is applied in sensitive contexts, developing mechanisms for interpreting its decisions could be crucial. Explainable AI techniques might be integrated to ensure that the model's predictions are transparent and easily understood by non-expert users.

This future direction aims to make the motivational quote classification system more versatile, applicable across diverse contexts, and aligned with the growing demand for AI transparency.

References

[1] Baccianella S, Esuli A, Sebastiani F. Sentiwordnet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proceedings of the seventh international conference on language resources and evaluation (LREC'10). 2010:2200-2204.

[2] Nielsen FÅ. AFINN: A New Word List for Sentiment Analysis. In: Proceedings of the ESWC 2011 workshop on 'Making Sense of Microposts': Big things come in small packages. 2011:93-98.

[3] Reimers N, Gurevych I. Sentence-Bert: Sentence Embeddings Using Siamese Bertnetworks. In: Proceedings of the 2019 conference on empirical methods in natural language processing (EMNLP). 2019:3982-3992.

[4] Joachims T. Text Categorization With Support Vector Machines: Learning With Many Relevant Features. In: European conference on machine learning. Springer. 1998:137-142.

[5] Rumelhart DE, Hinton GE, Williams RJ. Learning Representations by Back-Propagating Errors. Nature. 1986;323:533-536.

[6] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9:1735-1780.

[7] Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014:1746-1751.

[8] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pretraining of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.Association for Computational Linguistics. 2019:4171–4186.

[9] Liu Y, Ott M, Goyal N, Du J, Joshi M, et al. Roberta: A robustly optimized bert pretraining approach.2019. ArXiv preprint: https://arxiv.org/pdf/1907.11692

[10] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, et al. Albert: A Lite Bert for Self-Supervised Learning of Language Representations. 2020. ArXiv preprint: https://arxiv.org/pdf/1909.11942

[11] Sun C, Qiu X, Xu Y, Huang X. How to Fine-Tune Bert for Text Classification? In: Proceedings of the 2019 China national conference on Chinese computational linguistics. 2019:194-206.

[12] Salton G, McGill MJ. Vector Space Model for Automatic Indexing. Addison-Wesley. 1983.

[13] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. In: Proceedings of the international conference on learning representations (ICLR). 2013.

[14] McCallum A, Nigam K. A Comparison of Event Models for Naive Bayes Text Classification. In: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization. 1998;752:41-48.

[15] Cortes C, Vapnik V. Support-Vector Networks. Mach Learn. 1995;20:273-297.

[16] Cover T, Hart P. Nearest Neighbor Pattern Classification. IEEE Trans Inf Theor. 1967;13:21-27.

[17] Bergstra J, Bengio Y. Random Search for Hyper-Parameter Optimization. J Mach Learn Res. 2012;13:281-305.

[18] https://www.kaggle.com/datasets/manann/quotes-500k

[19] Song K, Tan X, Qin T, Lu J, Liu TY. Mpnet: Masked and Permuted Pretraining for Language Understanding. Adv Neural Inf Process Syst (NeurIPS). 2020:16857-16867.

[20] Karpukhin V, Oğuz B, Min S, Lewis P, Wu L, et al. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics. 2020:6769–6781.

[21] van der Maaten L, Hinton G. Visualizing Data Using T-Sne. J Mach Learn Res. 2008;9:2579-2605.

[22] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. J Mach Learn Res. 2014;15:1929-1958.

[23] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd international conference on machine learning (ICML). 2015:448-456.

[24] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2015. ArXiv preprint: https://arxiv.org/pdf/1412.6980

[25] Rubinstein RY. The Cross-Entropy Method for Combinatorial and Continuous Optimization. Methodol Comput Appl Probab. 1999;1:127-190.

[26] Smith LN. Cyclical Learning Rates for Training Neural Networks. In: Proceedings of the IEEE winter conference on applications of computer vision (WACV); 2017:464-472.

[27] Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th international conference on machine learning (ICML). 2010:807-814.

[28] Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of the 14th international joint conference on artificial intelligence (IJCAI). 1995:1137-1143.

[29] Xu R, Wunsch D. Survey of Clustering Algorithms. IEEE Trans Neural Netw. 2005;16:645-678.

[30] Tan PN, Steinbach M, Kumar V. Introduction to Data Mining. Pearson. 2016.

[31] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Morgan Kaufmann. 2011.

[32] Sun X, Li X, Li J, Wu F, Guo S, Zhang T, Wang G. Text Classification via Large Language Models. 2023. ArXiv preprint; https://arxiv.org/pdf/2305.08377

[33] Eang SLC. Improving the Accuracy and Effectiveness of Text Classification Based on the Integration of the Bert Model and a Recurrent Neural Network (Rnn_bert_based). Appl Sci. 2022.