VOICE: Vocabulary Size in News Classification - A Comparative Evaluation of Machine Learning Models

Byeongjin Son
AIFFEL Research 12th
sonsation91@gmail.com

Abstract

This study systematically analyzes the impact of vocabulary size on the classification performance of various machine learning models in multi-label news categorization tasks. Experiments were conducted by adjusting the *num_words* parameter from 1,000 to the full vocabulary size of 30,980. The primary classification models evaluated in this study include Naïve Bayes (NBC), Complement Naïve Bayes (CNB), Logistic Regression (LR), Support Vector Classifier (SVC), Decision Tree (DT), Random Forest (RF), Gradient Boosting Tree (GBT), and a soft voting ensemble. The performance of each model was assessed using the weighted F1-score. Experimental results show that decision tree-based models (DT, RF, GBT) exhibited limited performance improvement with increasing vocabulary size and, in some cases, even suffered from performance degradation due to the inclusion of irrelevant features. In the case of probabilistic models, NBC demonstrated a consistent decline in performance as the vocabulary size increased, whereas CNB maintained stable classification performance or showed slight improvements even with larger vocabularies. Linear models (LR) and the soft voting ensemble maintained relatively stable performance across different vocabulary sizes, while SVC achieved the highest performance with an F1-score of 0.8236 when *num_words* was set to 10,000. This study experimentally verifies the influence of vocabulary size on text classification model performance and proposes practical guidelines for selecting the optimal vocabulary size depending on the model type. The findings provide useful insights for designing efficient news classification systems and other text categorization applications.

1 Introduction

The rapid proliferation of digital news content has significantly increased the demand for efficient and accurate text classification techniques. News classification plays a vital role in various applications, including information retrieval, personalized recommendation systems, and sentiment analysis. In this context, one of the fundamental challenges in text classification lies in managing the trade-off between vocabulary size and model performance, particularly when dealing with high-dimensional and sparse data representations.

Vocabulary size directly impacts the quality and efficiency of text classification models. A larger vocabulary may capture more nuanced information but also increases the feature space's dimensionality, potentially introducing noise and leading to sparsity issues. Conversely, a smaller vocabulary can reduce computational complexity but risks losing critical information required for accurate classification.

This study systematically investigates the effect of vocabulary size on the classification performance of various machine learning models in multi-label news categorization tasks. Specifically, we adjust the *num_words* parameter from 1,000 to the full vocabulary size of 30,980 and compare the performance of eight widely-used classification models: Naïve Bayes (NBC), Complement Naïve Bayes (CNB), Logistic Regression (LR), Support Vector Classifier (SVC), Decision Tree (DT), Random Forest (RF), Gradient Boosting Tree (GBT), and a soft voting ensemble.

Our experimental results demonstrate that model performance varies significantly depending on the vocabulary size and the underlying algorithm. Notably, the SVC model achieved the highest weighted F1-score of 0.8236 when the vocabulary size was set to 10,000, suggesting an optimal balance between information richness and dimensionality.

The key contributions of this study are as follows:

- We systematically analyze and compare the impact of vocabulary size on the classification performance of various machine learning models.
- We identify model-specific performance patterns and sensitivities to vocabulary size, offering insights into optimal vocabulary configurations.
- We provide practical guidelines for preprocessing and model selection in multi-label news classification tasks, contributing to the development of high-performance text classification systems.

We believe that our findings can serve as valuable guidelines not only for news classification but also for a wide range of text classification applications. This research highlights the importance of optimizing preprocessing strategies—specifically vocabulary size selection—to improve classification accuracy and computational efficiency.

2 Methods

2.1 Dataset

In this study, the Reuters-21578 news dataset provided by Keras was used. This dataset consists of a total of 11,228 news articles, with each article classified into one of 46 distinct categories. The dataset was split into training and test subsets using a ratio of `test_split=0.2`.

The total number of unique words across the entire dataset (both training and test sets combined) was confirmed to be 30,980. This value was determined by counting the occurrences of unique words within the corpus.

2.2 Data Preprocessing and Feature Extraction

The data preprocessing pipeline consisted of the following steps:

1. Convert the integer-indexed data from Keras back to its original text form.
2. Construct a Document-Term Matrix (DTM) using `CountVectorizer`.
3. Transform the DTM into a TF-IDF matrix using `TfidfTransformer` to capture term importance.

