Comparative Analysis of Classical Machine Learning and Deep Neural Networks for Sogou News Classification

Wei Chen Huang*, Harine Choi†, Harshavardhan Reddy Varicuti‡, Shou-En Liu‡
*Department of Computer Science and Engineering, Texas A&M University
†Department of Multidisciplinary Engineering, Texas A&M University
‡Department of Electrical and Computer Engineering, Texas A&M University
College Station, Texas, United States
Emails: wilson0623@tamu.edu; harinec@tamu.edu; harshavaricuti@tamu.edu; shouenliu@tamu.edu

***Abstract*—This project examines several approaches for classifying news articles in the Sogou News dataset, which contains about 510,000 samples across five categories. We built a complete pipeline that includes dataset download from Hugging Face, data cleaning and normalization, exploratory analysis, and multiple supervised text classification models. Our main baselines use character-level TF-IDF features combined with three standard classifiers: Linear Support Vector Machines (LinearSVC), Multinomial Naive Bayes (MNB), and k-Nearest Neighbors (KNN). Trained on 450,000 examples and evaluated on the official 60,000-sample test split, the TF-IDF + LinearSVC model reached roughly 90% macro-F1 and 90% accuracy, outperforming the other classical baselines.**

We further explored word-level TF-IDF with LinearSVC using randomized search, grid search, and Bayesian hyperparameter optimization to tune n-gram and regularization settings. After hyperparameter tuning, the best TF-IDF + LinearSVC reached an accuracy of approximately 97% on test data. To compare these feature-engineering methods with representation learning, we also implemented two convolutional neural networks: a basic 1D CNN and a TextCNN model with multiple kernel sizes and batch normalization. Both neural models operate on tokenized title-and-content sequences and are trained end-to-end with learned embeddings. The overall framework provides a clear comparison between sparse TF-IDF models and CNN-based models, and it serves as a solid baseline for future work with deeper neural networks or transformer-based encoders.

I. Introduction

Text classification is widely used in many NLP applications, such as news filtering, topic grouping, and information retrieval. Although recent research often emphasizes deep learning, traditional methods based on TF-IDF features and classical classifiers still perform well on large datasets and offer practical advantages in speed and interpretability. In this project, we evaluate both classical and neural models on the Sogou News dataset, which consists of Chinese-language news articles labeled into five topical categories: Sports, Finance, Entertainment, Automobile, and Technology.

Chinese text presents several challenges because it typically does not include whitespace and may appear either in characters or in Romanized pinyin, depending on preprocessing. To avoid relying on language-specific segmentation tools, we start with a character-level TF-IDF representation using short n-grams. On top of these features, we train three classical classifiers: LinearSVC, Multinomial Naive Bayes, and KNN, -to compare margin-based, probabilistic, and distance-based learning in a high-dimensional sparse setting. Before training, we perform exploratory data analysis, including class-distribution plots, text-length statistics, example inspection, and a dimensionality-reduction visualization of TF-IDF features. We then extend the classical pipeline with word-level TF-IDF and LinearSVC, using randomized search, grid search, and Bayesian optimization to tune parameters such as minimum document frequency, maximum document frequency, and the SVM regularization constant.

To complement these baselines, we implement two convolutional neural networks: a 1D CNN and an optimized TextCNN model with multiple filter sizes and batch normalization. These models operate directly on tokenized and padded sequences derived from the combined title and content fields and are trained end-to-end with learned embeddings. Evaluating classical TF-IDF models and CNN-based models under the same data splits allows us to compare their performance and complexity in a controlled way. Overall, the character-level TF-IDF + LinearSVC model provides the strongest results, while the CNN models offer an alternative representation-learning approach and a path toward more advanced architectures in future work.

Previous work on Chinese text categorization has shown that Support Vector Machines and k-Nearest Neighbors perform strongly on high-dimensional Chinese news data, often outperforming earlier probabilistic methods [3]. Studies have also noted that segmentation inconsistencies in Chinese text can make character or n-gram features more reliable than dictionary-based tokenization [4]. These findings support our use of TF-IDF n-gram features and LinearSVC as competitive baselines for the Sogou News dataset.

II. Method

A. Data preprocessing

1) Data cleansing and transformation: The data used is the SougouNews dataset from the hugging face which is in pinyin. A custom normalization pipeline has been used to normalize the incoming text data. The normalize_text function performs the following operations using regular expressions:

a) URL removal: Patterns that match with www, http, https are identified and removed, so that the model does not overfit on specific hyperlinks.

b) Collapse whitespace: Different whitespace characters such as tab, space, newlines are collapsed to single space to reduce noise in the data.