In order to assess the effect of vocabulary size on classification performance, the `num_words` parameter was adjusted to limit the vocabulary size. The vocabulary sizes tested in this study were as follows:

1,000 / 3,000 / 5,000 / 8,000 / 10,000 / 15,000 / 20,000 / 30,980

Words that were not included in the specified vocabulary size were replaced by the special token `UNK`. Table 1 presents the number of `UNK` tokens observed in both the training and test datasets for each vocabulary setting.

Interestingly, even when the full vocabulary size (`num_words=30,980`) was used, a small number of `UNK` tokens were still present. This phenomenon is attributed to the tokenization process of the original dataset, where certain words were already mapped to the `UNK` token. An example sentence from the dataset is as follows:

Table 1: Number of UNK Tokens in Training and Test Sets

Vocabulary Size	**UNK (Train)**	**UNK (Test)**
1,000	284,013	72,116
3,000	136,346	34,589
5,000	86,091	21,958
8,000	48,792	12,633
10,000	35,703	9,177
15,000	18,196	4,725
20,000	9,222	2,436
30,980	1	0

2.3 UNK Token Count Visualization

Figure 1 illustrates the trend of UNK token counts in both the training and test datasets as the vocabulary size increases. The x-axis represents different vocabulary sizes (num_words), and the y-axis indicates the number of UNK tokens. As shown in the figure, the number of UNK tokens decreases significantly as the vocabulary size increases. Beyond a vocabulary size of 20,000, the number of UNK tokens reaches a saturation point, indicating minimal additional benefit in reducing unknown tokens when increasing the vocabulary size further.

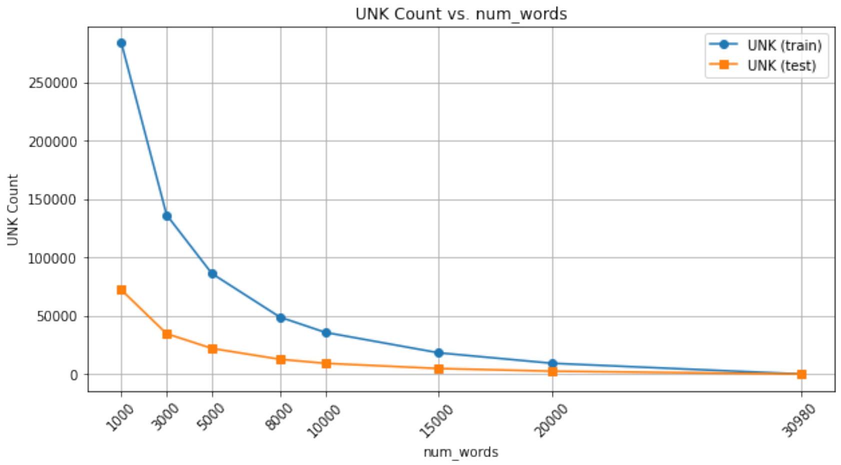

Figure 1: UNK Token count in training and test datasets according to vocabulary size. As num_words increases, the number of UNK tokens significantly decreases, with diminishing returns beyond 20,000 words.

2.4 Classification Models and Hyperparameters

A total of eight machine learning classification models were employed to conduct multi-label news classification tasks. All models were implemented using the Scikit-learn library.

The classification models and their respective hyperparameters are summarized in Table 2.

The Voting Classifier was constructed by combining Logistic Regression, Complement Naïve Bayes, and Gradient Boosting Tree models through a soft voting mechanism. This approach leverages the complementary strengths of different algorithms to improve classification robustness.

Table 2: Classification Models and Hyperparameters

Model	Hyperparameters
Naïve Bayes (NBC)	Default settings
Complement Naïve Bayes (CNB)	Default settings
Logistic Regression (LR)	C=10,000, penalty='l2', max_iter=3000
Support Vector Classifier (SVC)	C=1,000, penalty='l1', max_iter=3000, dual=False
Decision Tree (DT)	max_depth=10, random_state=42
Random Forest (RF)	n_estimators=5, random_state=42
Gradient Boosting Tree (GBT)	random_state=42
Voting Classifier	LR, CNB, GBT combined / soft voting

2.5 Evaluation Metrics

The performance of each classification model was evaluated using two key metrics:

- **Accuracy**: The proportion of correctly classified samples out of all samples.
- **Weighted F1-score**: A harmonic mean of precision and recall, weighted by the number of instances in each class. This metric addresses class imbalance issues by giving proportional weight to each class.