2) Data Splitting: The Sougounews dataset has 5,10,000 rows. By default, when the Sougounews dataset is imported and loaded, it is split into two classes: train and test, with train having 4,50,000 rows, and test having 60,000 rows of data. We further split the train data into train and validation sets using train_test_split from sklearn with random_state=42 to allow reproducibility. , with 10% of the original train data as validation data. So, the final split is

Training set : 450,000
Validation set : 45,000
Test set: 60,000

B. Exploratory Data Analysis

To understand the structure, quality and separability of data before training the models Exploratory Data Analysis was done. The EDA pipeline was divided into four distinct phases:

1) Class Distribution Analysis: To understand if there could be potential class imbalances that could induce bias into the model, the distribution of target variables within the training data set was done and visualized.

Fig. 1: Class Distribution of the Training Set.

The class distribution analysis as seen in Fig. 1 reveals that the training dataset is perfectly balanced, with each class having 90,000 articles. Because the classes are equal, the model will not be biased toward a majority class which is a common issue in text classification where one topic dominates.

2) Sequence Length Distribution: A histogram with a Kernel Density Estimate (KDE) was generated to analyze the distribution of article lengths (in terms of characters) after preprocessing. Descriptive statistics (mean, median, min, max) were calculated to assess the variance in document size. The visualization as seen in Fig. 2 was explicitly clipped to the 3,000-character limit to observe how much data density existed near the truncation threshold.

TABLE I: Descriptive Statistics for Text Length

Statistic	Value
Count	450,000
Mean	1,882.27
Std	1,089.85
Min	20.00
25%	721.00
50%	2117.00
75%	3000.00
Max	3000.00

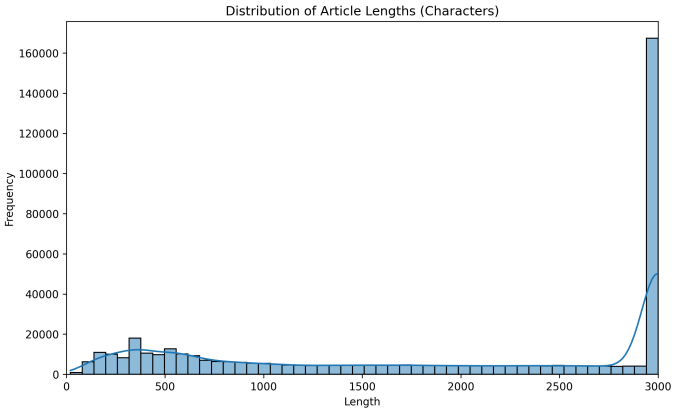

Fig. 2: Distribution of Article lengths (Training Set)

3) Qualitative Data Inspection: A random sampling mechanism was implemented to display two examples from every unique class. This qualitative review served the purpose to perform sanity check, where verification was done whether text normalization functions (URL removal/whitespace trimming) were performing as expected. And the result of the check was that the normalization was working as expected.

4) High Dimensional Feature Visualization: To assess the linear separability of the classes, dimensionality reduction was applied to a stratified 10,000 sample from the training set. To perform this, the following tasks were done.

a) Feature Extraction: A TF-IDF vectorizer was applied at the character level (1-gram and 2-gram) to capture morphological patterns.

b) Manifold Learning: Truncated SVD (Latent Semantic Analysis) was used to project the sparse high-dimensional data into 2D space.

c) Visualization: The resulting components were plotted in a scatterplot, colored by class label, to visually inspect if distinct clusters emerged or if classes were heavily overlapping.

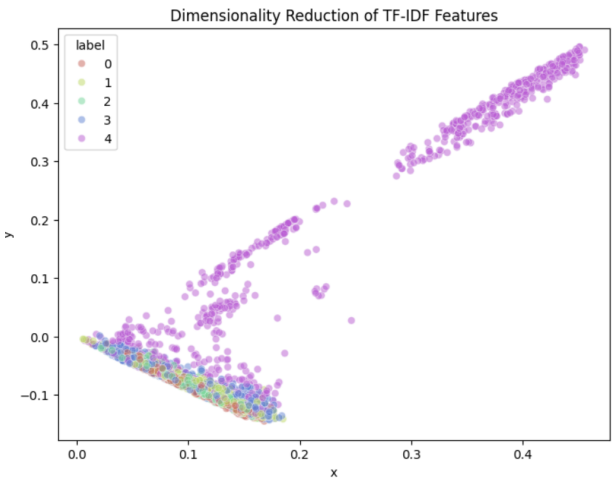

Fig. 3: Dimensionality Reduction (Training Set)

The visualization in Fig. 3 reveals that Label 4 (Purple) forms a distinct, semi-isolated cluster with high variability along the Y-axis. This suggests that the vocabulary or character patterns used in Label 4 are significantly different from the other categories.