All experiments were conducted under identical data preprocessing, vocabulary size configurations, and hyperparameter settings to ensure a fair comparison across all models. The primary focus of analysis was on the weighted F1-score, as it provides a more comprehensive evaluation of classification performance in the presence of class imbalance.

3 Experiments and Results

3.1 Experimental Environment

All experiments were conducted under a controlled environment to ensure reproducibility and consistency of the results. The hardware and software configurations are summarized below:

- **Operating System:** Ubuntu 20.04 LTS A stable Linux distribution widely used in research for its compatibility and support for machine learning frameworks.
- **CPU:** Intel x86_64 architecture with 2 physical cores and 4 logical threads While the CPU played a minor role in this GPU-assisted training, it was essential for preprocessing and data pipeline operations.
- **RAM:** 17.57 GB Sufficient for handling large document-term matrices and intermediate representations without memory bottlenecks.
- **GPU:** NVIDIA Tesla T4 (Driver Version: 535.230.02, CUDA Version: 12.2) Accelerated computations for TF-IDF processing and some model training components. GPU acceleration reduced training time, ensuring efficient experiment cycles.
- **Python Version:** 3.8 Selected for its compatibility with TensorFlow 2.6.0 and Scikit-learn 1.0, which are crucial to model development in this study.

The key Python libraries and their versions are listed below: These libraries were pinned to specific versions to guarantee the replicability of the experimental outcomes.

- **TensorFlow:** 2.6.0 Used for initial data preprocessing and vectorization stages.
- **Scikit-learn:** 1.0 Served as the core framework for building, training, and evaluating the machine learning models.
- **Matplotlib:** 3.4.3 and **Seaborn:** 0.11.2 Utilized for data visualization and performance analysis.
- **Pandas:** 1.3.3 and **NumPy:** 1.21.4 Used for data manipulation, statistical computations, and supporting the pipeline for dataset management.

Reproducibility Control To ensure experimental reproducibility, random seeds were fixed at 42 for all models and relevant functions within the libraries. Hardware-based determinism was maintained by disabling non-deterministic algorithms in TensorFlow and Scikit-learn where applicable.

3.2 Experimental Results and Analysis

Figure 2 presents the weighted F1-scores of each classification model as a function of vocabulary size (`num_words`). The graph highlights how different models respond to changes in vocabulary size. The Support Vector Classifier (SVC) achieved the highest F1-score of 0.8236 when the vocabulary size was set to 10,000. The Voting Classifier (Soft) demonstrated stable performance as the vocabulary size increased, achieving an F1-score of 0.8165 at the maximum vocabulary size of 30,980.

Tree-based models (DT, RF, GBT) generally showed limited improvements and sometimes performance degradation with larger vocabulary sizes due to the high dimensionality and sparsity of features. On the other hand, Complement Naïve Bayes (CNB) maintained robust performance across various vocabulary sizes.