The remaining four classes show significant overlap, forming a dense, continuous band along the X-axis. This indicates that these categories share significant lexical or morphological similarities, they might all be related to general news topics with shared vocabulary.

C. Feature Engineering

A distinct implementation detail in this project is that the title and content are combined in such a way that more importance(weight) is given to the title as it contains the most dense topical information about the content. The text column is constructed by concatenating the data as follows: Text = Title + ”[SEP]” + Title + ” “ + Content The title is duplicated so that the frequency (TF) of the words appearing in the title are doubled.This increases the weight of the title-specific key words during the vectorization process. This combined text with title and content is truncated to a maximum length of 3000, to optimize the processing speed and usage of memory.

D. Vectorization

To convert the raw data into numerical form, vectorization is performed using Term Frequency Inverse Document Frequency (TF-IDF) . The Tfidfvectorizer from sklearn is used to do this, with specific hyperparameters configured to capture context beyond a single word. Analyzer, ngram_range, min_df, max_df are some hyperparameters that have been used.

E. Dimensionality reduction for visualization

Truncated SVD (Single Value Decomposition) is used to visualize higher dimensional data in the 2D space. To make it computationally efficient and also to represent the class distribution properly, 10,000 stratified documents are used to generate the scatter plot.

F. Classification Models

Different classifiers were used to compare the performance on the dataset and a pipeline approach was implemented to ensure the vectorizer is fit only on the training data in order to prevent data leakage.

III. Experimental Results and Discussion

All experiments were conducted using the official Sogou News dataset, consisting of 405,000 training samples, 45,000 test samples obtained through a 10% stratified split, and a 60,000-sample held-out test set. Our goal was to compare classical TF-IDF-based models with convolutional neural network approaches, and to examine how hyperparameter tuning affects model performance.

A. Initial TF-IDF Baseline Models

We first evaluated three classical models, Multinomial Naive Bayes, k-Nearest Neighbors (KNN with k=7 after hyperparameter tuning), and Linear Support Vector Machines (LinearSVC) utilizing a character-level TF-IDF representation with n-grams of lengths 1 and 2. We utilized Multinomial Naive Bayes to build a probabilistic baseline, which computes class probabilities based on feature frequencies while assuming independence. Subsequently, we evaluated the distance-based k-Nearest Neighbors (KNN) approach, which classifies articles by determining the predominant label among the nearest examples in the vector space without explicit model training. We then implemented LinearSVC, a margin-based classifier that optimizes a separating hyperplane to maximize the distinction between distinct categories. This geometric method shown notable efficacy for the high-dimensional, sparse TF-IDF feature space characteristic of text data. LinearSVC achieved the strongest results, across all metrics, succeeded by KNN and Multinomial Naive Bayes. The validation performance is summarized below in Table II:

TABLE II: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-score
NaiveBayes	0.91	0.92	0.91	0.91
KNN	0.96	0.96	0.96	0.96
LinearSVC	0.97	0.97	0.97	0.97

These results already indicate that margin-based classifiers are well suited for the high-dimensional TF-IDF space, which is consistent with observations from previous text-classification work.

B. Effect of TF-IDF Analyzer and N-gram Settings

To understand the sensitivity of the model to TF-IDF hyperparameters, we tested several configurations of the vectorizer. Comparing analyzers, we found that a word-level analyzer (accuracy 0.97) outperformed character-level TF-IDF (accuracy 0.90). This matches the structure of Sogou News text, which is provided in pinyin, where meaningful semantic units tend to be multi-word fragments rather than individual characters.

Analyzer	accuracy
char	0.90
word	0.97

TABLE III: Analyzer Accuracy Comparison

ngram_range	accuracy
(1,1)	0.9450
(1,2)	0.9691
(1,3)	0.9698
(2,3)	0.9689
(2,4)	0.9690

TABLE IV: N-gram Accuracy Results

We then tested several n-gram ranges using the word analyzer. The following accuracies were observed:

The best performance was achieved by (1,3), likely because many pinyin terms consist of multi-token structures (e.g., "jing ji ti yu"). Thus, a range of (1,3) captures both individual word signals and short multi-word patterns.