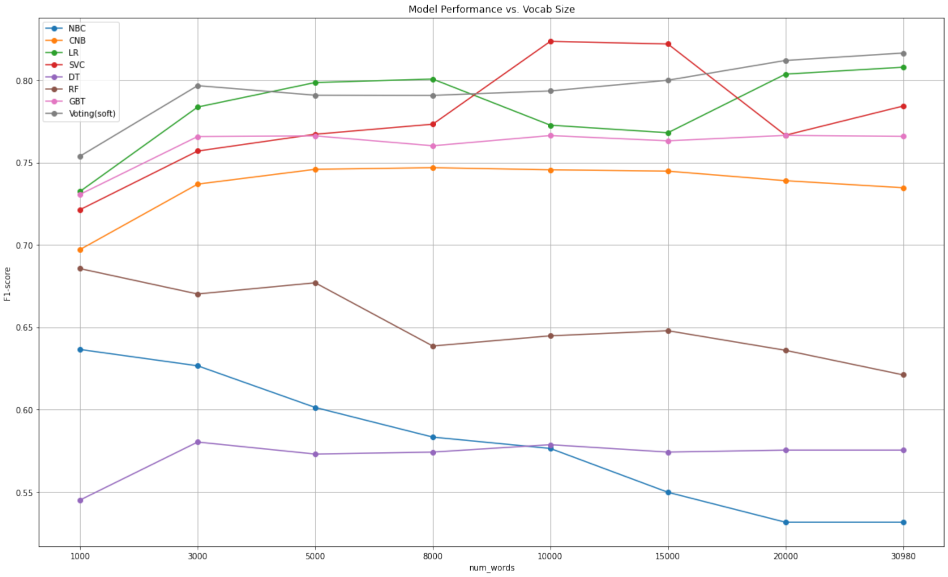

Figure 2: Model performance comparison across different vocabulary sizes. The weighted F1-scores of eight machine learning models are plotted against vocabulary size. SVC shows optimal performance at 10,000 words, while ensemble methods such as Voting (Soft) maintain stable performance at larger vocabulary sizes.

Detailed Analysis by Model

- **Support Vector Classifier (SVC):** Achieved the highest F1-score of **0.8236** at a vocabulary size of 10,000. However, increasing the vocabulary size beyond this point slightly reduced the performance, possibly due to increased feature sparsity and overfitting risks.
- **Voting Classifier (Soft):** Demonstrated consistent and stable performance improvements as the vocabulary size increased. It reached an F1-score of **0.8165** with the full vocabulary size of 30,980. This result highlights the robustness of ensemble learning methods in leveraging diverse classifiers.
- **Logistic Regression (LR):** Showed stable performance across all vocabulary sizes, achieving an F1-score of **0.8079** at the maximum vocabulary size. This suggests that linear models can generalize well even in high-dimensional sparse feature spaces.
- **Complement Naïve Bayes (CNB):** Exhibited gradual improvement as vocabulary size increased but plateaued beyond 15,000 words. CNB maintained moderate robustness to vocabulary scaling compared to traditional Naïve Bayes.

- **Decision Tree (DT), Random Forest (RF), and Gradient Boosting Tree (GBT):** These tree-based models showed minimal improvements with increasing vocabulary size. In some cases, their performance slightly degraded, indicating sensitivity to noisy, high-dimensional data common in text representations.
- **Naïve Bayes (NBC):** Experienced consistent performance degradation as vocabulary size increased. This behavior is attributed to NBC's sensitivity to feature sparsity and its assumption of feature independence, which breaks down in large vocabularies.

3.3 Key Results Summary

Table 3 summarizes the best-performing models at specific vocabulary size thresholds.

Table 3: Best performing models and their F1-scores by vocabulary size.

Vocabulary Size	Best Performing Model	F1-score
10,000	Support Vector Classifier (SVC)	0.8236
30,980	Voting Classifier (Soft)	0.8165

Findings and Implications

- **Mid-range vocabulary sizes (10,000)** are ideal for SVC, balancing dimensionality and data sparsity.
- **Ensemble methods**, such as the Voting Classifier, benefit from increased vocabulary sizes by combining model strengths to offset the weaknesses of individual classifiers.
- **Tree-based models** are prone to performance degradation when handling high-dimensional sparse vectors derived from large vocabularies.
- **Probabilistic models** like NBC struggle with vocabulary expansion, whereas CNB maintains resilience due to its adjusted learning mechanism for class imbalances.

Experimental Reliability To confirm the reliability of the results, each experiment was repeated three times. The F1-scores reported represent the average across these runs, ensuring that the results were not biased by random initialization or sampling variance.

3.4 Results and Analysis

In this section, we present a comprehensive analysis of the experimental results derived from varying the vocabulary size across multiple machine learning models. Performance was primarily measured using the weighted F1-score to address class imbalance within the dataset.

Support Vector Classifier (SVC): SVC demonstrated the best overall performance, achieving a weighted F1-score of **0.8236** when the vocabulary size was set to 10,000. However, increasing the vocabulary size beyond this threshold resulted in a marginal decline in performance. This trend suggests that SVC performs optimally when a balance is struck between vocabulary size and feature sparsity.