C. Tuning min_df and max_df: Random + Grid Search

Because min_df and max_df interact heavily with vocabulary size, we used a two-stage search strategy. Following prior work, random search was used to quickly explore a wide parameter space, followed by a more focused grid search to refine the results. The best results were achieved with:

Max_df	0.8
Min_df	2

TABLE V: Hyperparameter Settings

This approach is more efficient than pure grid search and avoids missing good regions in the parameter space, a limitation sometimes observed with randomized search alone.

D. Bayesian Tuning of the LinearSVC Regularization Parameter

After tuning the TF-IDF vectorizer, we applied Bayesian optimization over the SVC regularization parameter C, searching logarithmically between 0.001 and 10 for 20 iterations. Bayesian optimization is well suited for SVM tuning because it requires fewer evaluations and tends to converge faster than a full grid search. The best value found was C = 8.365. However, the improvement over the default C = 1 was minimal. This confirms a commonly observed behavior in high-dimensional sparse text representations: once TF-IDF n-grams sufficiently separate the classes, the value of C has limited effect on performance. In our case, the decision boundaries were already well-formed due to the large dataset and expressive feature space.

E. CNN and TextCNN Experiments

We trained two convolutional neural networks on tokenized title-content sequences, each shortened to 1,000 tokens, to compare classical feature-engineering methods with learnt representations. Initially, we implemented a basic 1D CNN;

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.98	0.97	0.98	12000
1	0.95	0.96	0.95	12000
2	0.98	0.99	0.98	12000
3	0.98	0.98	0.98	12000
4	0.96	0.95	0.96	12000
Accuracy			0.97	
Macro Avg	0.97	0.97	0.97	60000
Weighted Avg	0.97	0.97	0.97	60000

TABLE VI: Classification evaluation metrics for news dataset.

	Pred 0	**Pred 1**	**Pred 2**	**Pred 3**	**Pred 4**
True 0	11667	104	132	49	48
True 1	60	11467	58	119	296
True 2	65	37	11830	24	44
True 3	37	88	34	11790	51
True 4	25	427	61	104	11383

Fig. 4: Confusion matrix for the classification model on the test data.

however, its performance was constrained by a narrow receptive field and the tendency towards overfitting. We developed an improved TextCNN architecture to more effectively capture multi-scale semantic information, employing 256-dimensional embeddings and parallel convolution filters with kernel sizes of 3, 4, and 5. We additionally integrated Batch Normalization and optimized hyperparameters, including dropout rates, to enhance optimization stability and accelerate convergence. Although these enhancements enabled the TextCNN to identify both brief word-level patterns and extended contextual signals exceeded the performance of the basic model 1D CNN, neither deep learning method topped the traditional TF-IDF + LinearSVC model, which attained a superior test accuracy of 0.97. The performance disparity could be ascribed to training CNNs from the ground up without pretrained embeddings, especially on pinyin text where semantic associations are more challenging to acquire.

F. Discussion

Across all experiments, the TF-IDF + LinearSVC model consistently delivered the strongest performance, achieving approximately 97% validation accuracy and around 90% macro-F1 on the official test set. Classical models benefited from the high-dimensional TF-IDF representation, which provides strong linear separability and requires minimal parameter tuning. The neural models demonstrated that representation learning is feasible on this dataset, but without pretrained embeddings or transformer architectures, they did not surpass the classical baselines.

Hyperparameter tuning helped refine the model but did not fundamentally change the ranking of model performance. This supports the conclusion that character- and word-level TF-IDF remain extremely competitive for large news-classification tasks, particularly when the text is provided in pinyin or mixed Chinese formats. CNN-based models may outperform these baselines only when combined with pretrained embeddings or significantly deeper networks.

Overall, the experiments highlight the importance of evaluating both classical and neural approaches and demonstrate that strong baselines can be achieved without heavy reliance on modern deep architectures.

For business insights, several actionable insights can be drawn from these experiments for news platforms and related business applications. First, the high accuracy and macro-F1 score of the TF-IDF + LinearSVC model indicate that automated multi-class news classification can reliably reduce the cost of manual labeling and enhance the efficiency of content recommendation systems. Second, the confusion matrix suggests that Finance (Label 1) and Technology (Label 4) have certain semantic overlaps, which indicates that closely related categories should be considered when performing recommendation algorithms and targeted advertising in order to avoid misclassifications. Furthermore, the emphasis on weighting titles in feature engineering underlines the importance of headlines in grasping the key topical information of news articles, which may lead to strategies for optimizing user engagement and click-through rates. Lastly, although CNN-based models hold potential for end-to-end representation learning, results indicate that simple, interpretable, and computationally efficient classical methods are highly effective on large-scale Chinese or pinyin news datasets and enable businesses to achieve high performance without excessive resource expenditure.

G. Final Evaluation on Test Set

Table VI reports the final classification performance of the optimized TF–IDF + LinearSVC model on the official 60,000-sample test set. The model achieved 97% accuracy and a macro-F1 score of 0.97 across all five categories, with individual class F1-scores ranging from 0.95 to 0.98. These strong and consistent per-class results indicate that the model generalizes well to all news topics in the dataset.

Fig. 4 presents the corresponding confusion matrix and provides insight into the remaining sources of error. Most predictions fall on the diagonal, showing that the classes are largely separable under the tuned TF-IDF feature space. The off-diagonal confusion is sparse and primarily concentrated between semantically related categories such as Finance (label 1) and Technology (label 4), which often share overlapping terminology in business and industry reporting. Overall, the results demonstrate that the tuned LinearSVC model is highly effective for this task and provides a strong baseline for future work.

IV. Conclusion

This project successfully demonstrated a comprehensive text classification pipeline for the Sougou News dataset, comparing classical feature engineering and Machine Learning approaches against complex representation learning approaches. By implementing strict preprocessing measures, such as the elimination of URLs and the concatenation of titles and text, we created a controlled setting to evaluate model performance across five separate news categories. The experimental findings indicate that traditional machine learning models maintain persistent effectiveness in high-dimensional text classification.

The word level TF-IDF coupled with a Support Vector Machine (LinearSVC) had the most robustness among the evaluated models, attaining a validation accuracy of 97% and a macro F1 score of 0.97. This affirms that margin-based classifiers are highly appropriate for sparse, high-dimensional feature spaces. The experiments revealed that word-level analysis outperformed character-level approaches, achieving 97% accuracy compared to 90%. Moreover, adjusting the n-grams to a range of (1,3) was crucial for identifying these patterns.

The 1D CNN and TextCNN models, although effective for end-to-end representation learning, did not exceed the performance of the optimized traditional TF-IDF and LinearSVC model. The performance disparity may be linked to the absence of pre-trained embeddings, hindering the networks' ability to acquire intricate semantic associations from pinyin text independently.

This study emphasizes that, although deep learning architectures provide significant potential, well-optimized classical models continue to be highly competitive in terms of accuracy, speed, and interpretability.

For future work, the text classification could be improved in two ways. First, the confusion matrix shows that some labels are misclassified. In order to address that, we could improve our data preprocessing by extracting some representative keywords for each class using methods like KeyBERT in NLP, and train another classifier to handle these ambiguous problems. Second, the current baselines rely on TF-IDF features and traditional classifiers like SVM, Naive Bayes, and KNN, but future improvements could leverage pre-trained word embeddings such as Word2Vec or GloVe, or transformer-based models like BERT or RoBERTa, to capture richer semantic information and contextual relationships, which is likely to improve classification performance, especially for longer or more complex articles.

All the code, trained models, Python files, and notebooks of this project are available at this GitHub repository https://github.com/wellsonhuang/ECEN-758-Fall-2025-Project. A summary of the work completed in the project, with supporting text and visuals, is provided in the blog post https://shouenliu.github.io/Group6.github.io/.

References

[1] A. K. Pathak, M. Chaubey, and M. Gupta, "Randomized-grid search for hyperparameter tuning in decision tree model to improve performance of cardiovascular disease classification," DST-CIMS, Institute of Science, Banaras Hindu University, Varanasi, India. [Online]. Available: https://example.com. [Accessed: Nov. 18, 2025].

[2] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, "Robust optimization of SVM hyperparameters in the classification of bioactive compounds," *Journal of Cheminformatics*, vol. 7, article 38, 2015, doi: 10.1186/s13321-015-0088-0.

[3] J. He, A. S. Khoury, S. Y. Sung, and S. J. Pan, "A Comparative Study on Chinese Text Categorization Methods," Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp. 43–48, Apr. 2007.

[4] C.-H. Wang, H.-D. Chou, and J.-H. Lin, "An Intelligent Chinese Text Clustering System," International Journal of Innovative Computing, Information and Control, vol. 1, no. 4, pp. 667–677, 2005.