Soft Voting Classifier: The ensemble Voting classifier (soft voting) exhibited stable and consistent performance improvements as the vocabulary size increased, culminating in an F1-score of **0.8165** when the entire vocabulary (30,980 words) was utilized. The ensemble's ability to aggregate diverse classifiers helped mitigate the negative effects of increasing feature space dimensionality.

Logistic Regression (LR): LR maintained stable classification performance across different vocabulary sizes, achieving an F1-score of **0.8079** with the full vocabulary. Its robustness can be attributed to its inherent ability to handle high-dimensional sparse data efficiently, especially with proper regularization.

Complement Naïve Bayes (CNB): CNB exhibited incremental performance improvements with vocabulary expansion, stabilizing after 15,000 words. Its resilience to class imbalance and sparse data makes CNB a reliable choice in multi-class text classification.

Tree-based Models (DT, RF, GBT): Decision Tree (DT), Random Forest (RF), and Gradient Boosting Tree (GBT) models showed limited or no performance improvement with increasing vocabulary size. In some instances, their performance degraded, indicating sensitivity to noise introduced by less informative or rare features in large vocabularies.

Naïve Bayes (NBC): NBC consistently underperformed as vocabulary size increased. Its simplifying assumption of feature independence, coupled with high feature sparsity, likely contributed to its inability to generalize effectively in larger feature spaces.

Table 4: Best Performing Models by Vocabulary Size

Vocabulary Size	Best Performing Model	F1-score
10,000	Support Vector Classifier (SVC)	0.8236
30,980	Voting Classifier (Soft)	0.8165

4 Discussion

This section elaborates on the key findings derived from the experiments and provides insights into the observed behavior of different machine learning models with respect to vocabulary size.

4.1 Implications of Vocabulary Size

The results highlight the importance of selecting an appropriate vocabulary size in text classification tasks. An excessively small vocabulary may omit informative terms, while an excessively large vocabulary can introduce noise and sparsity, degrading model performance. A mid-range vocabulary size (10,000) provided the best balance between expressiveness and model complexity for the SVC model.

4.2 Analysis of Tree-based Models

Tree-based models, including DT, RF, and GBT, showed limited adaptability to increasing vocabulary size. The sparsity of text data in high-dimensional spaces hindered their ability to form meaningful splits, often resulting in overfitting or suboptimal generalization. This aligns with the known limitations of decision trees in high-dimensional, sparse data scenarios.

4.3 Analysis of Probabilistic Models

NBC suffered from vocabulary expansion due to its simplifying assumption of feature independence and its reliance on frequency-based probabilities, which become skewed in sparse representations. Conversely, CNB demonstrated robustness, effectively handling increased dimensionality and class imbalance by incorporating complementary class information.

4.4 Analysis of Linear and Ensemble Models

LR and the Voting classifier consistently demonstrated stable and reliable performance. The success of these models in high-dimensional spaces can be attributed to regularization techniques and the ensemble approach's ability to combine complementary strengths of individual models (LR for linearity, CNB for class balance, and GBT for non-linear relationships).

4.5 Analysis of Support Vector Classifier (SVC)

SVC's superior performance at a mid-range vocabulary size suggests its capability to manage the trade-off between dimensionality and sparsity. However, its sensitivity to noisy features in larger vocabularies points to the necessity of careful feature selection or dimensionality reduction techniques.

4.6 Key Insights

- Mid-sized vocabularies (around 10,000 words) are optimal for SVC, balancing complexity and sparsity.
- Ensemble methods (soft voting) demonstrate stable and scalable performance as vocabulary size increases.
- Tree-based models are susceptible to degradation when dealing with high-dimensional sparse vectors.

5 References

1. Lewis, D. D. (1997). Reuters-21578 text categorization test collection, Distribution 1.0. AT&T Labs-Research.
2. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523.
3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
4. Rennie, J. D., et al. (2003). Tackling the poor assumptions of naive bayes text classifiers. ICML.
5. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
6. Hosmer, D. W., et al. (2013). *Applied Logistic Regression*. John Wiley & Sons.
7. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
8. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
9. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232.
10. Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33, 1–39.